

Міністерство освіти і науки України

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ АВТОМОБІЛЬНО-ДОРОЖНІЙ
УНІВЕРСИТЕТ

**КЛАСИЧНА ТА ПРИКЛАДНА МАТЕМАТИЧНА ОСНОВА
ГАЛУЗЕВИХ ДОСЛІДЖЕНЬ У НАУКОВОМУ ПРОСТОРІ
ЗДОБУВАЧІВ ВИЩОЇ ОСВІТИ І МОЛОДИХ ВЧЕНИХ**

***Матеріали Всеукраїнської науково-практичної
конференції здобувачів вищої освіти і молодих вчених
09 квітня 2026 року***

Харків
ХНАДУ
2026

Редакційна колегія:

ЯРХО Тетяна Олександрівна – доктор педагогічних наук, професор, завідувач кафедри вищої математики ХНАДУ (Україна),

ЄМЕЛЬЯНОВА Тетяна Вікторівна – кандидат фізико-математичних наук, доцент, професор кафедри вищої математики ХНАДУ (Україна).

Всеукраїнська науково-методична конференція здобувачів вищої освіти і молодих вчених проведена згідно з планом проведення міжнародних, всеукраїнських та науково-методичних конференцій та семінарів ХНАДУ у 2026 р. (Посвідчення УкрІНТЕІ № 978 від «11» грудня 2025 року)

Відповідальний редактор:

ЄМЕЛЬЯНОВА Тетяна Вікторівна – кандидат фізико-математичних наук, доцент, професор кафедри вищої математики ХНАДУ (Україна).

Класична та прикладна математична основа галузевих досліджень у науковому просторі здобувачів вищої освіти і молодих вчених: матеріали Всеукраїнської науково-практичної конференції здобувачів вищої освіти і молодих вчених. – Харків: ХНАДУ. – 2026. – 242 с.

В збірку включені матеріали Всеукраїнської науково-практичної конференції здобувачів вищої освіти і молодих вчених, в яких розглянуто математичні методи у дослідженнях видатних вчених у технічних, транспортних та економічних галузях, новітні математичні та методичні підходи у вивченні природничо-математичних дисциплін, фундаментальні основи розв’язання професійно-прикладних задач.

Збірник матеріалів розраховано на студентів, магістрів, аспірантів і фахівців, діяльність яких пов’язана з науково-дослідною роботою в різних галузях сучасної науки і освіти. Він може бути корисним методистам, викладачам середніх і вищих навчальних закладів, аспірантам та стажерам.

© Харківський національний
автомобільно-дорожній
університет, 2026

МАТЕМАТИЧНІ МЕТОДИ У ДОСЛІДЖЕННЯХ ВИДАТНИХ ВЧЕНИХ У ТЕХНІЧНИХ, ТРАНСПОРТНИХ ТА ЕКОНОМІЧНИХ ГАЛУЗЯХ

<i>Красільников О.О., Таранець С. С.</i> Метод Ейлера чисельного інтегрування диференціальних рівнянь	7
<i>Сухоставський І. О.</i> Математична спадщина С. Тимошенка у фаховій підготовці інженера-дорожника: від теорії пружності до сучасних дорожніх конструкцій	10

НОВІТНІ МАТЕМАТИЧНІ ТА МЕТОДИЧНІ ПІДХОДИ У ВИВЧЕННІ ПРИРОДНИЧО-МАТЕМАТИЧНИХ ДИСЦИПЛІН

<i>Александров Д. М.</i> Виклики дистанційного навчання вищої математики в технічних ЗВО та шляхи підвищення його ефективності: погляд здобувача.....	14
<i>Базака А. А.</i> Математичні підходи та засоби об'єктно орієнтованого програмування у моделюванні аномалій фінансових транзакцій	17
<i>Бесараба Є. В.</i> Математичні методи та алгоритмічні засоби моделювання інтелектуальної поведінки в реалістичних стратегічних ігрових системах	22
<i>Білозуб М. М.</i> Розроблення математичної структури інформаційної системи для створення і відстеження фінансових витрат.....	30
<i>Білошицький Н. Д.</i> Проектування інформаційної системи “Розклад занять для студентів” із використанням математичних методів технології об'єктно орієнтованого програмування	34
<i>Бондаренко М. В.</i> Математичне підґрунтя проектування графічного калькулятора для обчислення об'ємів і площ поверхонь геометричних тіл	38
<i>Бондаренко О. В.</i> Математичні методи та алгоритмічні засоби побудови інформаційної системи ідентифікації діамантів	44
<i>Бугай О. Ю.</i> Основи алгоритмічного моделювання програмних систем автоматизації біржової торгівлі	50
<i>Вавілов Д. Ю.</i> Математичні принципи розроблення розважального чат-боту на платформі Telegram	54
<i>Кайдалова А. С., Вельма А. Р., Базунова С. М.</i> Оптимізація логістичних витрат: від теорії матриць до реальних розрахунків	58
<i>Кириченко К. Ю.</i> Математичні та алгоритмічні засади створення	

інформаційної системи підтримки продажу автомобілів.....	61
Клименко Д. А. Алгоритми шифрування текстів на основі об'єктно орієнтованих математичних моделей	67
Ключище І. С. Математичні методи та технології штучного інтелекту в інформаційно-аналітичних системах прогнозування ресурсів підприємства.....	73
Корнілов Р. В. Математичні основи проектування і розроблення інформаційної системи транспортної компанії.....	80
Костенко Є.О, Старіков Н.Є. Простіший метод прогнозу та коригування при наближеному інтегруванні диференціальних рівнянь першого порядку.....	85
Краснокутський Б.В. Елементарні випадкові функції як складові випадкових процесів з неперервним часом і неперервними станами.....	89
Красько К. В. Математичні моделі та технології проектування головоломкових комп'ютерних ігор із використанням геймдизайн-орієнтованих підходів.....	94
Могильницький М.В. Шляхи викладання математичних дисциплін для потреб робітників-верстатників	98
Nazar Tanasiuk, Liubov Nemchenko Stable pairs in bipartite graphs and examples of their use in mathematical modular arithmetic	102
Огнєва Х. О. Інтегральний підхід до оцінювання сумарного попиту на транспортні послуги.....	109
Скорик А. В. Характеристики та закон розподілу випадкової функції, випадкова величина якої підпорядкована розподілу Релея	113
Ференс Д.Ю. Застосування SOLIDWORKS у вивченні природничо-математичних дисциплін	117

ФУНДАМЕНТАЛЬНІ ОСНОВИ РОЗВ'ЯЗАННЯ ПРОФЕСІЙНО-ПРИКЛАДНИХ ЗАДАЧ

Богач І.А., Разумовський В.Е. Аналіз теоретичних основ моделювання транспортних процесів засобами диференціальних рівнянь.....	121
Гаркуша Я.О., Богдан А.О. Використання методів лінійної алгебри для аналізу структури транспортних мереж	125
Кукульчук В. О. Інформаційна освітня інфраструктура нового покоління: математичне моделювання і штучний інтелект.....	128

Локтіонова А.О., Троян Є.В. Дослідження моделі накопичення транспортної черги з використанням диференціальних рівнянь	133
Лунчев Г. О. Інформаційно-математичне забезпечення системи казуальних головоломок	137
Максименко В. М. Математичні основи модульної медичної інформаційної системи обробки біомедичних даних пацієнтів.....	144
Maxim Ptashniy, Yurii Dubovskyi, Maksim Yatsyna Searching for mutually exclusive properties in selection problems using ultrafilters.....	150
Міщенко М.С. , Кравчик Д.В. Характеристики та закони розподілу елементарних випадкових функцій, як складових випадкових процесів.....	156
Миколук П. О. Розроблення інформаційної системи структурного математичного моделювання бізнес-процесів	160
Миронюк С. О. Дослідження і реалізація прикладних математичних методів розпізнавання символно-цифрових знаків	164
Новіцький В. В. Розроблення гри «Баскетбол» із використанням математичних основ технології об'єктно орієнтованого програмування.....	168
Онофрійчук В.О. Математичне підґрунтя розроблення ігрової системи на платформі Unity з використанням геймдизайн орієнтованих підходів.....	173
Пилипенко Д. Д. Математичні принципи проектування калькулятора тригонометричних функцій «SYNTAXMIND».....	178
Посмаренко О. С. Розроблення ігрової системи жанру “Аркадаїд” з використанням математичних основ технології об'єктно орієнтованого програмування.....	184
Прилуцький М. І. Розробка математично обґрунтованої інформаційної системи парсингу даних.....	192
Руденко О. В. Розроблення гри «Три в ряд» із використанням математичних основ технології об'єктно орієнтованого програмування.....	197
Свидан А. А. Розроблення гри «Футбол» із використанням математичних основ технології об'єктно орієнтованого програмування.....	200
Сидоров Н. Г. Математичні принципи створення інформаційно-аналітичної системи для дослідження ринку нерухомості.....	207
Сіленко Д. О. Концепція математичного моделювання системи цифрового моніторингу стану здоров'я людини.....	212
Степура І. В. Методики математичного та нейромережевого аналізу психофізіологічних даних.....	216
Таран В.В., Чірва А.А. Застосування інтегрального числення для оцінювання інтенсивності транспортних потоків	220
Турушева М. В. Математичне моделювання ігрового застосунку жанру спортивного симулятора футболу.....	222
Філіпенков О. В. Дослідження математично обґрунтованої	

інформаційної системи для аналізу діяльності співробітників компанії.....	229
Шитова М. Д. Розроблення математично обгрунтованої моделі ігрової системи в жанрі Платформера із використанням рушія Unity.....	236

МАТЕМАТИЧНІ МЕТОДИ У ДОСЛІДЖЕННЯХ ВИДАТНИХ ВЧЕНИХ У ТЕХНІЧНИХ, ТРАНСПОРТНИХ ТА ЕКОНОМІЧНИХ ГАЛУЗЯХ

Красільніков О.О. (здобувач бакалаврату, 1 курс),

Таранець С. С. (здобувач бакалаврату, 1 курс),

Науковий керівник – ст. викл. Нестеренко В. О.

Харківський національний автомобільно-дорожній університет

МЕТОД ЕЙЛЕРА ЧИСЕЛЬНОГО ІНТЕГРУВАННЯ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ

Чисельні методи інтегрування диференціальних рівнянь застосовують звичайно частіше інших, оскільки, у питаннях практичного застосування математики частіше за все потрібна не формула, яка дає розв’язок, а його числові значення при певних значеннях аргументу, які ще необхідно буде обчислити.

Розгляд чисельних методів починаємо з метода Ейлера. Цей метод є дещо грубим і може застосовуватись для орієнтованих обчислень. Однак ідеї, що закладені у основу цього методу, є, по суті, відхідними для дуже широкого класу чисельних методів. Тому корисно розібрати його докладніше.

Нехай задано диференціальне рівняння першого порядку

$$y' = f(x, y) \quad (1)$$

з початковою умовою

$$y_0 = y(x_0) \quad (2)$$

Візьмемо число h настільки малим, щоб для всіх x в інтервалі (x_0, x_1) , де $x_1 = x_0 + h$, значення функції y мало відрізняється від y_0 (функція y неперервна). Тоді для вказаного інтервалу змінення x можна написати

$$y = y_0 + (x - x_0)y'_0 = y_0 + (x - x_0)f(x_0, y_0),$$

де $y'_0 = f(x_0, y_0)$ є значення похідної y' в точці $x = x_0$. Іншими словами, на цьому інтервалі крива замінюється відрізком прямої (дотичної до кривої на лівому кінці інтервалу).

Для правого кінця інтервалу $x_1 = x_0 + h$ одержимо

$$y|_{x=x_1} = y_0 + hy'_0 = y_1 .$$

Так саме для $x = x_2 = x_0 + 2h$ можна написати

$$y_2 = y_1 + hy'_1, \quad y'_1 = f(x_1, y_1). .$$

Якщо продовжимо знаходити значення функції за тим же законом, дістанемо

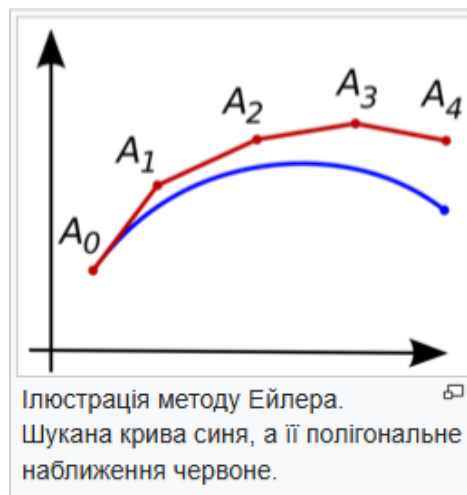
$$y_{k+1} = y_k + hy'_k ,$$

причому $y'_k = f(x_k, y_k)$.

Користуючись введеними позначеннями, схему метода Ейлера можна подати формулами

$$\begin{cases} \Delta y_k = hy'_k \\ y_{k+1} = y_k + \Delta y_k \end{cases} . \quad (3)$$

Таким чином, мова йде про послідовне обчислення різниць шуканої функції.



Геометричний зміст методу Ейлера можемо побачити із наведеного рисунку. Інтегральна крива замінюється ламаною, в якій ланки мають сталу горизонтальну проекцію h . Перша ланка дотикається до істинної інтегральної кривої у точці (x_0, y_0) .

Установлені формули можуть бути одержані і з інших міркувань, які будуть використовуватися в подальшому. З рівняння (1) можна написати

$$y(x_{k+1}) = y(x_k) + \int_{x_k}^{x_{k+1}} f(x, y(x)) dx, \quad y \quad (4)$$

Якщо в інтегралі формули (4) взяти функцію $f(x, y(x))$ сталою, яка дорівнює початковому значенню в точці $x = x_k$, то величина інтеграла буде дорівнювати $hf(x_k, y_k)$, так що формула (4) перетворюється у формулу (3).

На практиці всі обчислення зводимо у таблицю такого вигляду:

(1)	(2)	(3)	(4)
x	y	Δy	y'

Похибка та збіжність

Локальна похибка: На одному кроці метод Ейлера дає похибку порядку $O(h^2)$ (помилка на одному кроці пропорційна квадрату кроку).

Глобальна похибка: На всьому інтервалі розв'язання метод має порядок $O(h)$, що робить його методом першого порядку. Це означає, що при зменшенні кроку h вдвічі, глобальна похибка також зменшується приблизно вдвічі.

Варто зауважити, що вибір кроку h є компромісом: занадто великий крок призводить до значної втрати точності, тоді як надто малий крок різко збільшує кількість обчислень ітерацій h .

Збіжність: Метод Ейлера є збіжним, тобто при $h \rightarrow 0$ наближений розв'язок прямує до точного.

Переваги та недоліки методу Ейлера.

Переваги: Простота реалізації. "Самостартування звітку": для початку роботи потрібна лише одна початкова точка. Легко змінювати розмір кроку h у процесі обчислень.

Недоліки: Низька точність. Повільна збіжність до точного розв'язку. Накопичення похибки на великих інтервалах.

***Анотація.** В роботі розглядається метод Ейлера, який є основою для подальшого розвитку методів наближеного інтегрування диференціальних рівнянь. Його основні формули, похибки, переваги та недоліки. Показано геометричний зміст методу.*

Література

1. Гой О.В., Диференціальні рівняння / Гой О.В., Махней О.В. – Тернопіль – Навчальна книга – Богдан, 2014. – 24 с.
2. Богач І.В. Чисельні методи розв’язання диференціальних рівнянь засобами МATHCFD / Богач І.В., Краковецький О.Ю., Крилик Л.В., Навчальний посібник – Вінниця, ВНТУ, 2020. – 107 с.

Сухоставський І. О. (здобувач бакалаврату, 1 курс),
Науковий керівник – доц Михайленко І. В.
Харківський національний автомобільно-дорожній університет

МАТЕМАТИЧНА СПАДЩИНА С. ТИМОШЕНКА У ФАХОВІЙ ПІДГОТОВЦІ ІНЖЕНЕРА-ДОРОЖНИКА: ВІД ТЕОРІЇ ПРУЖНОСТІ ДО СУЧАСНИХ ДОРОЖНИХ КОНСТРУКЦІЙ

Степан Прокопович Тимошенко (1878–1972) - постать, яка змінила обличчя світової інженерії. Будучи вихованцем української школи механіки, він зумів поєднати математичну теорію з практичними потребами будівництва. Для сучасного інженера-дорожника спадщина Тимошенка - це не просто історія науки, а фундаментальний освітній стандарт. Його методи дозволяють створювати дорожні конструкції, здатні витримувати сучасні екстремальні навантаження: від великовагового транспорту до різких кліматичних змін.

В основі проектування доріг лежить теорія пружності, яка, якщо пояснювати просто, вивчає «характер» дорожнього полотна - те, як воно пручається навантаженню від коліс. Ще Степан Тимошенко свого часу довів, що будь-яку ділянку траси треба оцінювати як складну систему, де одночасно працюють три умови. По-перше, це звичайна статика: кожна піщинка ґрунту чи крапля бітуму має перебувати в рівновазі, щоб сума всіх діючих на них сил дорівнювала нулю.

По-друге, важлива цілісність або «геометрична сумісність» - дорога повинна прогинатися плавно і безперервно, як єдине ціле, не розсипаючись на окремі фрагменти й не утворюючи внутрішніх пустот. І нарешті, головний інструмент інженера - закон Гука. Саме він допомагає зрозуміти фізику процесу: чи виявиться покриття достатньо «пружинистим», щоб повернути форму після проїзду важкої фури, чи пасує перед вагою, залишаючи після себе вичавлену колію як ознаку залишкової деформації.

Одним із найбільших досягнень Тимошенка було впровадження бігармонійного рівняння для розрахунку плоских задач:

$$\frac{\partial^4 \phi}{\partial x^4} + 2 \frac{\partial^4 \phi}{\partial x^2 \partial y^2} + \frac{\partial^4 \phi}{\partial y^4} = 0$$

Для інженера цей математичний інструмент — справжній лайфхак для розрахунку тиску. Головна суть у тому, що навантаження від колеса не «тисне» просто вниз вузьким стовпом, а розходитьсь вглиб широким конусом. Розуміючи це, можна зекономити: на самій поверхні ми кладемо міцний і дорогий асфальт, а нижче, де напруження розсіюється і стає слабшим, використовуємо простіші та дешевші матеріали.

Освітня модель Степана Тимошенка донині залишається фундаментом підготовки транспортних інженерів у всьому світі — від Стенфорда до українських політехнік. Головна цінність його підходу полягає у вихованні справжньої інженерної інтуїції: математика тут виступає лише інструментом, тоді як на перше місце виходить розуміння фізичної суті процесів. Замість простого підставляння цифр у формули, студент вчиться аналізувати наслідки кожного рішення, наприклад, як зміна товщини щебеневого шару всього на 5 см вплине на «життя» всієї магістралі. У еру цифрових технологій ця база стає незамінним фільтром для верифікації комп'ютерних розрахунків. Навіть працюючи в найскладніших програмах типу CAD чи BIM, фахівець, який знає методику

«балки Тимошенка», здатний миттєво помітити системну помилку алгоритму, що рятує від катастроф на етапі будівництва. Такий міждисциплінарний підхід перетворює майбутнього дорожника на універсала, який одночасно розуміє хімію в'язкопружних бітумів, складну геометрію мостових опор та логіку опору матеріалів.

Револьюційна модель балки Тимошенка й досі вважається еталоном точності для розрахунку жорстких дорожніх покриттів, адже вона враховує критично важливі чинники - зсув та інерцію. У дорожніх конструкціях зі значною товщиною шарів виникають специфічні деформації, які стають очевидними під час гальмування важкого транспорту. Завдяки розрахункам Тимошенка інженери можуть точно спроектувати зчеплення між шарами так, щоб верхній шар асфальту не «зсунувся» відносно основи під дією горизонтальних сил. Крім того, ці рівняння враховують вібраційні процеси, що є життєво необхідним при проектуванні трас із високою інтенсивністю руху, які щодня витримують тисячі ударних циклів.

Практичне застосування цих методів сьогодні охоплює найскладніші об'єкти - від автобанів до аеродромів. Коли йдеться про жорсткі цементобетонні плити, теорія пружності допомагає вирішити проблему термічних напружень, точно визначаючи відстань між температурними швами, щоб бетон не розтріскався від спеки. Також це дозволяє врахувати «крайовий ефект» і вчасно посилити найбільш вразливі зони плит. Навіть при роботі зі слабкими ґрунтами та геосинтетикою формули Тимошенка залишаються незамінними: вони дозволяють вирахувати критичну межу, за якою насип може втратити стійкість і просто «попливти», що допомагає запобігти катастрофі ще на етапі проекту.

Спадщина Степана Тимошенка органічно інтегрувалася у цифрову епоху: сучасний софт, як-от Autodesk Civil 3D чи ANSYS, повністю побудований на його енергетичних принципах. Зокрема, принцип мінімуму потенційної енергії

дозволяє комп'ютерним алгоритмам миттєво знаходити найефективнішу форму конструкції, що заощаджує мільйони бюджетних коштів на етапі будівництва.

Висновки. Підбиваючи підсумок, можна впевнено сказати, що математичні напрацювання Тимошенка - це фундамент, на якому тримається вся сучасна дорожня галузь. Для майбутнього інженера його праці відіграють подвійну роль: вони забезпечують точний математичний інструментарій та водночас формують критичне професійне мислення. Завдяки теорії пружності ми навчилися проектувати не просто асфальтне покриття, а складні й довговічні транспортні артерії, що працюють як єдиний пружний організм. Спадщина нашого великого земляка продовжує жити в кожному сучасному проекті, гарантуючи надійність інфраструктури майбутнього.

Анотація. У статті розглянуто фундаментальний внесок С.П. Тимошенка у розвиток теорії пружності та її практичне застосування в сучасному дорожньому будівництві. Особливу увагу приділено моделі балки Тимошенка, яка дозволяє враховувати зсувні деформації та динамічні навантаження, що є критичним для проектування жорстких покриттів та аеродромів. Висвітлено роль методичної спадщини вченого у формуванні інженерної інтуїції сучасних. Також обґрунтовано, що принципи Тимошенка залишаються основою для створення довговічних транспортних артерій та ефективного використання дорожньо-будівельних матеріалів у цифрову епоху.

Ключові слова: теорія пружності, Степан Тимошенко, дорожні конструкції, балка Тимошенка, закон Гука, напружено-деформований стан, жорсткі покриття, інженерна освіта.

Література

1. Тимошенко С. П., Гуд'єр Дж. Теорія пружності. Київ: Наукова думка, 1970. 560 с. (Класична праця, що описує фундаментальні принципи).
2. Тимошенко С. П. Опір матеріалів. У 2-х т. Київ: Вища школа, 1991. (Основа для розуміння розрахунків балок та плит).
3. Білятинський О. А., Хом'як Я. В. Проектування автомобільних доріг. Київ: Вища школа, 1997. 518 с. (Підручник, де інтегровані методики Тимошенка в українську дорожню науку).
4. Васильєв В. В. Механіка конструкцій: від Тимошенка до сучасних чисельних методів. Вісник транспортної академії. 2018. № 12. С. 45–52.
5. ДСТУ В.2.3-37641918-544:2014. Автомобільні дороги. Проектування нежорстких дорожніх одягів. Київ: ДерждорНДІ, 2014. (Нормативний документ, що базується на теорії пружності).

НОВІТНІ МАТЕМАТИЧНІ ТА ПЕДАГОГІЧНІ ПІДХОДИ У ВИВЧЕННІ ПРИРОДНИЧО-МАТЕМАТИЧНИХ ДИСЦИПЛІН

Александров Д. М.. (здобувач бакалаврату, 1 курс),
Науковий керівник – доц Михайленко І. В.
Харківський національний автомобільно-дорожній університет

ВИКЛИКИ ДИСТАНЦІЙНОГО НАВЧАННЯ ВИЩОЇ МАТЕМАТИКИ В ТЕХНІЧНИХ ЗВО ТА ШЛЯХИ ПІДВИЩЕННЯ ЙОГО ЕФЕКТИВНОСТІ: ПОГЛЯД ЗДОБУВАЧА

Перехід освітнього процесу в дистанційний формат став серйозним викликом для здобувачів вищої освіти, особливо під час вивчення фундаментальних дисциплін. Класична математика (математичний аналіз, лінійна алгебра, диференціальні рівняння, теорія ймовірностей) формує основу інженерного мислення та є необхідною передумовою для подальшого опанування фахових дисциплін.

Для здобувачів дорожньо-будівельного факультету математика має безпосередній прикладний характер. Розрахунок прогинів мостових конструкцій, аналіз напружено-деформованого стану дорожніх покриттів, моделювання навантажень на аеродромні плити, оцінювання стійкості насипів та укосів ґрунту — усі ці завдання базуються на використанні диференціальних рівнянь, методів інтегрування, векторного аналізу та чисельних методів. Без глибокого розуміння математичного апарату виконання інженерних розрахунків стає формальним і не забезпечує необхідної точності.

В умовах аудиторного навчання складні теоретичні положення супроводжуються детальним поясненням викладача та графічними побудовами. У

дистанційному форматі така взаємодія ускладнюється, що призводить до підвищення рівня абстракції та зниження якості первинного сприйняття матеріалу.

Аналіз досвіду дистанційного навчання дозволяє виокремити кілька ключових проблем.

По-перше, це обмеженість зворотного зв'язку. Відсутність миттєвої реакції аудиторії ускладнює виявлення моменту, коли група втрачає логіку математичних міркувань.

По-друге, труднощі з цифровим записом математичних виразів знижують ефективність комунікації. Складні формули, системи рівнянь та інтеграли важко оперативно вводити за допомогою стандартних засобів текстового спілкування.

По-третє, спостерігається когнітивне перевантаження. Велика кількість формул і доведень, поданих у вигляді статичних презентацій, зменшує концентрацію уваги та сприяє формуванню так званої «ілюзії розуміння», коли студент пасивно сприймає матеріал, але не здатен самостійно застосувати його під час виконання інженерних розрахунків.

Для подолання зазначених труднощів доцільним є впровадження комплексу практичних рекомендацій. Інструментальний аспект передбачає використання інтерактивних математичних середовищ та візуалізаторів [1], а також інтеграцію математичних розрахунків із системами автоматизованого проектування.

Застосування програмних засобів для побудови схем мостових елементів або моделювання геометрії дорожніх конструкцій дозволяє поєднати аналітичну формулу з її інженерною інтерпретацією [2].

Методологічний аспект пов'язаний з алгоритмізацією розв'язування задач. Перед виконанням розрахунку студент повинен визначити тип задачі, записати базові співвідношення, перевірити умови застосування формул та лише після цього переходити до числових обчислень. Такий підхід формує системність мислення та запобігає механічному копіюванню розв'язків.

Комунікативний аспект передбачає активну взаємодію у цифровому середовищі. Створення малих навчальних груп для спільного аналізу задач із розрахунку мостових прогонів або дорожніх конструкцій дозволяє відтворити атмосферу аудиторного обговорення та сприяє глибшому розумінню матеріалу.

Доцільним є розгляд дистанційного опанування класичної математики як трикомпонентної моделі навчального процесу. Перший компонент - теоретичний, що передбачає системне засвоєння понять, означень та математичних співвідношень. Другий - інженерно-прикладний, який полягає у перенесенні абстрактних формул у площину галузевих задач (розрахунок мостових конструкцій, аналіз напружено-деформованого стану дорожніх покриттів, моделювання навантажень на аеродромні споруди). Третій - цифровий, що забезпечує візуалізацію, перевірку та інтерпретацію результатів за допомогою сучасних програмних засобів. Взаємодія зазначених компонентів формує цілісне математичне мислення та зменшує ризик формального засвоєння матеріалу.

Висновки. Отже, дистанційне опанування класичної математики в технічному університеті є складним, але керованим процесом. Ефективність фундаментальної математичної підготовки визначається активною позицією здобувача, раціональним використанням цифрових інструментів та усвідомленням прикладного значення математичних методів у проектуванні та експлуатації мостів, доріг і аеродромних споруд.

Анотація. У статті розглядаються основні проблеми дистанційного вивчення класичної математики здобувачами вищої освіти технічних спеціальностей. Проаналізовано труднощі, пов'язані з втратою зворотного зв'язку, цифровим записом математичних виразів та когнітивним перевантаженням. Особливу увагу приділено ролі фундаментальної математичної підготовки у формуванні професійних компетентностей майбутніх інженерів дорожньо-будівельної галузі. Запропоновано комплекс практичних рекомендацій щодо підвищення ефективності засвоєння математичних дисциплін у дистанційному форматі.

Ключові слова: дистанційне навчання, вища математика, інженерна підготовка, дорожнє будівництво, цифрові інструменти, самотійна робота.

Література

1. Петрук, В. А., Клеопа, І. А. (2021). Дистанційне навчання вищої математики студентів технічного університету. *Сучасні інформаційні технології та інноваційні методики навчання у підготовці фахівців: методологія, теорія, досвід, проблеми*, (60), 290–299.
2. Кіяновська, Н. М. (2015). Використання систем комп'ютерної математики у процесі навчання вищої математики студентів технічних ВНЗ. *Сучасні інформаційні технології та інноваційні методики навчання в підготовці фахівців: методологія, теорія, досвід, проблеми*, (41), 337–342.

УДК 004.942:[327

Базака А. А. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О
Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-технічний університет імені академіка Юрія Бугая» (Київ)

МАТЕМАТИЧНІ ПІДХОДИ ТА ЗАСОБИ ОБ'ЄКТНО ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ У МОДЕЛЮВАННІ АНОМАЛІЙ ФІНАНСОВИХ ТРАНЗАКЦІЙ

Стрімкий розвиток цифрових фінансових послуг та перехід більшої частини грошового обігу в онлайн-простір призвели до значного зростання кількості транзакцій, що обробляються в режимі реального часу. Разом із технологічним прогресом спостерігається і професіоналізація фінансової злочинності, де зловмисники використовують дедалі складніші методи обходу систем безпеки. Традиційні підходи до моніторингу, що базуються на жорстких алгоритмах та статичних правилах, стають неефективними в умовах динамічних змін ринку та постійної появи нових типів шахрайства. Це зумовлює гостру потребу у розробці інтелектуальних моделей виявлення аномалій [1], здатних ідентифікувати підозрілі операції шляхом аналізу прихованих закономірностей у великих масивах даних.

Актуальність дослідження визначається необхідністю створення ефективного інструментарію для захисту активів фінансових установ та їх

клієнтів. Розробка моделі, що поєднує в собі сучасні алгоритми аналізу даних та надійну архітектуру об'єктно-орієнтованого проектування, дозволяє забезпечити високу швидкість реагування на загрози та мінімізувати економічні збитки від несанкціонованих операцій у довгостроковій перспективі [2].

Мета дослідження полягає у розробленні моделі для виявлення аномалій у фінансових операціях із використанням технології об'єктно орієнтованого програмування. *Завдання:* 1. Здійснити аналіз вимог до програмної системи і дослідити існуючі застосунки, сформувавши вимоги для виявлення аномалій фрод-моніторинг, а також сформувавши технічні вимоги до інтелектуальної моделі аналізу транзакцій. 2. Здійснити проектування модулів системи на основі принципів ООП, розробивши гнучку архітектуру з використанням відповідних патернів проектування та структуру бази даних для зберігання фінансових операцій і результатів їхнього аналізу. 3. Здійснити програмну реалізацію системи, розробивши функціонал для препроцесингу даних, інтеграції математичної моделі виявлення аномалій та створення інтерфейсу взаємодії з користувачем. 4. Здійснити тестування розробленої системи на предмет точності виявлення аномалій метрики Precision, Recall, оцінити продуктивність алгоритмів та перевірити відповідність системи вимогам безпеки та функціональності.

Система використовує файлову структуру та внутрішні об'єкти класів для тимчасового зберігання транзакцій. Обробка даних розділена на етапи: Вхід: об'єкт класу *Transaction* або *JSON*-пакет. Трансформація: перетворення логів у векторний формат, який зрозумілий моделі. Вихід: присвоєння статусу *Is Anomaly*: True/False та генерація лог-запису *Alert Report*.

У рамках дослідження було використано модуль валідації: перевіряє вхідний *JSON* на наявність усіх необхідних полів. Модуль трансформації: використовує *DataPreprocessor* для нормалізації даних, щоб великі значення не спотворювали роботу алгоритму. Модуль логування: записує кожну виявлену

аномалію у файл або консоль [3]. Для гарантування безпеки фінансових операцій та захисту звітів у системі реалізовано симетричний алгоритм шифрування *AES* через бібліотеку *cryptography.fernet* який використовується для безпечного обміну ключами та передачі сповіщень у Телеграм-бот. Виконує перетворення текстових лог-записів у зашифрований байт-код перед їх збереженням у Журналі операцій. Це запобігає витоку конфіденційної інформації у разі несанкціонованого доступу до файлів системи.

Програмна реалізація системи SHIELD-ANOMALY базується на клієнт-серверній архітектурі, де серверна частина реалізована на мові Python з використанням фреймворку Flask, а користувацький інтерфейс побудований на базі сучасних веб-технологій HTML5, CSS3, JavaScript. Відповідно до розробленого дизайну, інтерфейс системи орієнтований на антифрод-менеджера [4]. *Візуальна аналітика*: центральну частину екрана займає інтерактивний графік розсіювання. Він візуалізує розподіл транзакцій за двома ключовими параметрами тривалість *dur* та обсяг даних *sbytes*, що дозволяє оператору миттєво оцінити масштаб потенційної атаки. *Інформаційна панель показників* : у верхній частині відображаються агреговані дані *Total Transactions* загальна кількість завантажених записів. *Anomalies Detected* кількість виявлених підозрілих операцій. *Contamination Rate* відсоток аномалій у вибірці. *Елементи управління*: бічна панель містить кнопки для завантаження нових даних, запуску алгоритму аналізу та експорту результатів у формат лог-запису *Alert Report*.

Опис програмного коду та архітектурних рішень. Клас *DataLoader*: реалізує логіку зчитування вхідних *JSON* та *CSV* файлів. Він інкапсулює механізм перевірки наявності обов'язкових полів, що забезпечує стійкість системи до некоректних вхідних даних. Клас *DataPreprocessor*: виконує підготовку даних. Важливою частиною є використання *StandardScaler* для нормалізації ознак, що необхідно для коректної роботи алгоритму *Isolation Forest*. Реалізація моделі

AnomalyDetector: використовується принцип поліморфізму [5]. Базовий абстрактний клас визначає методи *fit()* та *predict()*, а конкретний клас-нащадок *IsolationForestModel* реалізує саму логіку виявлення аномалій.

Журнал операцій для ретроспективного аналізу, де зберігається історія всіх перевірених транзакцій із зазначенням їхніх параметрів час, тривалість, об'єм даних та вердикту моделі. *Налаштування Sentinel core для конфігурації системи*: користувач може обирати активну модель аналізу Isolation Forest або One-Class SVM. Встановлювати поріг чутливості для алгоритмів [6]. Керувати шляхами до наборів даних завантаження нових версій UNSW-NB15.

Telegram бот реалізує механізм миттєвих сповіщень. При виявленні критичної аномалії атаки типу Backdoor або DoS, сервер через API надсилає автоматичне повідомлення в Telegram [7]. Це дозволяє реагувати на загрози навіть за межами робочого місця. Після активації система готова до прийому поточних даних або завантаження статичних файлів для ретроспективного аналізу. Процес вибору даних є сполучною ланкою між технічними логами та інтелектуальним висновком моделі.

Анотація. У рамках дослідження здійснено проектування і практичну реалізацію інформаційної системи для виявлення аномалій у фінансових операціях із використанням технології об'єктно орієнтованого програмування. Перехід до глибокого навчання та впровадження архітектур автоенкодерів дозволить моделі самостійно виділяти складні ознаки шахрайства. Використання рекурентних нейронних мереж допоможе виявляти аномалії в цілих ланцюжках дій користувача. Інтеграція моделі в стрім-сервіси забезпечить миттєве блокування підозрілих транзакцій у момент їх проведення. У рамках дослідження розроблено повноцінний API-сервіс у контексті спрощення інтеграції алгоритму в чинні банківські системи. Тестування на масивах даних великого обсягу дозволило перевірити стійкість та швидкість роботи моделі в умовах реального навантаження.

Ключові слова: виявлення аномалій, фінансові транзакції, дослідницьке навчання, варіативні моделі, математичне моделювання, об'єктно орієнтованого програмування, комп'ютерне моделювання, варіативні моделі, інформаційна система, AI, КОМСДН

Література

1. Hrybiuk O. Experience in Implementing Computer-Oriented Methodological Systems of Natural Science and Mathematics Research Learning in Ukrainian Educational Institutions. In: Machado J., Soares F. (eds) *Innovations in Mechatronics Engineering. Lecture Notes in Mechanical Engineering*, 2022. P. 55-68. Springer, Cham Online.
2. Hrybiuk O. Problems of expert evaluation in terms of the use of variative models of a computer-oriented learning environment of mathematical and natural science disciplines in schools, [w:] *Zeszyty Naukowe Politechniki Poznańskiej. Seria: Organizacja i Zarządzanie*, Zeszyt Nr 79, Poznań: Wydawnictwo Politechniki Poznaskiej (WPP), 2019. P. 101-119.
3. Гриб'юк О.О. Імерсивні технології в освіті: особливості когнітивного розвитку дитини у віртуальному середовищі в процесі дослідницького навчання. *Сучасні інформаційні технології та інноваційні методики навчання в підготовці фахівців: методологія, теорія, досвід, проблеми: збірник наукових праць*. Київ-Вінниця: ТОВ Фірма «Планер», 2020.
4. Гриб'юк О.О. Комп'ютерне моделювання та робототехніка в навчально-виховному процесі сучасного навчального закладу. // *Сьома міжнародна науково-практична конференція FOSS Lviv-2017: Збірник наукових праць*, 27-30 квітня 2017 року, м. Львів. С. 38-43. ISBN 978-966-2598-86-5.
5. Hrybiuk Olena. Engineering in Educational Institutions: Standards for Arduino Robots as an Opportunity to Occupy an Important Niche in Educational Robotics in the Context of Manufacturing 4.0, in: *Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*. Volume 27-32, 2020. P. 770-785.
6. Hrybiuk O., Vedishcheva O. Experimental Teaching of Robotics in the Context of Manufacturing 4.0: Effective Use of Modules of the Model Program of Environmental Research Teaching in the Working Process of the Centers "Clever". In: , et al. *Innovations in Mechatronics Engineering II. icieng 2022. Lecture Notes in Mechanical Engineering*. Springer, Cham. 2022.
7. Hrybiuk O. Improvement of the Educational Process by the Creation of Centers for Intellectual Development and Scientific and Technical Creativity. In: Hamrol A., Kujawińska A., Barraza M. (eds). *Advances in Manufacturing II. MANUFACTURING 2019. Lecture Notes in Mechanical Engineering*, 2019. P. 370-382. Springer, Cham Online.

Бесараба Є. В (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О
*Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-
технічний університет імені академіка Юрія Бугая» (Київ)*

МАТЕМАТИЧНІ МЕТОДИ ТА АЛГОРИТМІЧНІ ЗАСОБИ МОДЕЛЮВАННЯ ІНТЕЛЕКТУАЛЬНОЇ ПОВЕДІНКИ В РЕАЛІСТИЧНИХ СТРАТЕГІЧНИХ ІГРОВИХ СИСТЕМАХ

Створення інтелектуальних систем управління для багатоагентних автономних середовищ – це один із найбільш пріоритетних викликів у сучасній індустрії розробки ігор. Традиційні підходи, що базуються на жорсткому кодуванні (scripted AI), стрімко поступаються місцем системам з адаптивною логікою, і проєкт «Last Bastion» зосереджений саме на розробці такої програмної архітектури. Сучасні RTS-ігри вимагають від агентів не лише здатності оцінювати загрози та приймати тактичні рішення в реальному часі, а й робити це в умовах жорстких апаратних обмежень.

Проблема «вузького місця процесора» (CPU Bottleneck) при обробці масових сцен робить критично актуальним пошук нових архітектурних рішень. Використання ієрархічних скінченних автоматів (HFSM), математичних моделей предиктивного наведення та компонентно-орієнтованого програмування (COP) у середовищі Unity 6 дозволяє досягти глибокої тактичної взаємодії при суворій оптимізації обчислювальних ресурсів.

Мета дослідження полягає у проєктуванні та програмній реалізації модульної ігрової системи «Last Bastion», де головний акцент зроблено на тактичній взаємодії автономних ботів, достовірній балістиці та архітектурній оптимізації в реальному часі. Програмна реалізація підсистеми управління у середовищі Unity 6 побудована на взаємодії компонентів введення-виведення з навігаційним рушієм (NavMesh). Основною метою цього етапу було створення

безперебійного конвеєра передачі команд від гравця до автономних бойових одиниць із забезпеченням наочного візуального зворотного зв'язку.

Оскільки RTS-ігри функціонують у парадигмі непрямого контролю, гравцеві необхідно чітко розуміти поточний намір агента. Для цього був розроблений скрипт `PathRenderer.cs` (Додаток Г, Лістинг Г.4), який відповідає за візуалізацію навігаційного сплайну. Для забезпечення стабільності архітектури скрипт використовує атрибут `[RequireComponent(typeof(LineRenderer))]`, що застраховує від помилок відсутності компонента на ігровому об'єкті. У методі `LateUpdate()` (який викликається після завершення всіх розрахунків фізики та навігації в кадрі) алгоритм перевіряє статус агента `agent.hasPath`. Якщо маршрут існує, викликається метод `DrawPath()`, який: Зчитує масив точок повороту маршруту `path.corners` із внутрішньої системи `NavMesh`. Задає необхідну кількість вершин для компонента `LineRenderer` (`line.positionCount = path.corners.Length`). У циклі копіює точки маршруту, примусово додаючи зміщення по осі Y (`point.y += 0.4f`). Це критичне архітектурне рішення, яке запобігає виникненню графічного артефакту «Z-fighting» (мерехтіння текстур), коли лінія шляху зливається з геометрією ландшафту.

Реалізація бойових одиниць у середовищі Unity 6 відбувалася з суворим дотриманням принципів компонентно-орієнтованого програмування (COP). Відповідно до спроектованої у другому розділі архітектури, було прийнято рішення відмовитися від створення єдиного глобального скрипта-моноліту (уникнення антипатерну «Божественний об'єкт»). Натомість логіка ігрового юніта (танка) була глибоко декомпозована на набір ізольованих класів-компонентів, кожен з яких імплементується як окремий об'єкт `MonoBehaviour` і виконує виключно свою вузьку задачу. На практиці такий підхід дозволив забезпечити слабку зв'язність (Loose Coupling) між ігровими системами. Компоненти практично не мають жорстких залежностей один від одного, а взаємодіють через

визначені C#-інтерфейси (наприклад, для отримання шкоди) та події. Це значно спростило процес тестування і надало можливість швидкої (plug-and-play) заміни модулів без ризику виникнення критичних каскадних помилок у суміжних ділянках програмного коду.

Для обробки життєвих показників та ідентифікації фракційної приналежності об'єктів було розроблено базовий компонент Health.cs (програмний код наведено у Додатку Г, Лістинг Г.3). Відповідно до принципу єдиної відповідальності (SRP), цей скрипт повністю ізолює логіку отримання шкоди від інших підсистем (таких як навігація чи штучний інтелект). Метод TakeDamage(float amount) приймає значення отриманої шкоди, віднімає його від змінної currentHealth та проводить перевірку на життєздатність об'єкта. У разі досягнення нульової позначки система автоматично викликає метод Die(), який коректно очищає ігрову сцену від знищеного об'єкта за допомогою базової функції Destroy(gameObject). Також компонент містить булевий маркер isEnemy, який використовується снарядами (через перевірку isEnemyProjectile == targetHealth.isEnemy) та ШІ для розпізнавання цілей.

Візуалізація стану здоров'я бойових одиниць реалізована через спеціалізований скрипт HealthBarUI.cs. Оскільки ігрова камера у стратегіях вільно переміщується у 3D-просторі, стаціонарне розміщення 2D-інтерфейсу (World Space Canvas) над танком призвело б до його графічного спотворення або повної невидимості при зміні кута огляду об'єктива (рис. 1). Для вирішення цього інженерного завдання імплементовано математичний алгоритм динамічного білбордингу (Billboarding).



Рис. 1. Візуалізація роботи скрипта HealthBarUI
Джерело: опрацювання власне

Для реалізації дата-орієнтованого підходу (Data-Driven Design) та забезпечення гнучкості налаштування ігрового балансу, архітектура модулів «Final Bastion» активно використовує технологію ScriptableObjects. Програмне впровадження цього патерну полягає у створенні спеціалізованих класів-контейнерів, які зберігають статичні параметри та існують незалежно від конкретних об'єктів на ігровій сцені. У проєкті створено клас конфігурації (наприклад, UnitStatsData), успадкований від ScriptableObject та позначений атрибутом [CreateAssetMenu]. Цей атрибут дозволяє розробнику або геймдизайнеру генерувати файли-асети конфігурації безпосередньо через графічний інтерфейс Unity Editor, не втручаючись у програмний код. Кожен такий асет-контейнер надійно інкапсулює критичні технічні константи типу бойової одиниці: базовий рівень здоров'я (maxHealth), швидкість навігації, радіус виявлення цілей (detectRange) та параметри стрільби (fireRate, damage).

Процес ініціалізації цих параметрів відбувається на старті ігрового циклу (Awake або Start). Логічні компоненти (такі як TurretController або Health) звертаються до призначеного їм асета ScriptableObject і зчитують необхідні константи. Такий підхід забезпечує системі дві фундаментальні архітектурні

переваги. Фундаментальним завданням при розробці супротивників у «Final Bastion» було створення непередбачуваної, але логічно обґрунтованої поведінки агентів. Для уникнення монолітної логіки систему було розбито на сенсорні підсистеми збору просторових даних та незалежний модуль прийняття рішень на базі ієрархічного скінченного автомата HFSM.

Програмна імплементація штучного інтелекту ворожих агентів зосереджена у скрипті AlienAI.cs. На рівні C#-коду архітектура HFSM реалізована через систему строгих перерахунків (enum): MacroState для визначення глобальних станів (Peaceful, Combat, Emergency) та вкладені PeacefulState і CombatState для обробки специфічних мікрозадач. Головний цикл обробки штучного інтелекту відбувається у системному методі Update(), який щокадру викликає дві ключові функції, що розділяють логіку аналізу та дії.

Важливим архітектурним рішенням стала імплементація підстану здивування (CombatState.Surprised). При першому виявленні гравця чи отриманні шкоди бот не починає миттєву атаку чи рух, а різко зупиняється (agent.ResetPath()) та ініціює таймер reactionTimer, який за замовчуванням встановлений на 0,6 секунди. Це програмно імітує людський фактор когнітивної затримки екіпажу, роблячи поведінку ІШ більш природною і даючи гравцеві невелике вікно для тактичної реакції.

Для забезпечення адекватної реакції агентів на динамічне середовище було реалізовано комплексну сенсорну підсистему. Функція зору агента імплементована у методі CheckLineOfSight(). Програмний алгоритм працює у два послідовні етапи: *Широка фаза (Кут огляду)*: Спочатку розраховується кут між вектором погляду та напрямком на гравця за допомогою векторної алгебри (Vector3.Angle). Якщо ціль не потрапляє у заданий конус огляду (fieldOfViewAngle = 110f), перевірка миттєво переривається для економії обчислювальних ресурсів процесора. *Вузька фаза (Лінія видимості)*: Якщо ціль

знаходиться у конусі видимості, система пускає математичний промінь `Physics.Raycast`, перевіряючи відсутність фізичних перешкод на заданому шарі `obstacleMask`.

Групова координація ворогів реалізована через метод `AlertNearbyAllies()`. При виявленні загрози бот сканує простір запитом `Physics.OverlapSphere` у радіусі `alertRadius` і примусово переводить усіх знайдених союзників (`allyAI != null`) у стан бойової готовності, встановлюючи прапорець `isAlerted = true`. Це провокує ланцюгову реакцію, формуючи ефект скоординованої хвильової атаки.

Верифікація поведінки III проводилася методом функціонального тестування безпосередньо в ігровому середовищі Unity 6. Результати перевірки підтверджують, що переходи між станами спокою та атаки, а також алгоритми обходу динамічних перешкод юнітами функціонують коректно і відповідають заданим вимогам. Завершальним етапом програмної реалізації модулів «Final Bastion» стала розробка бойових систем, здатних до автономного функціонування в динамічному 3D-середовищі. Основна складність полягала у забезпеченні високої влучності вогню по цілях, що постійно змінюють вектор свого руху, при збереженні низького навантаження на центральний процесор.

Програмна логіка наведення та балістичних розрахунків імплементована у класі `TurretController.cs`. Замість ресурсоємної симуляції польоту снаряда, система використовує математичну модель розрахунку точки зустрічі, засновану на векторній алгебрі. Для забезпечення візуальної плавності роботи механізмів повороту башти використано метод `RotateTowardsTarget()`, де застосовується сферична лінійна інтерполяція `Quaternion.Slerp`. Це дозволяє уникнути миттєвих «стрибків» повороту та забезпечує стабільне супроводження цілі зі швидкістю, заданою параметром `rotationSpeed`.

Верифікація точності автономного вогню підтвердила ефективність розробленої математичної моделі. Згідно з протоколом тестування ігрових систем

(рис. 2), алгоритм предикції успішно компенсує швидкість руху цілей, а система Raycast-балістики забезпечує абсолютну точність реєстрації влучань навіть при максимальних показниках швидкості снаряда bulletSpeed.

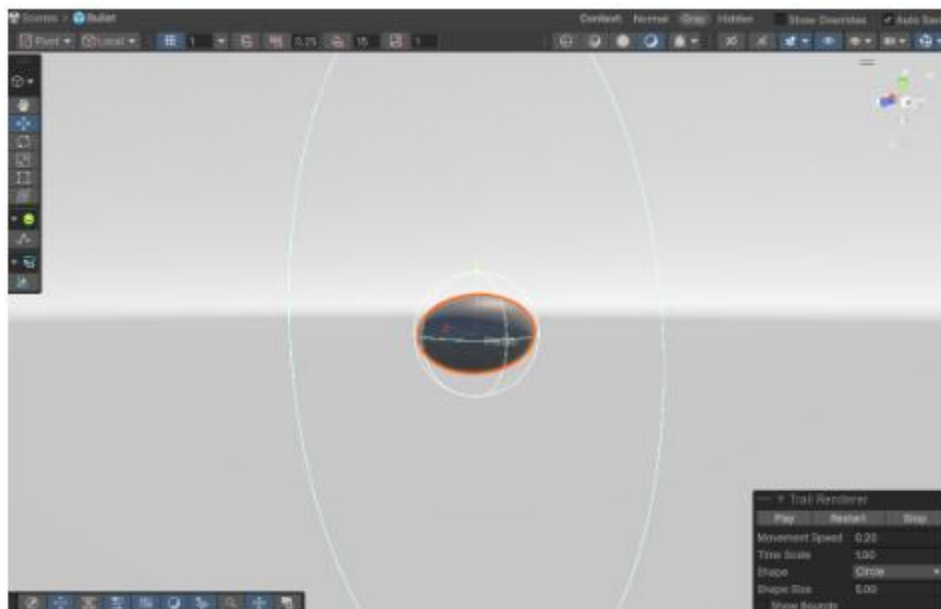


Рис. 2. Конфігурація параметрів фізики та шкоди балістичного снаряда
Джерело: опрацювання власне

Автономна поведінка супротивників – це вже парафія ієрархічного скінченного автомата у скрипті AlienAI. Завдяки йому боти навчилися адекватно реагувати на загрози, вчасно відступати та координувати хвильові атаки, спираючись на оптимізовану двоступеневу сенсорну систему. Також у бойовому модулі TurretController ми успішно реалізували алгоритм предиктивного наведення башти на основі векторної алгебри. Важливо, що ми свідомо відмовилися від «важких» фізичних компонентів Rigidbody на користь математичної балістики через Raycast у скрипті Projectile. Це дозволило повністю вирішити критичну проблему «квантового тунелювання» і, що важливо, відчутно розвантажити центральний процесор.

Проведене функціональне тестування підтвердило: синтез компонентної архітектури, предиктивного наведення та дата-орієнтованих патернів дійсно працює. Система видає стабільну частоту кадрів (FPS) та забезпечує високу тактичну глибину. Вдалося уникнути тих самих обчислювальних «вузьких місць» (CPU Bottleneck), які зазвичай стають вироком для масових RTS-систем.

Анотація. Розроблення ігрової системи «Final Bastion» передбачає не лише створення ігрової сцени, а й проєктування комплексної системи, у якій штучний інтелект, балістика, навігація та модульна архітектура взаємодіють у межах єдиного програмного середовища. Саме тому подальший аналіз зосереджується на теоретичних підходах до автономної поведінки агентів і побудови ефективних ігрових систем. Архітектура багатоагентної системи «Final Bastion» нарешті перейшла з теоретичних моделей у реальний функціональний код середовища Unity 6. Власне, цей практичний етап лише підтвердив: обрані нами підходи до оптимізації та компонентної розробки цілком життєздатні. З'явилася підсистема управління на базі контролера SelectionManager. Він вкрай ефективно транслює 2D-координати екрана у 3D-простір. Модулі реєстрації пошкоджень (Health) та динамічного білбордингу (HealthBarUI) тепер повністю ізольовані один від одного. Варто зазначити, що використання патерну ScriptableObjects для конфігурації технічних параметрів дало нам не тільки бажану гнучкість у налаштуванні балансу за парадигмою Data-Driven Design, а й допомогло суттєво зекономити ресурси оперативної пам'яті. У процесі обчислення повороту за віссю Y система примусово ігнорує вертикальну складову вектора, що гарантує горизонтальну стабільність бапти при наведенні на об'єкти, що знаходяться на різній висоті ландшафту.

Ключові слова: геймдизайн орієнтований підхід, дослідницьке навчання, варіативні моделі, комп'ютерне моделювання, КОМСДН, ігрова система, AI, інтелектуальна система, ігровий інтерфейс, математичне моделювання, стратегія, штучний інтелект, Final Bastion.

Література

1. Hrybiuk O. Improvement of the Educational Process by the Creation of Centers for Intellectual Development and Scientific and Technical Creativity. In: Hamrol A., Kujawińska A., Barraza M. (eds) *Advances in Manufacturing II. MANUFACTURING 2019. Lecture Notes in Mechanical Engineering*, 2019. P. 370-382. Springer, Cham Online.
2. Hrybiuk Olena. Engineering in Educational Institutions: Standards for Arduino Robots as an Opportunity to Occupy an Important Niche in Educational Robotics in the Context of Manufacturing 4.0, in: *Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*. Volume 27-32, 2020. P. 770-785.

3. Hrybiuk O. Problems of expert evaluation in terms of the use of variative models of a computer-oriented learning environment of mathematical and natural science disciplines in schools, [w:] *Zeszyty Naukowe Politechniki Poznańskiej. Seria: Organizacja i Zarządzanie*, Zeszyt Nr 79, Poznań: Wydawnictwo Politechniki Poznańskiej (WPP), 2019. P. 101-119.
4. Гриб'юк О.О. *Дослідницьке навчання учнів предметів природничо-математичного циклу з використанням комп'ютерно орієнтованих методичних систем* / О. О. Гриб'юк. Монографія. Київ: НПУ імені М. П. Драгоманова, 2019.
5. Hrybiuk O., Vedishcheva O. Experimental Teaching of Robotics in the Context of Manufacturing 4.0: Effective Use of Modules of the Model Program of Environmental Research Teaching in the Working Process of the Centers "Clever". In: , et al. *Innovations in Mechatronics Engineering II. 2022. Lecture Notes in Mechanical Engineering*. Springer, Cham. 2022.
6. Hrybiuk O. Experience in Implementing Computer-Oriented Methodological Systems of Natural Science and Mathematics Research Learning in Ukrainian Educational Institutions. In: Machado J., Soares F. (eds) *Innovations in Mechatronics Engineering. Lecture Notes in Mechanical Engineering*, 2022. P. 55-68. Springer, Cham Online.
7. Гриб'юк О.О. Комп'ютерне моделювання та робототехніка в навчально-виховному процесі сучасного навчального закладу. / Гриб'юк О.О. // *Сьома міжнародна науково-практична конференція FOSS Lviv-2017: Збірник наукових праць*, 27-30 квітня 2017 року, м. Львів. С. 38-43. ISBN 978-966-2598-86-5.

УДК 004.942:[327

Білозуб М. М. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О
Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-технічний університет імені академіка Юрія Бугая» (Київ)

РОЗРОБЛЕННЯ МАТЕМАТИЧНОЇ СТРУКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ СТВОРЕННЯ І ВІДСТЕЖЕННЯ ФІНАНСОВИХ ВИТРАТ

Актуальність дослідження. Сучасні користувачі щодня здійснюють десятки фінансових операцій у різних каналах: готівкових розрахунках, платіжних картках, онлайн-сервісах. Без систематизованого інструменту створення і відстеження фінансових витрат складно оцінити реальну структуру витрат, спланувати бюджет і контролювати виконання фінансових цілей [1]. Існуючі рішення або орієнтовані на професійний бухгалтерський чи податковий облік, або не враховують мотиваційні механізми, що підтримують регулярне внесення даних. Тому виникає потреба у простій веб-орієнтованій інформаційній системі, яка

поєднує створення і відстеження фінансових витрат з базовою аналітикою та елементами гейміфікації.

Мета дослідження полягає у проектуванні та розробленні інформаційної системи для створення і відстеження фінансових витрат, яка забезпечує просте введення операцій, аналіз витрат та підтримку фінансових цілей користувача.

У рамках дослідження розроблено веб-орієнтовану інформаційну систему для створення і відстеження фінансових витрат користувачів. У системі поєднано ручне введення операцій, агреговану аналітику за різні періоди та мотиваційні механізми на основі фінансових цілей і досягнень, ґрунтовно вивчено проблеми добору архітектури (рис. 1), інструментів і необхідних алгоритмів із метою здійснення проектування і розроблення інформаційної системи (ІС) для створення і відстеження фінансових витрат.

Запропоновано структурований набір функціональних модулів клієнтської і серверної частин, інформаційну модель добору та аналізу даних, відповідний набір детальних API, орієнтованих на реалізацію мінімально життєздатного продукту. Описано алгоритми розрахунку рекомендованих відкладень, побудови аналітичних зрізів і нарахування досягнень, а також сформульовано нефункціональні вимоги щодо безпеки, продуктивності, надійності та юзабіліті. Сформовано критерії приймання, що дозволяють підтвердити відповідність розробленого прототипу поставленим вимогам.

Наукова новизна роботи полягає в поєднанні гейміфікації, моделі фінансових цілей і чітко визначеного API в одному мінімальному, але розширювальному рішенні для персонального фінансового обліку. Практична значущість полягає в наданні реалістичної до впровадження специфікації веб-додатку, який може бути використаний як самостійний сервіс або як частина більших фінтех-рішень. Система реалізується за клієнт-серверною архітектурою [2]. Клієнтська частина відповідає за інтерфейс користувача, візуалізацію

аналітики та валідацію даних на стороні браузера. Серверна частина реалізує бізнес-логіку, керування даними, аналітичні обчислення та інтегровані планувальники для формування звітів.

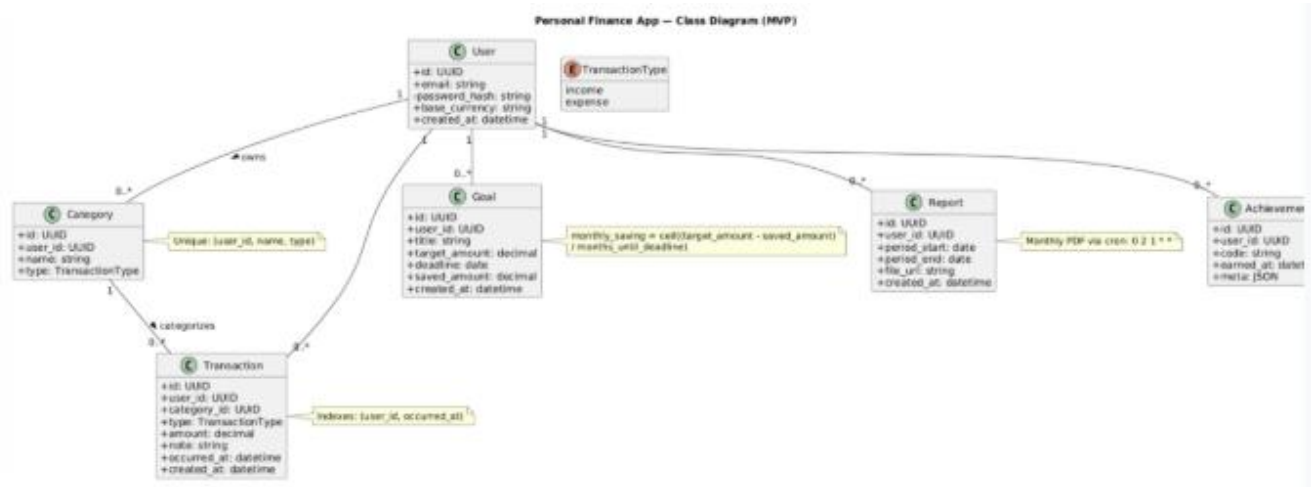


Рис. 1. Фрагмент ескізного проєкту ІС для створення і відстеження фінансових витрат
Джерело: опрацювання власне

Розвиток системи здійснювався поетапно. На першому етапі (MVP) реалізовано базовий функціонал: аутентифікація, керування категоріями й транзакціями, базова аналітика, експорт CSV, формування щомісячних PDF-звітів, мінімальний набір фінансових цілей і досягнень. На другому етапі додано бюджети за категоріями, розширена аналітика, локалізація інтерфейсу та режим PWA. Третій етап передбачав здійснення імпорту CSV із банківських виписок, інтеграцію системи нагадувань, підтримки багатовалютності та автокатегоризацію операцій.

У контексті здійснення коректної монетизації розглядається три рівні підписки [3]. Рівень Free забезпечує базовий функціонал: ручне введення транзакцій, обмежену кількість рахунків і бюджетів, базові звіти, одну фінансову ціль і базові бейджі. Рівень Plus додає необмежені категорії й бюджети, експорт CSV/PDF, кілька фінансових цілей, нагадування та розширений набір бейджів. Рівень Pro передбачає розширені звіти, автоматичний щомісячний PDF-звіт,

розширені фільтри пошуку, модуль підказок щодо відкладень, а також, за потреби, багатовалютність і автокатегоризацію.

Основними ризиками реалізації проєкту є обмежений час розроблення, вимоги до конфіденційності фінансових даних та можливі помилки введення з боку користувачів. Для зменшення впливу цих ризиків передбачено пріоритизацію функцій MVP, використання сучасних засобів захисту даних і продуману валідацію форм введення [4]. Вважається, що користувачі готові регулярно вносити дані про свої витрати та доходи, а також мають доступ до стабільного Інтернет-з'єднання й сучасних браузерів. За цих умов система забезпечує якісне створення і відстеження фінансових витрат та досягнення поставлених фінансових цілей.

Анотація. У рамках дослідження здійснено проєктування і розроблено інформаційну систему для створення і відстеження фінансових витрат, яка забезпечує просте введення операцій, аналіз витрат та підтримку фінансових цілей користувача. Сформовано та деталізовано концепцію веб-додатку як інформаційної системи для створення і відстеження фінансових витрат користувача. Запропоновано структуру клієнтських і серверних модулів, інформаційну модель даних, алгоритми побудови аналітики, розрахунку рекомендованих відкладень і нарахування досягнень. Сформульовані функціональні та нефункціональні вимоги, цільові платформи та критерії приймання задають чіткі рамки для реалізації й подальшого розвитку мінімально життєздатного продукту, що підвищує якість персонального фінансового обліку та ймовірність досягнення користувачем власних фінансових цілей. Наукова новизна роботи полягає в поєднанні гейміфікації, моделі фінансових цілей і чітко визначеного API в одному мінімальному, але обгрунтованому рішенні для персонального фінансового обліку. Практична значущість полягає в наданні реалістичної до впровадження специфікації веб-додатку, який може бути використаний як самостійний сервіс або як частина більших фінансово-технічного рішення.

Ключові слова: дослідницьке навчання, варіативні моделі, математичне моделювання, фінансові витрати, комп'ютерне моделювання, КОМСДН, інформаційна система, інтелектуальний аналіз даних, CleverCOMSRL, AI.

Література

1. Hrybiuk O., Kant G.S. CleverCOMSRL: Implementation of an AI Computer-Aided Design System in the Context of the Cognitive Science Paradigm for the Research Training Process. In: Machado, J., et al. *Innovations in Mechatronics Engineering III*. icieng 2024. *Lecture Notes in*

Mechanical Engineering. Springer, Cham. 2024. P. 351-362. https://doi.org/10.1007/978-3-031-61575-7_32.

2. Hrybiuk O. Improvement of the educational process by the creation of centers for intellectual development and scientific and technical creativity. In: Hamrol, A., Kujawińska, A., Barraza, M.F.S. (eds.) *MANUFACTURING 2019*. LNME, 2019. P. 370–382. Springer, Cham. https://doi.org/10.1007/978-3-030-18789-7_31.

3. Hrybiuk O.O. Research learning of the natural science and mathematics cycle using computer-oriented methodological systems. Monograph. Drahomanov NPU, Kyiv, 2019. P. 127–135.

4. Hrybiuk O. Experience in Implementing Computer-Oriented Methodological Systems of Natural Science and Mathematics Research Learning in Ukrainian Educational Institutions. In: Machado J., Soares F. (eds) *Innovations in Mechatronics Engineering. Lecture Notes in Mechanical Engineering*, 2022. P. 55-68. Springer, Cham Online.

УДК 004.942:[327

Білошицький Н. Д. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О
Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-технічний університет імені академіка Юрія Бугая» (Київ)

ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ “РОЗКЛАД ЗАНЯТЬ ДЛЯ СТУДЕНТІВ” ІЗ ВИКОРИСТАННЯМ МАТЕМАТИЧНИХ МЕТОДІВ ТЕХНОЛОГІЇ ОБ’ЄКТНО ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ

Сучасні освітні заклади активно впроваджують інформаційні технології в організацію навчального процесу. Одним із ключових напрямів цифровізації є автоматизація процесу формування, зберігання та відображення розкладу занять. Традиційні методи, що передбачають розміщення розкладу у вигляді таблиць на сайтах чи стендах, є малоефективними, адже не забезпечують зручного доступу, швидкого оновлення та персоналізації для кожного користувача.

Інформаційні програмні системи є основою сучасних цифрових сервісів і призначені для збору, оброблення, зберігання та представлення даних у зручному форматі. Відповідно до міжнародного стандарту ISO/IEC 2382, інформаційна система - це система, яка підтримує прийняття рішень [1]. У контексті освітніх установ такі системи забезпечують студентів та викладачів актуальними даними

про навчальний процес, включаючи розклад занять, події, нагадування та інші параметри навчальної діяльності.

Актуальність створення інформаційної системи “Розклад занять” пояснюється тим, що на практиці розклад занять у багатьох університетах подається у вигляді статичних документів, що публікуються на сайті. Це створює низку труднощів: такі документи не адаптуються під мобільні пристрої, не можуть автоматично оновлюватися, не підтримують вибір групи та підгрупи, не зберігають налаштування користувача та не мають інтегрованої системи сповіщень [2]. Розклад може змінюватися декілька разів на місяць, і студенти змушені кожного разу вручну перевіряти зміни. Наявна проблема прямо обґрунтовує необхідність створення окремого застосунку з автоматизацією всіх цих процесів.

Мета дослідження полягає у розробленні інформаційної програмної системи “Розклад занять” із використанням мови програмування Python і бібліотеки Flet для створення графічного інтерфейсу. *Завдання:* 1. Здійснити ґрунтовний аналіз програмного забезпечення, існуючих застосунків [3], виокремивши переваги та недоліки систем; добір технологій та інструментальних засобів для розроблення інформаційної системи “Розклад занять”. 2. Розробити концепцію інформаційної системи з урахуванням сучасних вимог до інтерфейсу та зручності користування. 3. Здійснити проєктування архітектури інформаційної системи. 4. Реалізувати модулі інформаційної системи “Розклад занять” із використанням технології об’єктно орієнтованого програмування. 5. Забезпечити перевірку коректності функціонування системи, передбачивши можливість розширення модулів.

Розроблення інформаційної системи “Розклад занять” базується на сучасних принципах архітектури програмного забезпечення [4]. У застосунку використовується модульний підхід: окремо працюють модуль для парсингу розкладу, модуль інтерфейсу, модуль бази даних, модуль оброблення PDF та

DOCX документів, календарний модуль та модуль сповіщень. У рамках дослідження програмна архітектура чітко розділена. Наприклад, збереження налаштувань користувача реалізоване в окремій функції (рис. 1).

```
def save_settings_to_db(notifications_enabled: bool, notify_minutes: int, notify_group: str):  
    conn = sqlite3.connect(DB_FILE)  
    c = conn.cursor()  
    c.execute(  
        "UPDATE settings SET notifications_enabled = ?, notify_minutes = ?, notify_group = ? WHERE id = 1",  
        (1 if notifications_enabled else 0, notify_minutes, notify_group)  
    )  
    conn.commit()  
    conn.close()
```

Рис. 1. Демонстрація застосування принципів модульності та розділення відповідальностей
Джерело: опрацювання власне

Однією з ключових теоретичних задач є автоматичне отримання та аналіз розкладу, що поданий на сайті у вигляді HTML-сторінок та файлів різних форматів [5]. У програмі використовується офіційно рекомендована зв'язка інструментів requests та BeautifulSoup для веб-скрейпінгу: Оскільки розклад може бути у форматі PDF, у системі реалізовано теоретично правильний і технічно підтверджений метод оброблення PDF через бібліотеку PyMuPDF (fitz). Для збереження налаштувань користувача застосовано локальну базу SQLite, яка є оптимальним рішенням для офлайн застосунків [6]. Календар у застосунку побудований на основі стандартного модуля calendar із Python, який реалізує загальновідомі алгоритми побудови календарної матриці. Графічний інтерфейс створено за допомогою Flet - фреймворку, що реалізує декларативну модель UI, подібну до Google Flutter. Flet забезпечує виділення компонентів, контейнерну структуру та адаптивну побудову інтерфейсу [7]. Окремим елементом системи є механізм сповіщень про початок пар.

У рамках дослідження здійснено автоматичне отримання і парсинг даних із сайтів та забезпечується збереження розкладу у локальній базі SQLite,

передбачається можливість добору навчальної групи та тижня (парний/непарний), у т.ч. забезпечується використання календаря із вибором дати, push-сповіщення про початок пари (за 5–20 хв.), перегляд новин університету та інтеграція з Telegram-ботом для дублювання повідомлень.

Анотація. Актуальність теми полягає у необхідності створення зручної інформаційної системи, яка дозволяє студентам і викладачам швидко отримувати актуальний розклад, переглядати зміни, а також отримувати сповіщення про початок занять. Такий підхід не лише підвищує ефективність організації навчального процесу, а й сприяє дисципліні та плануванню часу. Розроблений застосунок є кросплатформним, інтуїтивно зрозумілим і простим у використанні, адаптивним до різних екранів і може бути застосований як допоміжний інструмент для організації навчального процесу в закладах освіти різного рівня. Інформаційна система забезпечує інтеграцію з базою даних SQLite, автоматичний парсинг розкладу з офіційного сайту університету, збереження даних на пристрої користувача.

Ключові слова: об'єктно орієнтоване програмування, дослідницьке навчання, варіативні моделі, комп'ютерне моделювання, КОМСДН, інформаційна система, планування освітнього процесу, математичне моделювання, AI.

Література

1. Hrybiuk Olena. Engineering in Educational Institutions: Standards for Arduino Robots as an Opportunity to Occupy an Important Niche in Educational Robotics in the Context of Manufacturing 4.0, in: *Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*. Volume 27-32, 2020. P. 770-785.
2. Hrybiuk O. Problems of expert evaluation in terms of the use of variative models of a computer-oriented learning environment of mathematical and natural science disciplines in schools, [w:] *Zeszyty Naukowe Politechniki Poznańskiej. Seria: Organizacja i Zarządzanie*, Zeszyt Nr 79, Poznań: Wydawnictwo Politechniki Poznańskiej (WPP), 2019. P. 101-119.
3. Гриб'юк О.О. Дослідницьке навчання учнів предметів природничо-математичного циклу з використанням комп'ютерно орієнтованих методичних систем / О. О. Гриб'юк. Монографія. Київ: НПУ імені М. П. Драгоманова, 2019.
4. Hrybiuk O., Vedishcheva O. Experimental Teaching of Robotics in the Context of Manufacturing 4.0: Effective Use of Modules of the Model Program of Environmental Research Teaching in the Working Process of the Centers “Clever”. In: , et al. *Innovations in Mechatronics Engineering II. icieng 2022. Lecture Notes in Mechanical Engineering*. Springer, Cham. 2022.
5. Hrybiuk O. Experience in Implementing Computer-Oriented Methodological Systems of Natural Science and Mathematics Research Learning in Ukrainian Educational Institutions. In: Machado J., Soares F. (eds) *Innovations in Mechatronics Engineering. Lecture Notes in Mechanical Engineering*, 2022. P. 55-68. Springer, Cham Online.
6. Гриб'юк О.О. Комп'ютерне моделювання та робототехніка в навчально-виховному процесі сучасного навчального закладу. / Гриб'юк О.О. // *Сьома міжнародна*

науково-практична конференція FOSS Lviv-2017: Збірник наукових праць, 27-30 квітня 2017 року, м. Львів. С. 38-43. ISBN 978-966-2598-86-5.

7. Hrybiuk O. Improvement of the Educational Process by the Creation of Centers for Intellectual Development and Scientific and Technical Creativity. In: Hamrol A., Kujawińska A., Barraza M. (eds) *Advances in Manufacturing II. MANUFACTURING 2019. Lecture Notes in Mechanical Engineering*, 2019. P. 370-382. Springer, Cham Online.

УДК 004.942:[327

Бондаренко М. В. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гріб'юк О. О
Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-технічний університет імені академіка Юрія Бугая» (Київ)

МАТЕМАТИЧНЕ ПІДГРУНТЯ ПРОЕКТУВАННЯ ГРАФІЧНОГО КАЛЬКУЛЯТОРА ДЛЯ ОБЧИСЛЕННЯ ОБ'ЄМІВ І ПЛОЩ ПОВЕРХОНЬ ГЕОМЕТРИЧНИХ ТІЛ

Актуальність розроблення графічного калькулятора полягає у поєднанні освітньої, наукової та дослідницької компонент. З одного боку, він є дидактичним інструментом, який допомагає студентам краще розуміти зв'язок між формулами і геометричними моделями. З іншого боку, він має прикладне значення у професійних задачах, де важливо швидко отримати результат з візуальним підтвердженням правильності. Одним із таких рішень є графічний калькулятор для обчислення об'ємів і площ поверхонь геометричних тіл, який не лише полегшує виконання обчислень, але й сприяє розвитку просторового мислення користувачів [1]. Він може бути використаний як навчальний засіб у школах і закладах вищої освіти, а також як допоміжний інструмент для інженерів, дизайнерів і дослідників, які мають справу з геометричними моделями.

Мета дослідження полягає у розробленні програмної системи, що забезпечує інтерактивне введення параметрів геометричних тіл, автоматичне обчислення їх площі та об'єму, а також графічне відображення результатів у зручній формі. *Завдання:* 1. Здійснити ґрунтовний аналіз інформаційного та

програмного забезпечення, існуючих застосунків на ринку, виокремивши переваги та недоліки систем; добір технологій та інструментальних засобів для розроблення програмної системи. 2. Розробити концепцію власного програмного продукту з урахуванням сучасних вимог до інтерфейсу та зручності користування. 3. Здійснити проєктування архітектури програмної системи на основі об'єктно-орієнтованого підходу. 4. Реалізувати програму у середовищі Python Builder із використанням графічного інтерфейсу та власних класів для геометричних фігур. 5. Забезпечити перевірку правильності введених даних та обробку можливих помилок користувача. Передбачити можливість розширення системи новими типами тіл або додатковими функціями. У процесі розроблення застосовувалися базові шаблони проєктування, які дозволяють організувати структуру програми відповідно до кращих практик розробки програмного забезпечення [2]. Зокрема, для створення об'єктів геометричних фігур використовується підхід, подібний до фабричного методу, що дозволяє централізувати логіку створення об'єктів і спростити додавання нових типів фігур у майбутньому. Для організації взаємодії між інтерфейсом користувача та обчислювальним модулем застосовується підхід, близький до шаблону Model-View-Controller.

Графічний калькулятор має модульну архітектуру, яка забезпечує розділення логіки обчислень, візуалізації та взаємодії з користувачем. Це дозволяє забезпечити зручну підтримку, масштабованість і можливість розширення функціоналу [3]. Основу системи складають класи, які представляють геометричні фігури, математичні операції, інтерфейсні компоненти та допоміжні модулі (рис. 1).

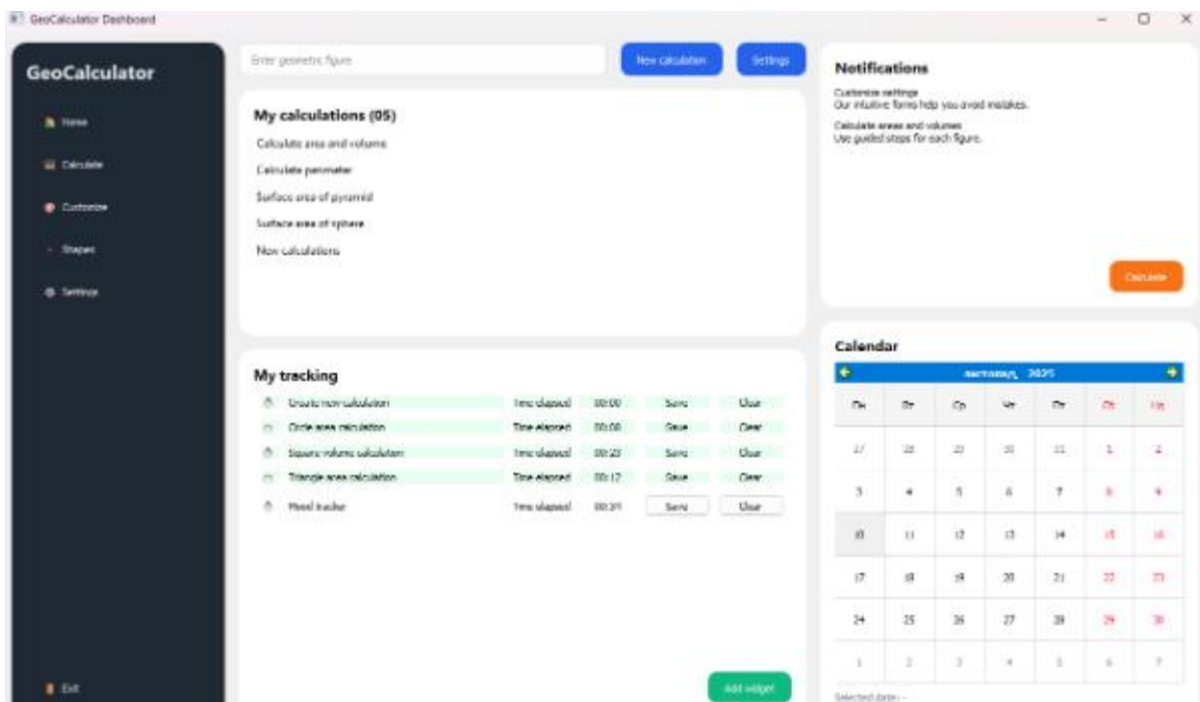


Рис. 1. Прототип екранної форми головної сторінки калькулятора
Джерело: опрацьовано власне

Взаємодія між компонентами системи організована таким чином, що інтерфейс користувача передає введені дані до обчислювального модуля, який, у свою чергу, виконує необхідні розрахунки та повертає результат для відображення. Модуль перевірки даних виступає проміжною ланкою, яка гарантує коректність обчислень. Основна увага приділялася забезпеченню інтуїтивності взаємодії, що дозволяє користувачеві швидко освоїти роботу з програмою [3]. Інтерфейс передбачає можливість вибору геометричного тіла з відповідного списку, після чого користувачеві пропонується ввести параметри, необхідні для обчислення. Поля введення динамічно змінюються залежно від обраної фігури, що дозволяє уникнути надлишкових елементів на екрані.

Результати обчислень відображаються у зрозумілому вигляді з відповідними поясненнями. Додатково передбачено повідомлення про помилки, які інформують користувача про некоректність введених даних. Реалізовано обчислення площі та об'єму геометричних фігур. Логіка обчислення зосереджена

в окремих класах фігур, що містяться у модулі calculate. Для кожного типу фігури реалізовано відповідні методи area() та/або volume() згідно з математичними формулами [4]. Відповідні обчислення реалізуються через відповідні методи класу Cylinder, які приймають радіус і висоту як вхідні параметри.

Інтерфейс введення параметрів: Інтерфейс побудований на основі діалогу NewCalculationDialog, який динамічно адаптується до вибраної фігури - відображає лише ті поля, які є необхідними для обчислення (рис. 2). Усі обчислення виконуються після попередньої перевірки введених значень. Якщо параметри не відповідають допустимим умовам, наприклад є від'ємними або рівними нулю у випадках, де це недопустимо, система не виконує розрахунок і повідомляє користувача про помилку.

Помилки також фіксуються через механізм логування (traceback) для розробника. Після успішного обчислення результати виводяться у вікні повідомлення з точністю до двох знаків після коми. Інтерфейс також може повідомити користувача про тип виконаного розрахунку [5].

У процесі тестування програмного продукту було використано підходи тестування. Тестування із використанням методу чорної скриньки здійснювалося з позиції кінцевого користувача в контексті забезпечення перевірки правильності роботи функціоналу без урахування внутрішньої структури програми [6]. У ході такого тестування перевірялися основні сценарії використання, коректність обчислень при різних вхідних даних, а також реакція системи на некоректні значення параметрів. Окремо перевірялися граничні значення параметрів, наприклад дуже малі або великі значення радіуса чи висоти, а також некоректні дані, такі як від'ємні числа або відсутність обов'язкових параметрів. Результати тестування показали, що система коректно реагує на всі передбачені сценарії, відображає правильні результати обчислень та інформує користувача про помилки введення.

```

10 class NewCalculationDialog(QDialog):
11     def __init__(self, parent=None):
12         super().__init__(parent)
13         self.result = None # Тут збережемо рядок з результатом
14         self.setWindowTitle("New Calculation")
15         self.setupUI()
16
17     def setupUI(self):
18         self.formLayout = QFormLayout()
19
20         # Вибір типу фігури
21         self.figureCombo = QComboBox()
22         self.figureCombo.addItem([
23             "Regular Polygon", "Regular Pyramid", "Regular Prism", "Parallelepiped"
24         ])
25         self.formLayout.addRow("Select figure:", self.figureCombo)
26
27         # Поля введення параметрів
28         self.param1LabelWidget = QLabel("Parameter 1:")
29         self.param1Edit = QLineEdit()
30         self.formLayout.addRow(self.param1LabelWidget, self.param1Edit)
31
32         self.param2LabelWidget = QLabel("Parameter 2:")
33         self.param2Edit = QLineEdit()
34         self.formLayout.addRow(self.param2LabelWidget, self.param2Edit)
35
36         self.param3LabelWidget = QLabel("Parameter 3:")
37         self.param3Edit = QLineEdit()
38         self.formLayout.addRow(self.param3LabelWidget, self.param3Edit)
39

```

Рис. 2. Фрагмент програмної реалізації класу NewCalculationDialog
Джерело: опрацьовано власне

Тестування методом білого ящика було спрямоване на аналіз внутрішньої логіки програми та перевірку правильності реалізації алгоритмів. У процесі тестування перевірялися окремі методи класів, зокрема функції обчислення об'єму та площі поверхні для різних геометричних тіл. Було виконано перевірку всіх можливих гілок виконання програми, включаючи обробку помилкових ситуацій [7]. Особлива увага приділялася правильності використання математичних формул, точності обчислень та обробці виключень. Проведене тестування підтвердило відсутність логічних помилок у реалізації алгоритмів, а також відповідність отриманих результатів очікуваним значенням. Усі перевірені методи продемонстрували стабільну роботу при різних наборах вхідних даних, що

свідчить про коректність їх програмної реалізації. Додатково було підтверджено, що обробка виключних ситуацій виконується належним чином і не призводить до порушення роботи системи в цілому. Отримані результати тестування дозволяють зробити висновок про надійність програмного коду та його готовність до практичного використання.

Анотація. У рамках дослідження ґрунтовно розроблено архітектуру та здійснено практичну реалізацію графічний калькулятор для обчислення об'ємів і площ поверхонь геометричних тіл. Мета дослідження полягає у розробленні програмної системи, що забезпечує інтерактивне введення параметрів геометричних тіл, автоматичне обчислення їх площі та об'єму, а також графічне відображення результатів у зручній формі. Було описано модулі, з яких складається система, зокрема: модуль введення параметрів, модуль обчислень, візуалізаційний блок, чат-помічник та механізми перевірки даних. Реалізовано механізми коректної обробки помилок, що забезпечує надійність системи. Вбудований чат-помічник підвищив доступність системи для користувача та спростить навігацію в інтерфейсі. Проведене тестування засвідчило стабільність роботи ПС, відповідність функціональним вимогам і показало, що система потребує доопрацювань.

Ключові слова: об'єктно орієнтоване програмування, дослідницьке навчання, варіативні моделі, математика, комп'ютерне моделювання, КОМСДН, інформаційна система, графічний калькулятор, AI.

Література

1. Hrybiuk Olena. Engineering in Educational Institutions: Standards for Arduino Robots as an Opportunity to Occupy an Important Niche in Educational Robotics in the Context of Manufacturing 4.0, in: *Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*. Volume 27-32, 2020. P. 770-785.
2. Hrybiuk O. Problems of expert evaluation in terms of the use of variative models of a computer-oriented learning environment of mathematical and natural science disciplines in schools, [w:] *Zeszyty Naukowe Politechniki Poznańskiej. Seria: Organizacja i Zarządzanie*, Zeszyt Nr 79, Poznań: Wydawnictwo Politechniki Poznańskiej (WPP), 2019. P. 101-119.
3. Гриб'юк О.О. Дослідницьке навчання учнів предметів природничо-математичного циклу з використанням комп'ютерно орієнтованих методичних систем / О. О. Гриб'юк. Монографія. Київ: НПУ імені М. П. Драгоманова, 2019.
4. Hrybiuk O., Vedishcheva O. Experimental Teaching of Robotics in the Context of Manufacturing 4.0: Effective Use of Modules of the Model Program of Environmental Research Teaching in the Working Process of the Centers "Clever". In: , et al. *Innovations in Mechatronics Engineering II. icieng 2022. Lecture Notes in Mechanical Engineering*. Springer, Cham. 2022.
5. Hrybiuk O. Experience in Implementing Computer-Oriented Methodological Systems of Natural Science and Mathematics Research Learning in Ukrainian Educational Institutions. In:

Machado J., Soares F. (eds) *Innovations in Mechatronics Engineering. Lecture Notes in Mechanical Engineering*, 2022. P. 55-68. Springer, Cham Online.

6. Гриб'юк О.О. Комп'ютерне моделювання та робототехніка в навчально-виховному процесі сучасного навчального закладу. / Гриб'юк О.О. // *Сьома міжнародна науково-практична конференція FOSS Lviv-2017: Збірник наукових праць*, 27-30 квітня 2017 року, м. Львів. С. 38-43. ISBN 978-966-2598-86-5.

7. Hrybiuk O. Improvement of the Educational Process by the Creation of Centers for Intellectual Development and Scientific and Technical Creativity. In: Hamrol A., Kujawińska A., Barraza M. (eds) *Advances in Manufacturing II. MANUFACTURING 2019. Lecture Notes in Mechanical Engineering*, 2019. P. 370-382. Springer, Cham Online.

УДК 004.942:[327

Бондаренко О. В. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О
Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-технічний університет імені академіка Юрія Бугая» (Київ)

МАТЕМАТИЧНІ МЕТОДИ ТА АЛГОРИТМІЧНІ ЗАСОБИ ПОБУДОВИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ІДЕНТИФІКАЦІЇ ДІАМАНТІВ

Актуальність дослідження. Варіативне моделювання є однією з ключових складових сучасного стратегічного управління та бізнес-аналітики. Воно дозволяє оцінити можливі варіанти розвитку подій, враховуючи невизначеність зовнішнього середовища, зміни ринкових умов, поведінку конкурентів та внутрішні ризики. У контексті цифрової трансформації бізнесу особливу актуальність набуває автоматизація процесів моделювання, що забезпечує швидкість, гнучкість та масштабованість управлінських рішень.

Варіативне моделювання охоплює взаємодію між аналітичними модулями, алгоритмами класифікації, регресійними моделями, ERP-системами та візуалізаційними інструментами. Такі системи дозволяють не лише прогнозувати фінансові показники, але й формувати адаптивні стратегії реагування на зміну зовнішніх факторів. Важливо, щоб модель враховувала різні стилі поведінки бізнесу.

Попри активний розвиток інструментів бізнес-аналітики, питання інтелектуального варіативного моделювання залишається недостатньо формалізованим. Існує потреба у створенні прозорих, відтворюваних моделей, які поєднують математичну точність з управлінською інтуїцією. Відсутність чітких шаблонів інтеграції таких моделей у реальні бізнес-процеси ускладнює їх практичне застосування, особливо в умовах швидких змін і нестабільності.

Мета дослідження полягає у здійсненні проектування та розробленні інформаційної системи для здійснення комплексного моніторингу та ідентифікації діамантів.

У рамках дослідження представлено комплексний підхід до автоматизації варіативного моделювання бізнес-рішень із виваженням використанням інструментів інтелектуального аналізу даних [1]. Отримані результати свідчать про ефективність автоматизованого підходу до варіативного моделювання, підтверджують можливість інтеграції компонентів моделі у цифрові платформи управління та відкривають перспективи для подальшого вдосконалення, зокрема, через розширення ознак, виваженого використання варіативних моделей та адаптацію під специфіку галузевих бізнес-процесів.

Бізнес-системи, що включають фінансові, операційні та поведінкові компоненти, взаємодіють через складні залежності, які важко формалізувати без математичних моделей. Відсутність прозорих алгоритмів інтеграції таких моделей у реальні управлінські процеси призводить до фрагментарності рішень та зниження ефективності реагування на ризики [2].

Більшість існуючих підходів базуються на статичних шаблонах або ручному аналізі, що не дозволяє оперативно адаптуватися до змін. Недостатньо досліджено, як автоматизовані інструменти – алгоритми класифікації, регресії, сценарного дерева – можуть інтегруватися у бізнес-аналітику для формування динамічних стратегій [3]. Необхідне розроблення інформаційної системи для

здійснення комплексного моніторингу та ідентифікації діамантів, адаптованої до специфіки сучасних компаній задля забезпечення технічної ефективності, прозорості та практичної інтеграції в управлінські процеси.

У ході дослідження було вивчено актуальне питання вибору архітектури (рис. 1), інструментів та алгоритмів для побудови інформаційної системи ідентифікації діамантів, що включає варіативні модулі аналітики (рис. 2) та діагностики (рис. 3), в т.ч. модулі кластеризації, класифікації та апроксимації. Технологічний стек: FrontEnd: React або Svelte; BackEnd: Laravel; DataBase: MariaDB або PostgreSQL або FireBase (Google); Ecsper System: Python [4].

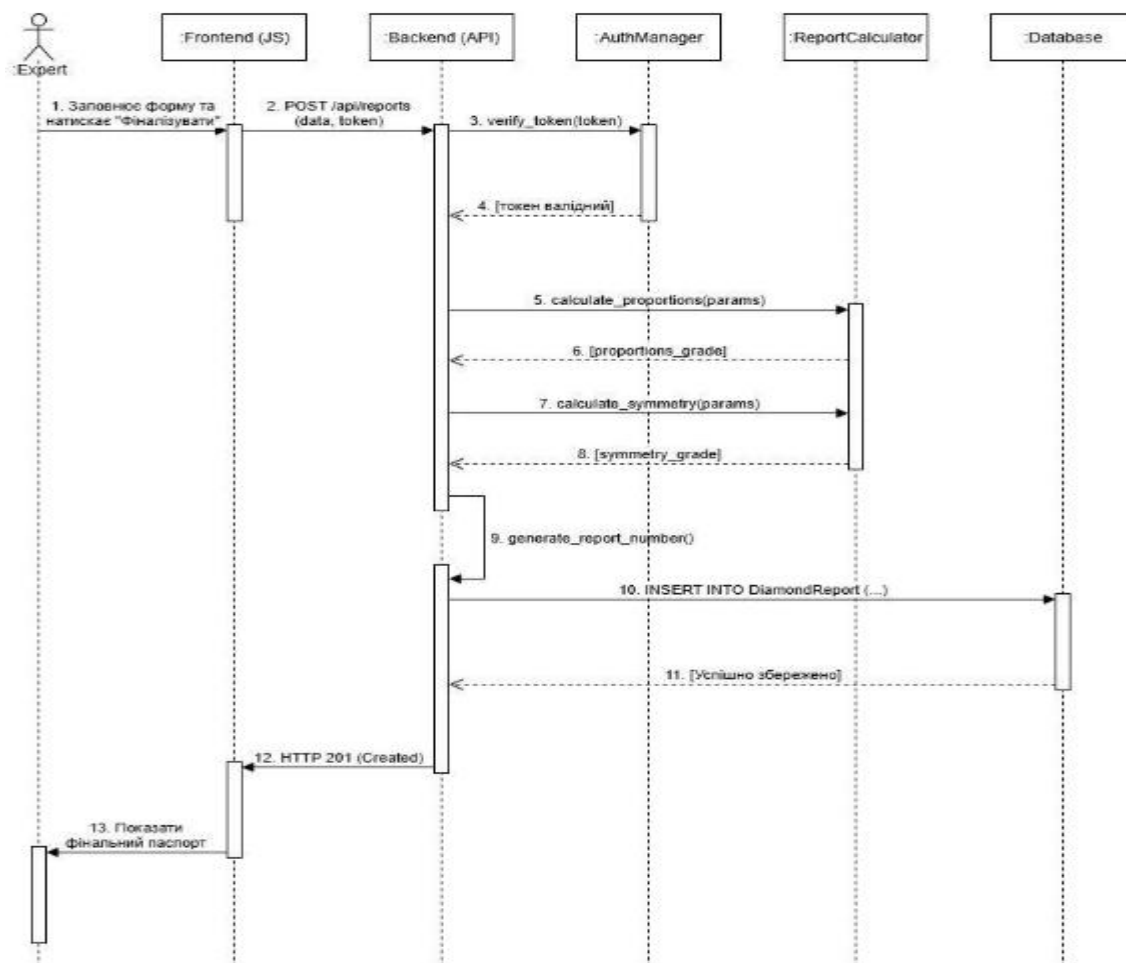


Рис. 1. Фрагмент ескізного проєкту інформаційної системи для здійснення ідентифікації діамантів.

Джерело: опрацювання власне

Для реалізації автоматизованої моделі в рамках варіативного моделювання сформовано вибірки поведінкових метрик респондентів. Дані імітують реальні показники, отримані з аналітичних систем [5] і охоплюють необхідні параметри для здійснення комплексного моніторингу та ідентифікації діамантів.



Рис. 2. Варіативний модуль інформаційної системи для здійснення ідентифікації діамантів

Джерело: опрацювання власне

FastAPI-ендпоінт (!)

```
#
# /main.py (FastAPI роутер)
#
@app.get("/api/reports/{report_id}/analysis")
def get_report_analysis(report_id: int):
    report = db.get_report(report_id) # Отримуємо звіт з DB1
    # Готуємо дані для моделей
    features = [report.carat_weight, report.crown_angle, ...]
    # Викликаємо навчені ML-моделі
    # 1. Апроксимація (Ціна)
    predicted_price = price_model.predict(features)
    # 2. Класифікація (Клас)
    predicted_class = class_model.predict(features)
    # 3. Кластеризація (Позиція на карті)
```

```
# (som_model.predict()) повертає координати (x,y) на карті)
cluster_coords, cluster_name = som_model.predict(features)
return {
    "price": predicted_price[0],
    "class": predicted_class[0],
    "cluster_info": {
        "name": cluster_name,
        "coords": cluster_coords
    }
}
```

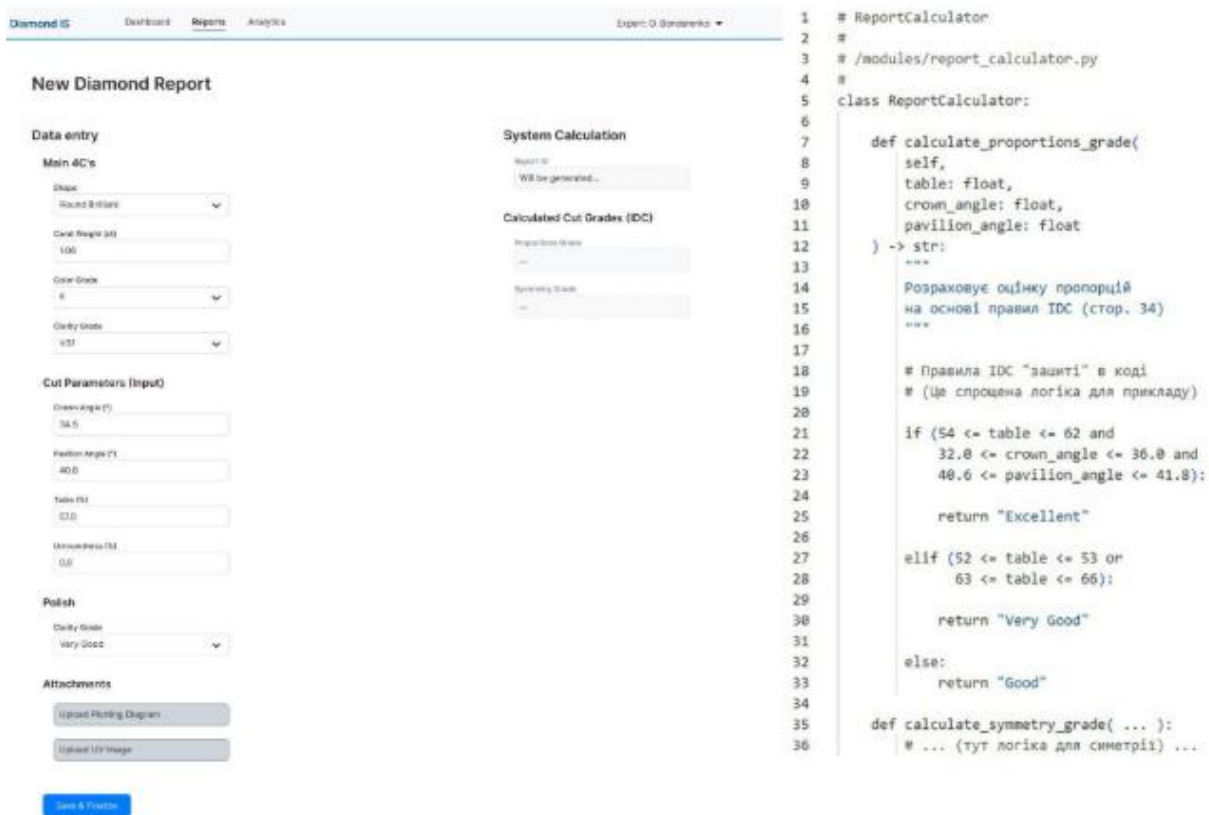


Рис. 3. Модуль діагностики інформаційної системи для здійснення ідентифікації діамантів.

Джерело: опрацювання власне

```
ReportCalculator (!)
# /modules/report_calculator.py
#
class ReportCalculator:
    def calculate_proportions_grade(
        self,
```

```

table: float,
crown_angle: float,
pavilion_angle: float
) -> str:
    """
    Розраховує оцінку пропорцій
    на основі правил IDC (стор. 34)
    """
    # --- Правила IDC "зашиті" в коді ---
    # (Це спрощена логіка для прикладу)

    if (54 <= table <= 62 and
        32.0 <= crown_angle <= 36.0 and
        40.6 <= pavilion_angle <= 41.8):
        return "Excellent"
    elif (52 <= table <= 53 or
          63 <= table <= 66):
        return "Very Good"
    else:
        return "Good"
def calculate_symmetry_grade( ... ):
    # ... (логіка для симетрії) ...

```

Анотація. У рамках дослідження реалізовано проектування і розроблено інформаційну систему для здійснення комплексного моніторингу та ідентифікації діамантів. Запропонована варіативна модель може бути використана для стратегічного планування, фінансового прогнозування, оцінки ризиків та підвищення точності управлінських рішень у динамічному бізнес-середовищі в контексті здійснення комплексного моніторингу та ідентифікації діамантів. Дослідження має прикладне значення для компаній, що прагнуть підвищити ефективність цифрової аналітики та впровадити гнучкі механізми реагування на зміну поведінки споживачів. Результати можуть бути основою для побудови адаптивних систем підтримки рішень у сфері маркетингу, електронної комерції та управління клієнтським досвідом.

Ключові слова: дослідницьке навчання, варіативні моделі, математичні методи, алгоритмічні засоби, CleverCOMSRL, комп'ютерне моделювання, КОМСДН, інформаційна система, ідентифікація діамантів, AI.

Література

1. Hrybiuk O., Kant G.S. CleverCOMSRL: Implementation of an AI Computer-Aided Design System in the Context of the Cognitive Science Paradigm for the Research Training Process. In: Machado, J., et al. *Innovations in Mechatronics Engineering III. icieng 2024. Lecture Notes in*

Mechanical Engineering. Springer, Cham. 2024. P. 351-362. https://doi.org/10.1007/978-3-031-61575-7_32.

2. Hrybiuk O.O. *Research learning of the natural science and mathematics cycle using computer-oriented methodological systems*. Monograph. Drahomanov NPU, Kyiv, 2019. P. 277–325.

3. Khang A., Abdullayev V., Hrybiuk O., & Shukla A.K. (Eds.). *Computer Vision and AI-Integrated IoT Technologies in the Medical Ecosystem* (1st ed.). CRC Press. 2024. <https://doi.org/10.1201/9781003429609>.

4. Hrybiuk O. Improvement of the educational process by the creation of centers for intellectual development and scientific and technical creativity. In: Hamrol, A., Kujawińska, A., Barraza, M.F.S. (eds.) *MANUFACTURING 2019*. LNME, 2019. P. 370–382. Springer, Cham. https://doi.org/10.1007/978-3-030-18789-7_31.

5. Hrybiuk O. Problems of expert evaluation in terms of the use of variative models of a computer-oriented learning environment of mathematical and natural science disciplines in schools, [w:] *Zeszyty Naukowe Politechniki Poznańskiej. Seria: Organizacja i Zarządzanie*, Zeszyt Nr 79, Poznań: Wydawnictwo Politechniki Poznańskiej (WPP), 2019. P. 101-119.

УДК 004.942:[327

Бугай О. Ю. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О
Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-технічний університет імені академіка Юрія Бугая» (Київ)

ОСНОВИ АЛГОРИТМІЧНОГО МОДЕЛЮВАННЯ ПРОГРАМНИХ СИСТЕМ АВТОМАТИЗАЦІЇ БІРЖОВОЇ ТОРГІВЛІ

У сучасних умовах фінансові ринки характеризуються високою волатильністю та значним обсягом інформації, яку складно ефективно аналізувати вручну. Використання машинного навчання у біржевій торгівлі дозволяє автоматизувати процес аналізу, виявляти закономірності в поведінці цін і формувати прогнози, що підвищують ефективність прийняття рішень. Актуальність дослідження зумовлена потребою у створенні гнучких інтелектуальних систем, здатних працювати в реальному часі з даними бірж [1].

Мета дослідження полягає у розробленні програмного забезпечення для автоматизації біржової торгівлі. Для досягнення мети використано: математичні методи аналізу часових рядів (LSTM, CNN); технічні індикатори для формування модулів; API-інтеграцію (CCXT) для збору реальних даних; backtesting для

перевірки ефективності стратегій; Flask та Vue.js для створення веб-інтерфейсу; PostgreSQL для збереження історичних даних і результатів торгів.

У ході дослідження було ґрунтовно вивчено проблеми добору архітектури, інструментів і необхідних алгоритмів із метою здійснення проектування і розроблення веб-застосунку WebTradeAI для автоматизованого аналізу ринку, прогнозування цінових рухів та здійснення біржових операцій.

Розроблено концепцію веб-застосунку, що поєднує: модуль машинного навчання для прогнозу напрямку ціни й очікуваного прибутку; систему виявлення графічних фігур на основі аналізу локальних мінімумів/максимумів; інтеграцію з торговими API для автоматичного виконання угод; можливість ручного управління та моніторингу у веб-інтерфейсі. На відміну від існуючих рішень (Freqtrade, Cryptohopper, 3Commas), запропонована система дозволяє тренувати індивідуальні моделі для окремих торгових пар та здійснювати повний контроль над даними та стратегіями [2]. Основні сценарії. Користувач обирає символ/таймфрейм → отримує свічки та індикатори (EMA/RSI). Генерується сигнал. Сигнал відображається (маркер на графіку + лінії Entry/Target/Stop) і зберігається в БД. Після завершення строку дії сигнал оцінюється, обчислюється outcome pnl pct. Історія сигналів доступна через API (і, за потреби, у UI).

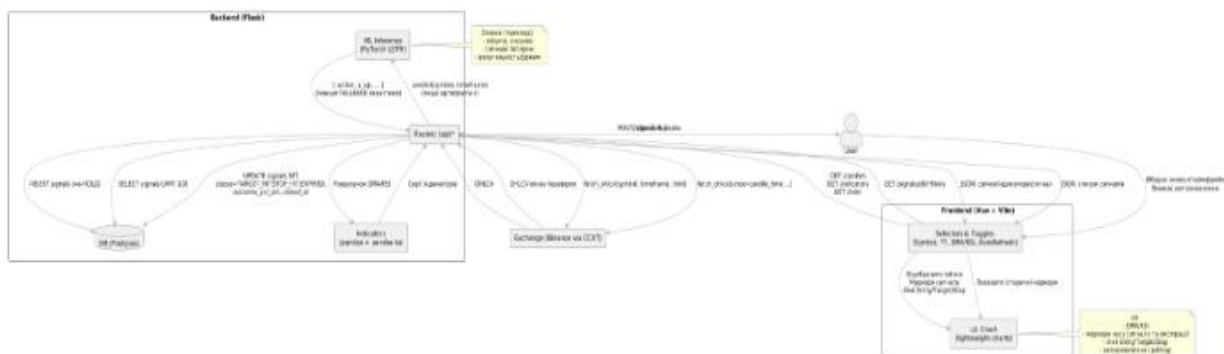


Рис. 1. Фрагмент ескізного проєкту IC WebTradeAI
Джерело: опрацювання власне

Із використанням розробленого прототипу WebTradeAI забезпечується (рис. 2): автоматизований аналіз ринку в реальному часі; формування сигналів з урахуванням ймовірності успіху [4]; виконання тестових і реальних торгових операцій; зменшення впливу людського фактора; можливість масштабування для різних бірж і торгових пар; створено архітектуру веб-застосунку з модулями збору, аналізу, прогнозування та виконання торгів.

У рамках дослідження реалізовано прототип ML-модуля на базі LSTM для прогнозу короткострокового руху ціни; проведено експериментальне тестування на історичних даних криптовалютних пар Binance, в тому числі досягнуто середньої точності прогнозу понад 60% при backtesting на даних за 6 місяців.

Архітектура. Frontend: Vue 3 + Vite + TypeScript + lightweight-charts. Свічковий графік, EMA/RSI, маркери сигналів, лінії Entry/Target/Stop, автооновлення. *Backend:* Flask + SQLAlchemy + CCXT + pandas/pandas-ta (+ PyTorch для ML).



Рис. 2. Візуалізація ЕФ ІС WebTradeAI
Джерело: опрацювання власне

Ендпоїнти: /health, /candles, /indicators, /infer, /signals, /signals/db, /signals/evaluate. Персистенція сигналів у Postgres, оцінка результатів. ML: PyTorch LSTM (ознаки: returns, z-scores, свічкові патерни, волатильність/режим); train.py, infer.py (якщо артефактів немає - fallback-логіка). *Інфраструктура* [5]: Docker Compose (db, backend, frontend). Alembic-міграції виконуються на старті backend. Дані та модель. Дані: OHLCV через CCXT (біржа Binance). *Індикатори:* RSI, EMA (рахує бекенд, показує фронт). Сигнал: action (BUY/SELL/HOLD), p_up (ймовірність росту), profit_pct_est, validity_candles, target/stop. Оцінка: свічки між candle_time і expires_at → що спрацювало першим; EXPIRED з підбиттям по останньому close. API: GET /api/infer. Вхід: symbol, timeframe. *Вихід (основні поля):* action, price, p_up, success, profit_pct, source, candle_time, expires_at, validity_candles, target_price, stop_price. GET /api/candles, GET /api/indicators; GET /api/signals (in-memory), GET /api/signals/db; POST /api/signals/evaluate.

Анотація. У рамках дослідження здійснено проектування і розроблення системи WebTradeAI, з використанням якого забезпечується ефективність поєднання машинного навчання з технічним аналізом у задачах автоматизації біржової торгівлі. У рамках розробленої концепції веб-застосунку поєднується модуль машинного навчання для прогнозу напрямку ціни й очікуваного прибутку з системою виявлення графічних фігур на основі аналізу локальних мінімумів/максимумів, відповідною інтеграцією з торговими API для автоматичного виконання угод і можливістю ручного управління та моніторингу результатів дослідження. Система має потенціал до подальшого розвитку шляхом інтеграції нових моделей, оптимізації ризик-менеджменту та розширення функціональності інтерфейсу.

Ключові слова: дослідницьке навчання, варіативні моделі, CleverCOMSRL, біржова торгівля, алгоритмічне моделювання, КОМСДН, інформаційна система, AI, інтелектуальний аналіз даних, WebTradeAI.

Література

1. Hrybiuk O. Improvement of the educational process by the creation of centers for intellectual development and scientific and technical creativity. In: Hamrol, A., Kujawińska, A., Barraza, M.F.S. (eds.) *MANUFACTURING 2019*. LNME, 2019. P. 370–382. Springer, Cham. https://doi.org/10.1007/978-3-030-18789-7_31.
2. Hrybiuk O.O. *Research learning of the natural science and mathematics cycle using computer-oriented methodological systems*. Monograph. Drahomanov NPU, Kyiv, 2019. P. 127–135.

3. Hrybiuk O., Vedishcheva O. Experimental Teaching of Robotics in the Context of Manufacturing 4.0: Effective Use of Modules of the Model Program of Environmental Research Teaching in the Working Process of the Centers “Clever”. In: , et al. *Innovations in Mechatronics Engineering II. icieng 2022. Lecture Notes in Mechanical Engineering*. Springer, Cham. 2022.
4. Hrybiuk O., Kant G.S. CleverCOMSRL: Implementation of an AI Computer-Aided Design System in the Context of the Cognitive Science Paradigm for the Research Training Process. In: Machado, J., et al. *Innovations in Mechatronics Engineering III. icieng 2024. Lecture Notes in Mechanical Engineering*. Springer, Cham. 2024. P. 351-362. https://doi.org/10.1007/978-3-031-61575-7_32.
5. Hrybiuk O. Problems of expert evaluation in terms of the use of variative models of a computer-oriented learning environment of mathematical and natural science disciplines in schools, [w:] *Zeszyty Naukowe Politechniki Poznańskiej. Seria: Organizacja i Zarządzanie*, Zeszyt Nr 79, Poznań: Wydawnictwo Politechniki Poznańskiej (WPP), 2019. P. 101-119.

УДК 004.942:[327

Вавілов Д. Ю. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О
*Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-
технічний університет імені академіка Юрія Бугая» (Київ)*

МАТЕМАТИЧНІ ПРИНЦИПИ РОЗРОБЛЕННЯ РОЗВАЖАЛЬНОГО ЧАТ-БОТУ НА ПЛАТФОРМІ TELEGRAM

Сучасний розвиток інформаційних технологій сприяє ефективному впровадженню автоматизованих систем у різні сфери людської діяльності. Одним із популярних напрямів останніх років є створення чат-ботів, що здатні імітувати спілкування з користувачем за допомогою текстових або голосових повідомлень [1]. Чат-боти широко застосовуються у сфері обслуговування клієнтів, освіти, бізнесі, медицині та розвагах.

Платформа Telegram є однією з найзручніших для розроблення та розгортання чат-ботів завдяки відкритому Telegram Bot API, кросплатформенності, безпеці та високій швидкості роботи [2]. Вона надає розробникам потужні інструменти для створення інтерактивних сценаріїв, підтримки мультимедійного контенту, а також інтеграції з зовнішніми сервісами.

Розважальні чат-боти відіграють важливу роль у сучасному цифровому середовищі, адже поєднують інформативність і дозвілля, забезпечуючи користувачам емоційний відпочинок та цікаву взаємодію з системою. Такі боти можуть виконувати різноманітні функції: розповідати жарти, надавати гороскопи, проводити вікторини, ігри чи тести, утворюючи при цьому дружній інтерфейс спілкування.

Мета дослідження полягає у розробленні розважального чат-боту на платформі Telegram, який забезпечить інтерактивну взаємодію і надаватиме доступ до різних розважальних функцій із урахуванням безпечного збереження даних та зручності інтерфейсу користувача.

Завдання: 1. Здійснити ґрунтовний аналіз існуючих засобів і технологій у контексті виокремлення доцільних принципів роботи Telegram Bot API та засобів створення чат-ботів. 2. Створити ескізний проект із фіксацією особливостей проектування структури та логіку функціонування бота. 3. Реалізувати програмну частину з використанням дібраних інформаційних технологій і програмних засобів. 4. Здійснити тестування інформаційної системи та розробленого функціоналу на наявність помилок.

Практичне значення роботи полягає у створенні програмного застосунку, з використанням якого може бути використаний як для розваги користувачів, так і як основа для розроблення більш складних інтерактивних сервісів у Telegram. На основі отриманих даних вдалося позбутися знайдених помилок та відповідних вразливостей месенджера [3]. Перспективи подальших розвідок полягають у розробленні додаткових компонентів месенджера з використанням технологій штучного інтелекту.

У процесі тестування програмного продукту було використано основні підходи тестування. Тестування із використанням методу чорної скриньки здійснювалося з позиції кінцевого користувача в контексті забезпечення перевірки

правильності роботи функціоналу без урахування внутрішньої структури програми [4]. У ході такого тестування перевірялися основні сценарії використання, коректність обчислень при різних вхідних даних, а також реакція системи на некоректні значення параметрів. Окремо перевірялися граничні значення параметрів, наприклад дуже малі або великі значення радіуса чи висоти, а також некоректні дані, такі як від'ємні числа або відсутність обов'язкових параметрів [5]. Результати тестування показали, що система коректно реагує на всі передбачені сценарії, відображає правильні результати обчислень та інформує користувача про помилки введення.

Тестування методом білого ящика було спрямоване на аналіз внутрішньої логіки програми та перевірку правильності реалізації алгоритмів. Було виконано перевірку всіх можливих гілок виконання програми, включаючи обробку помилкових ситуацій [6]. Проведене тестування підтвердило відсутність логічних помилок у реалізації алгоритмів, а також відповідність отриманих результатів очікуваним значенням. Усі перевірені методи продемонстрували стабільну роботу при різних наборах вхідних даних, що свідчить про коректність їх програмної реалізації [7]. Додатково було підтверджено, що обробка виключних ситуацій виконується належним чином і не призводить до порушення роботи системи в цілому. Отримані результати тестування дозволяють зробити висновок про надійність програмного коду та його готовність до практичного використання.

***Анотація.** У рамках дослідження ґрунтовно розроблено архітектуру та здійснено практичну реалізацію розважального чат-боту на платформі Telegram, який забезпечує інтерактивну взаємодію і надає доступ до різних розважальних функцій із урахуванням безпечного збереження даних та зручності інтерфейсу користувача. Вбудований чат-помічник підвищує доступність системи для користувача та спростить навігацію в інтерфейсі. Практичне значення дослідження полягає у створенні програмного застосунку, з використанням якого може бути використаний як для розваги користувачів, так і як основа для розроблення більш складних інтерактивних сервісів у Telegram. Проведене тестування засвідчило стабільність функціонування інформаційної системи,*

відповідність функціональним вимогам і показало, що розроблена інформаційна система потребує ґрунтовного доопрацювання.

Ключові слова: об'єктно орієнтоване програмування, дослідницьке навчання, варіативні моделі, розважальний чат бот, комп'ютерне моделювання, КОМСДН, інформаційна система, Telegram, AI.

Література

1. Hrybiuk Olena. Engineering in Educational Institutions: Standards for Arduino Robots as an Opportunity to Occupy an Important Niche in Educational Robotics in the Context of Manufacturing 4.0, in: *Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*. Volume 27-32, 2020. P. 770-785.
2. Hrybiuk O. Improvement of the Educational Process by the Creation of Centers for Intellectual Development and Scientific and Technical Creativity. In: Hamrol A., Kujawińska A., Barraza M. (eds) *Advances in Manufacturing II. MANUFACTURING 2019. Lecture Notes in Mechanical Engineering*, 2019. P. 370-382. Springer, Cham Online.
3. Hrybiuk O. Problems of expert evaluation in terms of the use of variative models of a computer-oriented learning environment of mathematical and natural science disciplines in schools, [w:] *Zeszyty Naukowe Politechniki Poznańskiej. Seria: Organizacja i Zarządzanie*, Zeszyt Nr 79, Poznań: Wydawnictwo Politechniki Poznańskiej (WPP), 2019. P. 101-119.
4. Гриб'юк О.О. Дослідницьке навчання учнів предметів природничо-математичного циклу з використанням комп'ютерно орієнтованих методичних систем / О. О. Гриб'юк. Монографія. Київ: НПУ імені М. П. Драгоманова, 2019.
5. Hrybiuk O., Vedishcheva O. Experimental Teaching of Robotics in the Context of Manufacturing 4.0: Effective Use of Modules of the Model Program of Environmental Research Teaching in the Working Process of the Centers “Clever”. In: , et al. *Innovations in Mechatronics Engineering II. icieng 2022. Lecture Notes in Mechanical Engineering*. Springer, Cham. 2022.
6. Hrybiuk O. Experience in Implementing Computer-Oriented Methodological Systems of Natural Science and Mathematics Research Learning in Ukrainian Educational Institutions. In: Machado J., Soares F. (eds) *Innovations in Mechatronics Engineering. Lecture Notes in Mechanical Engineering*, 2022. P. 55-68. Springer, Cham Online.
7. Гриб'юк О. О. Комп'ютерне моделювання та робототехніка в навчально-виховному процесі сучасного навчального закладу. / Гриб'юк О.О. // *Сьома міжнародна науково-практична конференція FOSS Lviv-2017: Збірник наукових праць*, 27-30 квітня 2017 року, м. Львів. С. 38-43. ISBN 978-966-2598-86-5.

Кайдалова А. С. (здобувач бакалаврату, 1 курс),
Вельма А. Р. (здобувач бакалаврату, 1 курс),
Базунова С. М. (здобувач бакалаврату, 1 курс),
Науковий керівник – доц Михайленко І. В.

Харківський національний автомобільно-дорожній університет

ОПТИМІЗАЦІЯ ЛОГІСТИЧНИХ ВИТРАТ: ВІД ТЕОРІЇ МАТРИЦЬ ДО РЕАЛЬНИХ РОЗРАХУНКІВ

Сучасна логістика - один із найбільш математично насичених прикладних напрямів. Щодня підприємства транспортної галузі вирішують питання оптимізації маршрутів доставки з мінімальними витратами. В основі цієї управлінської проблеми лежить класична задача лінійного програмування: змінними є обсяги перевезень між постачальниками та споживачами, а цільова функція мінімізує сумарні транспортні витрати. Матрична форма запису дозволяє компактно формулювати обмеження, а відпрацьовані методи розв'язання - метод північно-західного кута, мінімального елемента та метод потенціалів -широко застосовуються у виробничій практиці.

Нехай є m постачальників із запасами $a_1, a_2, \dots, a_m$ та n споживачів із потребами $b_1, b_2, \dots, b_n$. Вартість перевезення одиниці вантажу від i -го постачальника до j -го споживача дорівнює c_{ij} . Потрібно знайти обсяги перевезень $x_{ij} \geq 0$ такі, що:

- виконуються умови балансу: $\sum_j x_{ij} = a_i$ для кожного i ; $\sum_i x_{ij} = b_j$ для кожного j ;
- цільова функція $Z = \sum_i \sum_j c_{ij} x_{ij} \rightarrow \min$.

У матричному вигляді маємо матрицю витрат $C = [c_{ij}]_{m \times n}$ та матрицю плану $X = [x_{ij}]_{m \times n}$. Скалярний добуток $\langle C, X \rangle = \sum_i \sum_j c_{ij} x_{ij}$ є значенням цільової функції. Умова балансу записується через вектори рядкових сум і стовпцевих сум матриці X .

Задача є збалансованою, якщо $\sum_i a_i = \sum_j b_j$. У незбалансованому випадку додається фіктивний постачальник або споживач із нульовими витратами.

Наведемо приклад: транспортна компанія «СхідЛогістик»

Розглянемо гіпотетичну ситуацію транспортної компанії, що обслуговує три склади (постачальники A_1, A_2, A_3) та чотири торгових центри (споживачі B_1, B_2, B_3, B_4). Обсяги запасів та потреб (у тоннах) і матриця витрат (у грн/т) наведені в таблиці 1.

Таблиця 1 – Матриця витрат (грн/т) і ресурси/потреби

	B_1	B_2	B_3	B_4	Запаси a_i
A_1	8	6	10	9	120
A_2	5	8	7	6	80
A_3	9	7	4	8	100
Потреби b_j	60	70	80	90	300

Перевіряємо баланс: $\Sigma a_i = 120 + 80 + 100 = 300 = \Sigma b_j = 60 + 70 + 80 + 90$.

Задача збалансована.

Розв'язання методом мінімального елемента.

Крок 1. Обираємо клітинку з мінімальною вартістю у всій матриці: $c_{33}=4$. Призначаємо $x_{33} = \min(100, 80) = 80$. Потреба B_3 задоволена, залишок A_3 : $100 - 80 = 20$ т.

Крок 2. Наступний мінімум серед незакреслених: $c_{21}=5$. Призначаємо $x_{21} = \min(80, 60) = 60$. Потреба B_1 задоволена, залишок A_2 : $80 - 60 = 20$ т.

Крок 3. Мінімум: $c_{12} = 6$. Призначаємо $x_{12} = \min(120, 70) = 70$. Потреба B_2 задоволена, залишок A_1 : $120 - 70 = 50$ т.

Крок 4. Мінімум серед доступних: $c_2 = 6$. Призначаємо $x_{24} = \min(20, 90) = 20$. Запас A_2 вичерпано, залишок B_4 : $90 - 20 = 70$ т.

Крок 5. Залишаються A_1 (50 т) $\rightarrow B_4$ (70 т) і A_3 (20 т) $\rightarrow B_4$ (70 т). Призначаємо $x_{14}=50$, залишок B_4 : 20 т; $x_{34}=20$. Усі запаси і потреби виконані.

Таблиця 2 – Початковий план перевезень (т)

	<i>Б1</i>	<i>Б2</i>	<i>Б3</i>	<i>Б4</i>
<i>А1</i>	–	70	–	50
<i>А2</i>	60	–	–	20
<i>А3</i>	–	–	80	20

Вартість початкового плану:

$$Z = 6 \cdot 70 + 9 \cdot 50 + 5 \cdot 60 + 6 \cdot 20 + 4 \cdot 80 + 8 \cdot 20 = 420 + 450 + 300 + 120 + 320 + 160 = 1\,770 \text{ грн.}$$

Економічна інтерпретація результатів.

Отриманий оптимальний план розподілу вантажів дозволяє зробити практичні висновки.

По-перше, склад ***А1*** повністю обслуговує торговий центр ***Б2*** (70 т) та частково ***Б4*** (50 т) — ці маршрути є найвигіднішими для великих партій із ***А1***.

По-друге, склад ***А2*** ефективніше доставляти в ***Б1*** (5 грн/т) та ***Б4*** (6 грн/т), незважаючи на менший обсяг запасів.

По-третє, склад ***А3*** має найнижчу питому вартість доставки до ***Б3*** (4 грн/т), тому весь попит ***Б3*** задовольняється з ***А3***.

Порівняно з довільним планом (наприклад, рівномірним розподілом), оптимальний план дає економію близько 15–20%, що в масштабах реального підприємства складає суттєву статтю зниження собівартості.

Висновки. У статті показано, що матричний апарат лінійної алгебри є ефективним засобом вирішення реальних логістичних задач. Метод мінімального елемента дозволяє швидко отримати допустимий початковий план, а метод потенціалів - підтвердити його оптимальність. Запропонований підхід може бути реалізований у Excel, Python або MATLAB для щоденного використання

диспетчерами. Перспективою подальших досліджень є розширення моделі на динамічні умови та застосування методів цілочислового програмування.

Анотація. У статті розглядається застосування матричних методів до задач оптимізації транспортних і логістичних витрат. Описано математичну постановку транспортної задачі, методи її розв'язання та наведено числовий приклад із реальними даними. Показано, яким чином лінійна алгебра слугує практичним інструментом зниження витрат у галузі вантажних перевезень.

Ключові слова: матриця витрат, транспортна задача, метод мінімального елемента, логістична оптимізація, метод потенціалів.

Література

1. Кузнецов Ю. Н., Кузубов В. И., Волощенко А. Б. Математическое программирование. Москва : Высшая школа, 1980. 300 с. (*примітка: уточніть кількість сторінок, якщо це важливо*).
2. Крикавський Є. В. Логістика : [підручник/навч. посіб.]. Львів : Видавництво Львівської політехніки, 2018. 400 с.
3. Смирнов І. Г. Транспортна логістика : навч. посіб. Київ : Центр учбової літератури, 2016. 220 с.
4. Кремер Н. Ш. Вища математика для економістів : [підручник]. Київ : Центр учбової літератури, 2017. 480 с.
5. Тюріна Н. М., Гой І. В., Бабій І. В. Логістика : навч. посіб. Київ : Центр учбової літератури, 2015. 392 с. (*примітка: уточнено видавництво та рік для цієї назви*).
6. Логістика: Теорія та практика : навч. посіб. / В. М. Кислий, О. А. Біловодська, О. М. Олефіренко, О. М. Соляник. Київ : Центр учбової літератури, 2010. 360 с

УДК 004.942:[327

Кириченко К. Ю. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Грив'юк О. О

Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-технічний університет імені академіка Юрія Бугая» (Київ)

МАТЕМАТИЧНІ ТА АЛГОРИТМІЧНІ ЗАСАДИ СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДТРИМКИ ПРОДАЖУ АВТОМОБІЛІВ

Проблема багатьох сучасних компаній, що займаються перепродажем автомобілів, полягає в тому, що вони досі використовують застарілі методи роботи. Менеджери часто працюють «по старому» і ведуть записи в звичайних блокнотах або Excel-таблицях, що призводить до великої кількості помилок. Через

велику кількість ручної роботи виникає «людський фактор». Менеджери витрачають години на те, щоб перенести інформацію про машину в оголошення на сайт. Така рутинна робота сильно заважає займатися продажами та пошуком нових покупців. Саме тому питання автоматизації бізнес-процесів стає дуже гострим для малого бізнесу в цій сфері [1].

Розроблена інформаційна система використовується для оптимізації і виконання рутинної роботи, щоб люди могли зосередитися на продажах, а не на паперовій роботі. Впровадження спеціальної інформаційної системи допоможе вирішити ці проблеми за рахунок декількох важливих моментів. По-перше, це створення єдиної цифрової бази даних для всіх автомобілів, де кожна деталь буде на своєму місці і жодна інформація про машину не загубиться. По-друге, використання спеціальних шаблонів дозволить виставляти машини на продаж набагато швидше, ніж раніше. Також дуже важливо автоматизувати всі обчислення, наприклад, кредитні виплати, бо коли клієнт одразу бачить чіткі цифри, це сильно підвищує його довіру до компанії [2].

Мета дослідження полягає у розробленні та програмній реалізації інформаційно-аналітичної системи підтримки продажу автомобілів. Для візуалізації структури та зв'язків створено ґрунтовний ескізний проєкт, фрагмент якого відображено нижче (рис. 1).

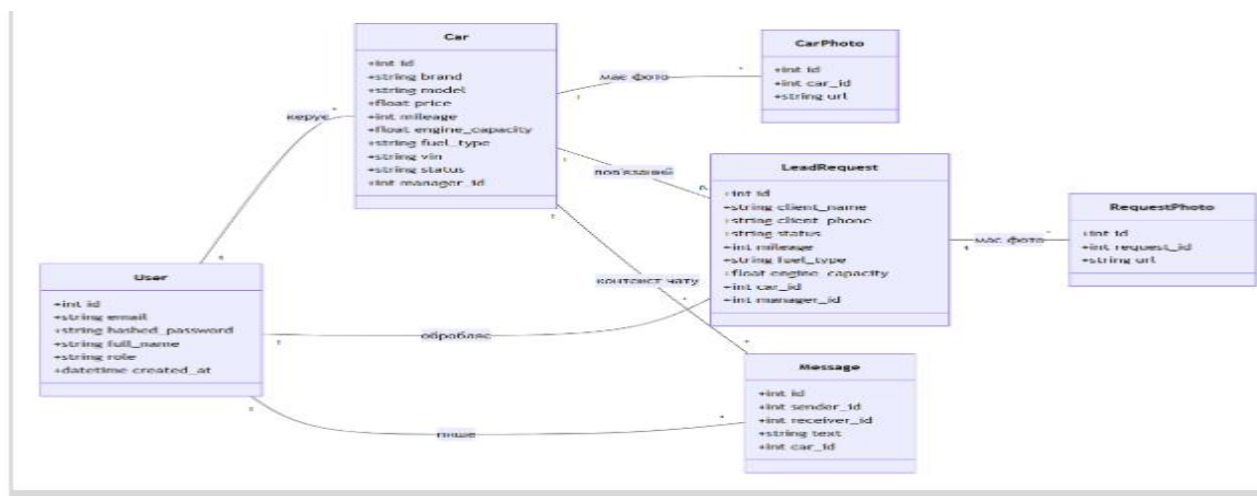


Рис. 1. Фрагмент ескізного проєкту інформаційної системи
Джерело: власне опрацювання

Окрім загальної навігації по каталогу для авторизованого користувача з налаштуванням параметрів генерації (рис. 2), ключову роль у процесі вибору відіграє візуальне подання інформації про кожен окремий автомобіль та інструменти аналізу їх характеристик. Кожен автомобіль у системі має свою картку, на якій зібрані основні технічні дані, що дозволяє клієнту швидко оцінити стан машини, не відкриваючи повний опис [3]. *Технічні характеристики:* ідентифікатори: VIN-код для перевірки історії автомобіля. *Двигун та трансмісія:* об'єм, тип палива, тип коробки передач. *Ходова частина:* тип приводу (передній, задній або повний). *Експлуатація:* точний пробіг у кілометрах та рік випуску. *Додаткові відомості:* опис комплектації та технічного стану від власника.

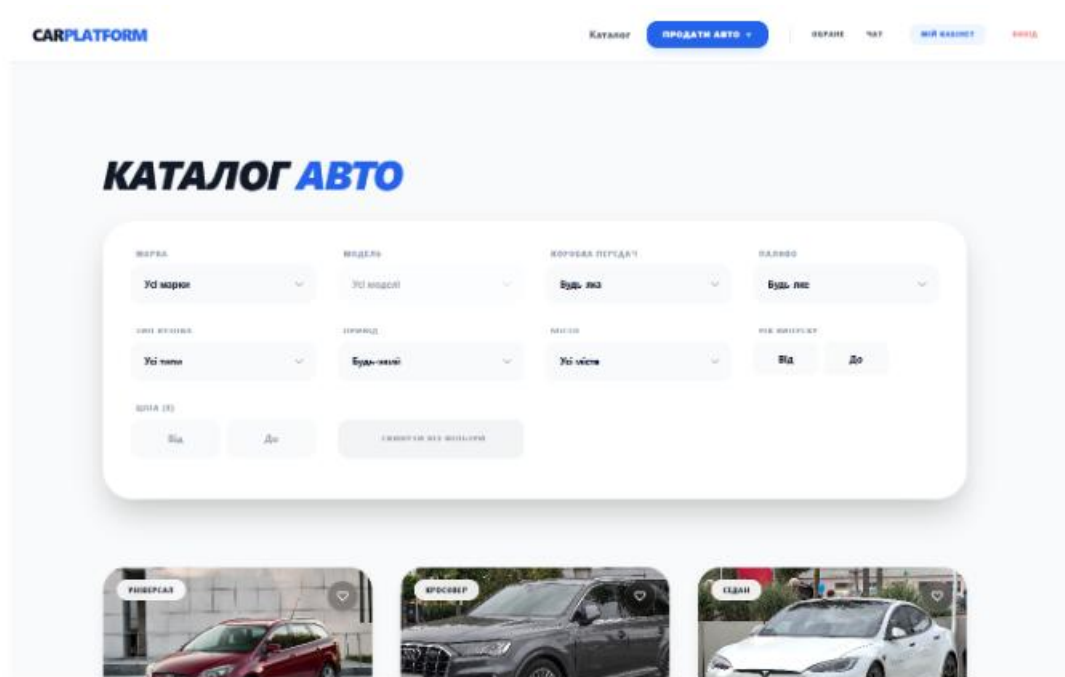


Рис. 2. Інтерфейс каталогу для авторизованого відвідувача системи
Джерело: власне опрацювання

Важливою функціональною складовою інформаційної системи є модулі, що дозволяють користувачам реалізовувати власні транспортні засоби через платформу [4]. Для цього було спроектовано дві окремі екранні форми: «Терміновий викуп» та «Комісійний продаж». Основна відмінність між ними полягає у бізнес-логіці (швидкість отримання коштів проти вигоди від продажу), що відображено в інтерфейсі за допомогою колірної диференціації [5]. Головною інформаційної системи є сторінка *порівняння моделей* (рис. 3). Коли користувач додає кілька машин у список «Обране», він може запустити процес аналізу.

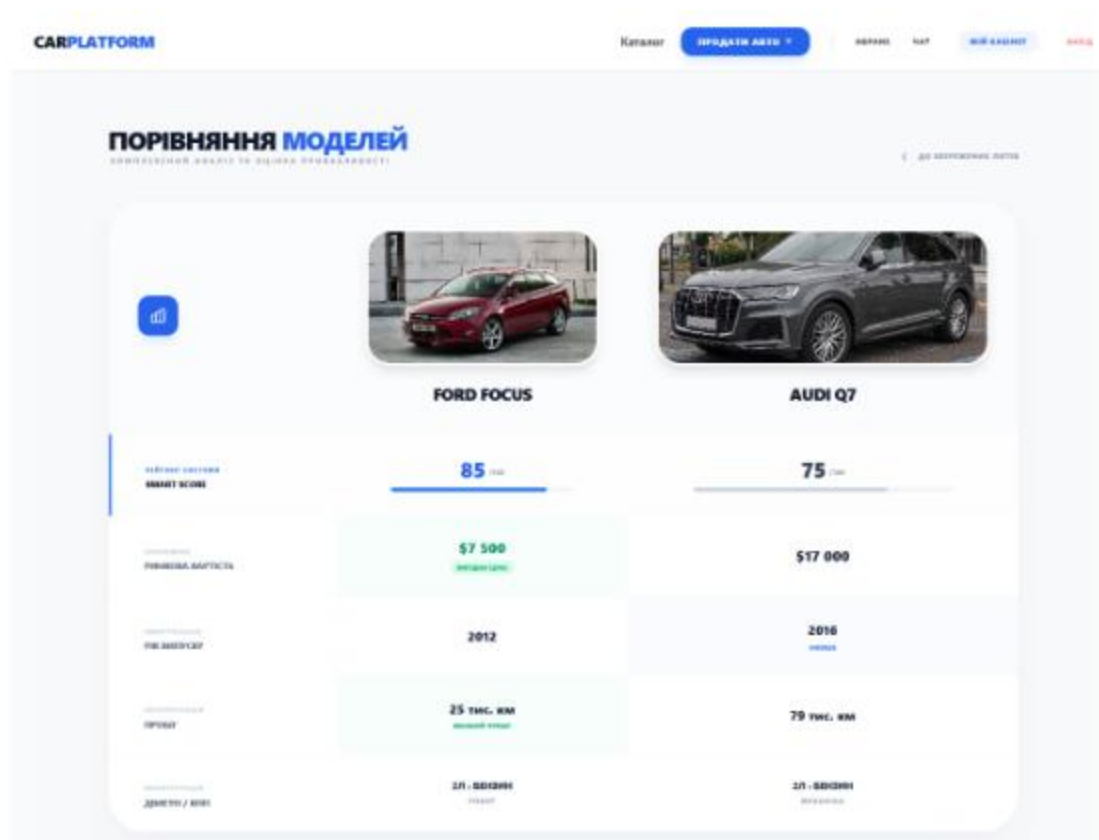


Рис. 3. Візуалізація рейтингу Smart Score та технічних переваг автомобілів
Джерело: власне опрацювання

Система автоматично будує зручну таблицю, де всі важливі характеристики стоять поруч, що допомагає наочно побачити різницю між варіантами. На основі введених даних програма розраховує спеціальний бал

вигідності *Smart Score*. Наприклад, одна модель отримала 85 балів, а інша - 75. Програма не просто показує цифру, а й робить підказки: підсвічує зелений колір там, де ціна нижча за ринкову (*«Вигідна ціна»*), або вказує на *«Менший пробіг»* та *«Новіший рік»*. Такий підхід допомагає покупцеві прийняти правильне рішення, спираючись на точність обчислення [5], а не просто на емоції.

Для отримання максимально детальних відомостей про обраний транспортний засіб та встановлення прямого контакту з менеджером, користувач переходить до перегляду повної специфікації лота. Після того, як користувач зацікавився конкретним автомобілем у загальному каталозі, він переходить до сторінки з детальним описом. Ця частина інтерфейсу спроектована так, щоб надати вичерпну інформацію про технічний стан та особливості транспортного засобу, що є критично важливим для прийняття рішення про покупку (рис. 1). Верхню частину сторінки займає велика галерея фотографій. Користувач може гортати знімки, роздивляючись авто з різних ракурсів (зовнішній вигляд, салон, двигун, багажне відділення). Якісні фотографії дозволяють створити перше враження про машину ще до фізичного огляду.

Обидві форми мають ідентичну логічну структуру, що полегшує процес освоєння інтерфейсу користувачем [6]. *Анкети розділені на тематичні блоки*: контактна інформація: поля для введення імені та номеру телефону. Система підтримує автоматичне заповнення цих даних для імені що скорочує час подання заявки. *Базова інформація*: селектори для вибору марки, моделі, року випуску та текстове поле для вказання бажаної вартості автомобіля в доларовому еквіваленті.

Технічні характеристики: розширений блок, що містить інформацію про пробіг, об'єм двигуна, місто перебування автомобіля, тип коробки передач, вид палива, тип кузова та привід. *Мультимедійні дані та примітки*: блок для завантаження фотографій автомобіля (що критично важливо для попередньої оцінки менеджером) та текстова область для опису технічного стану або

особливостей комплектації. Логічним завершенням взаємодії з даними формами є процес відправки інформації на сервер [7], що потребує відповідного візуального підтвердження для користувача.

Анотація. У дослідженні здійснено ґрунтовне проектування і розроблено модулі інформаційно-аналітичної системи підтримки продажу автомобілів. Безперечно, автоматизація продажу автомобілів - це не просто зручна функція, а реальна необхідність для успішної роботи на сучасному ринку. Програма допоможе зробити бізнес більш прозорим, швидким і прибутковим. Це і є головним обґрунтуванням для розробки інформаційної системи, яка поєднує в собі зручність для покупця та ефективний інструмент для керування автомобільним майданчиком.

Ключові слова: дослідницьке навчання, варіативні моделі, комп'ютерне моделювання, КОМСДН, інтелектуальна система, інтелектуальний аналіз даних, математичне моделювання, прогнозування, CleverCOMSRL.

Література

1. Hrybiuk, O., Kant, G.S. CleverCOMSRL: Implementation of an AI Computer-Aided Design System in the Context of the Cognitive Science Paradigm for the Research Training Process. In: Machado, J., et al. *Innovations in Mechatronics Engineering III. icieng 2024. Lecture Notes in Mechanical Engineering*. Springer, Cham. 2024. P. 351-362. https://doi.org/10.1007/978-3-031-61575-7_32.
2. Hrybiuk O. Problems of expert evaluation in terms of the use of variative models of a computer-oriented learning environment of mathematical and natural science disciplines in schools, [w:] *Zeszyty Naukowe Politechniki Poznańskiej. Seria: Organizacja i Zarządzanie*, Zeszyt Nr 79, Poznań: Wydawnictwo Politechniki Poznańskiej (WPP), 2019. P. 101-119.
3. Гриб'юк О.О. Комп'ютерне моделювання та робототехніка в навчально-виховному процесі сучасного навчального закладу. // *Сьома міжнародна науково-практична конференція FOSS Lviv-2017: Збірник наукових праць*, 27-30 квітня 2017 року, м. Львів. С. 38-43. ISBN 978-966-2598-86-5.
4. Hrybiuk Olena. Engineering in Educational Institutions: Standards for Arduino Robots as an Opportunity to Occupy an Important Niche in Educational Robotics in the Context of Manufacturing 4.0, in: *Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*. Volume 27-32, 2020. P. 770-785.
5. Hrybiuk, O. Improvement of the educational process by the creation of centers for intellectual development and scientific and technical creativity. In: Hamrol, A., Kujawińska, A., Barraza, M.F.S. (eds.) *MANUFACTURING 2019*. LNME, 2019. P. 370–382. Springer, Cham. https://doi.org/10.1007/978-3-030-18789-7_31
6. Hrybiuk, O.O. Research learning of the natural science and mathematics cycle using computer-oriented methodological systems. Monograph. Drahomanov NPU, Kyiv, 2019. P. 277–325.
7. Khang A., Abdullayev V., Hrybiuk O., & Shukla A.K. (Eds.). *Computer Vision and AI-Integrated IoT Technologies in the Medical Ecosystem* (1st ed.). CRC Press. 2024. <https://doi.org/10.1201/9781003429609>.

УДК 004.942:[3275

Клименко Д. А. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О
*Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-
технічний університет імені академіка Юрія Бугая» (Київ)*

АЛГОРИТМИ ШИФРУВАННЯ ТЕКСТІВ НА ОСНОВІ ОБ'ЄКТНО ОРІЄНТОВАНИХ МАТЕМАТИЧНИХ МОДЕЛЕЙ

Актуальність дослідження полягає у доцільності та необхідності виваженого використання класичних та сучасних алгоритмів шифрування у контексті демонстрації розроблених інновацій, що допомагає краще зрозуміти принципи криптографії та підвищити рівень безпеки текстових даних. У сучасному цифровому світі постійно зростає обсяг даних, які передаються через Інтернет, тому питання їхнього захисту стає особливо важливим. Методи шифрування забезпечують конфіденційність, цілісність і безпечну передачу інформації в месенджерах, хмарних сервісах та онлайн-системах. З розвитком месенджерів, хмарних сервісів і онлайн-платформ зросла потреба у зрозумілих та ефективних інструментах шифрування [1].

Сучасні алгоритми, такі як AES, RSA, а також наскрізне шифрування (E2EE), активно застосовуються у повсякденному житті - від захисту особистого листування до корпоративних інформаційних систем. Однак для багатьох користувачів принципи роботи криптографії залишаються складними для розуміння, що створює потребу у навчальних програмних рішеннях, які наочно демонструють процес шифрування.

Мета роботи полягає у накопиченні, аналізі та визначенні високорівневих потреб у сучасних методах шифрування текстових даних, дослідження можливостей існуючих криптографічних систем та створення зручної, ефективною і інтуїтивно зрозумілої програмної системи для демонстрації та використання

алгоритмів шифрування. Навчитись формулювати функціональні вимоги до власної розробленої системи, реалізувати практичні приклади класичних і сучасних алгоритмів шифрування (RSA, симетричні методи тощо), а також забезпечити зрозумілий інтерфейс для безпечної роботи з текстовими даними. Документ має акцентувати увагу на потребах користувачів і розробників, які працюють із захистом інформації на різних платформах (ПК, Web-системи), та пояснювати актуальність використання криптографії у сучасних інформаційних системах.

У програмній системі реалізовано кілька алгоритмів шифрування, зокрема: шифр Цезаря; шифр Віженера; шифр ADFGX / ADFGVX; алгоритм RSA. Для забезпечення гнучкості та розширюваності системи використано підхід об'єктно-орієнтованого програмування. Кожен алгоритм шифрування реалізовано у вигляді окремого класу, який містить методи для виконання операцій шифрування та дешифрування. Процес роботи програми можна описати таким чином: обирається алгоритм шифрування зі списку доступних методів; користувач вводить текст, який необхідно зашифрувати або розшифрувати; вводяться потрібні параметри шифрування; програма виконує відповідний алгоритм і відображає результат.

Архітектура програмної системи “Criptify” побудована за модульним принципом, що забезпечує ізоляцію логіки шифрування від інтерфейсу користувача та спрощує подальше розширення функціональності. Архітектура системи включає окремі компоненти для взаємодії з користувачем, виконання криптографічних операцій, перевірки параметрів та тестування алгоритмів. Основними компонентами архітектури є: User Interface - забезпечує взаємодію користувача із системою; CipherService - централізовано керує процесами шифрування та дешифрування; CipherAlgorithm - базовий абстрактний клас для реалізації алгоритмів; RSAService - реалізація асиметричного шифрування RSA;

TestingModule - модуль перевірки коректності роботи алгоритмів; KeyValidator - централізована перевірка коректності ключів та параметрів.

Вхідні та вихідні дані системи. Вхідними даними системи є: текст користувача для шифрування або дешифрування; ключ або параметри алгоритму; обраний алгоритм шифрування. Текст вводиться через інтерфейс користувача у форматі UTF-8 та може містити до 10 000 символів. Перед виконанням операцій усі параметри проходять перевірку на коректність. *Результатом роботи системи* є: зашифрований текст; розшифрований текст; повідомлення про помилки або результати перевірки алгоритмів. Отримані результати можуть відображатися на екрані, копіюватися або зберігатися у файл.

Усі алгоритми шифрування реалізовані як окремі класи, що дозволяє ізолювати криптографічну логіку та спростити підтримку програмного коду. Для реалізації симетричних алгоритмів (Cesar, Vigenere, ADFGX, ADFGVX) використовується централізований сервіс CipherService, який виконує вибір необхідного алгоритму залежно від параметрів користувача [2]. Для алгоритмів ADFGX та ADFGVX було реалізовано окремий допоміжний клас ADFGXHelper, який відповідає за колонкову перестановку символів. Такий підхід дозволив винести повторювану логіку в окремий компонент та уникнути дублювання коду. Перевірка коректності ключів виконується за допомогою класу KeyValidator, який забезпечує централізовану валідацію параметрів для всіх алгоритмів шифрування. У разі введення некоректних значень система генерує відповідні винятки та повідомлення про помилки.

Реалізація RSA. Для генерації ключів використовується клас KeyPairGenerator, а для виконання операцій шифрування та дешифрування - клас Cipher. Криптографічна логіка RSA винесена в окремий сервісний клас RSAService, який містить статичні методи для: генерації пари ключів; шифрування повідомлень; дешифрування даних; відновлення ключів із Base64-

представлення. Для демонстрації роботи RSA у системі також реалізовано модуль симуляції обміну повідомленнями між користувачами. Взаємодія між користувачами організована через клас MessageBus, який реалізований за шаблоном Singleton та забезпечує централізоване керування повідомленнями.

Для реалізації програмної системи було обрано мову програмування Java. Основними причинами вибору стали: кросплатформеність завдяки JVM; підтримка об'єктно-орієнтованого програмування; наявність стандартних криптографічних бібліотек; автоматичне керування пам'яттю; зручність підтримки та масштабування системи. У порівнянні з C++ та Python, Java забезпечує баланс між продуктивністю, безпекою та швидкістю розробки, що є важливим для створення навчальної криптографічної системи.

Інтерфейс реалізований з урахуванням простоти використання та мінімальної кількості дій користувача. Було створено окремі екрани для: головного меню, шифрування та дешифрування тексту, симуляції RSA, налаштувань, логів та історії шифрування [3]. Під час проєктування використовувались прототипи екранів для оцінки логіки взаємодії користувача із системою. *Головне меню та навігація* - у фінальній версії системи реалізовано окреме головне меню, через яке користувач може перейти до екрана шифрування, симуляції RSA, налаштувань, логгера та історії шифрування. Це дозволило зробити структуру програми більш зрозумілою.

Реалізація *екрана шифрування* - основний екран містить: поле введення тексту, вибір алгоритму, режим “Encrypt / Decrypt”, поле ключа та область результату [5]. У фінальній версії реалізовано динамічне оновлення результату та спадний список вибору режиму роботи, що спростило інтерфейс і зробило його зручнішим. *Демонстрація RSA* - у системі реалізовано окремий модуль Simulation для демонстрації асиметричного шифрування RSA. Модуль підтримує створення карток користувачів, генерацію ключів, шифрування, дешифрування та передачу

повідомлень між користувачами. Для взаємодії між компонентами використовується клас MessageBus. *Logger* - у системі реалізовано механізм логування для фіксації дій користувача та роботи програми. Логи зберігаються у файлі Logger.txt та містять дату, час, тип операції, алгоритм, параметри та статус виконання. Використовуються категорії логів: SYSTEM, SIMULATION, CIPHER.

Історія шифрування - реалізовано окремий екран для перегляду попередніх операцій шифрування [6]. У фінальній версії покращено структуру відображення записів, форматування та оновлення даних. Налаштування системи - реалізовано окремий екран налаштувань для зміни параметрів роботи системи та подальшого розширення конфігурації програми. *Особливості UX/UI* - під час розробки основна увага приділялась інтуїтивності інтерфейсу, швидкому доступу до функцій та читабельності результатів. Основна операція шифрування виконується не більше ніж за 3 дії користувача [7].

Тестування проводилось з метою перевірки коректності реалізованих алгоритмів, відповідності функціональним вимогам та оцінки стабільності роботи системи. У процесі розробки використовувались як функціональні, так і нефункціональні методи тестування. *Функціональне тестування* - спрямоване на перевірку відповідності реалізованих функцій заданим вимогам системи. Було протестовано: введення тексту; вибір алгоритму; введення ключа; шифрування тексту; дешифрування тексту; відображення результату. Для кожної функції перевірялась правильність роботи при стандартних та граничних сценаріях використання. *Нефункціональне тестування* - дозволило оцінити якість роботи програмної системи. Основна увага приділялась: продуктивності; надійності; стабільності; зручності використання; безпеці. Було встановлено, що час виконання операцій шифрування та дешифрування для текстів обсягом до 10 000 символів не перевищує 2–3 секунд. Також перевірялась коректна обробка:

неправильних ключів; пустих рядків; великих текстів; некоректних символів; помилкових параметрів алгоритмів.

Тестування методом “чорного ящика”. Перевірялась функціональність системи без аналізу внутрішньої структури коду. Основна увага приділялась правильності результатів шифрування та дешифрування. Було протестовано: коректність роботи алгоритмів; правильність відображення результатів; реакцію системи на неправильний ввід; стабільність роботи інтерфейсу. *Тестування методом “білого ящика”* - виконувалось на рівні програмного коду. Перевірялись: цикли; умовні оператори; обробка винятків; граничні значення; повторювані ділянки коду. Особлива увага приділялась перевірці алгоритмів шифрування та роботі класу KeyValidator. Для перевірки коректності реалізації вимог використовувались: аналіз документації; UML-діаграми; сценарії типу “вхід - обробка - вихід”; прототипування інтерфейсу.

Анотація. У сучасному цифровому середовищі обсяг інформації, яка передається через мережу Інтернет, постійно зростає. Повідомлення, файли, банківські операції, персональні дані та службова інформація щоденно обробляються різними програмними системами, тому питання їхнього захисту стає особливо важливим. Саме криптографія та методи шифрування дозволяють забезпечити конфіденційність, цілісність і безпечну передачу даних між користувачами. У рамках дослідження здійснено проєктування і практичну реалізацію інформаційної системи, яка дозволяє демонструвати роботу різних алгоритмів шифрування текстових даних. Програма реалізована на Java із використанням принципів об’єктно-орієнтованого програмування. Основною метою реалізації є створення зручного інструменту, який дозволяє користувачеві вводити текстові дані, обирати алгоритм шифрування та отримувати результат шифрування або дешифрування у режимі реального часу. Практична значущість роботи полягає у створенні програмної системи, яка дозволяє наочно демонструвати роботу різних алгоритмів шифрування, що може бути використана як навчальний інструмент для студентів та користувачів, які вивчають основи криптографії.

Ключові слова: криптографія, дослідницьке навчання, варіативні моделі, математичне моделювання, об’єктно орієнтованого програмування, комп’ютерне моделювання, варіативні моделі, інформаційна система, алгоритми шифрування, AI, КОМСДН, Cesar, Vigenere, ADFGX, ADFGVX.

Література

1. Hrybiuk O. Experience in Implementing Computer-Oriented Methodological Systems of Natural Science and Mathematics Research Learning in Ukrainian Educational Institutions. In: Machado J., Soares F. (eds) *Innovations in Mechatronics Engineering. Lecture Notes in Mechanical Engineering*, 2022. P. 55-68. Springer, Cham Online.
2. Hrybiuk Olena. Engineering in Educational Institutions: Standards for Arduino Robots as an Opportunity to Occupy an Important Niche in Educational Robotics in the Context of Manufacturing 4.0, in: *Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*. Volume 27-32, 2020. P. 770-785.
3. Hrybiuk O., Vedishcheva O. Experimental Teaching of Robotics in the Context of Manufacturing 4.0: Effective Use of Modules of the Model Program of Environmental Research Teaching in the Working Process of the Centers “Clever”. In: , et al. *Innovations in Mechatronics Engineering II. 2022. Lecture Notes in Mechanical Engineering*. Springer, Cham. 2022.
4. Hrybiuk O. Improvement of the Educational Process by the Creation of Centers for Intellectual Development and Scientific and Technical Creativity. In: Hamrol A., Kujawińska A., Barraza M. (eds). *Advances in Manufacturing II. MANUFACTURING 2019. Lecture Notes in Mechanical Engineering*, 2019. P. 370-382. Springer, Cham Online.
5. Hrybiuk O. Problems of expert evaluation in terms of the use of variative models of a computer-oriented learning environment of mathematical and natural science disciplines in schools, [w:] *Zeszyty Naukowe Politechniki Poznańskiej. Seria: Organizacja i Zarządzanie*, Zeszyt Nr 79, Poznań: Wydawnictwo Politechniki Poznańskiej (WPP), 2019. P. 101-119.
6. Гриб'юк О.О. Імерсивні технології в освіті: особливості когнітивного розвитку дитини у віртуальному середовищі в процесі дослідницького навчання. *Сучасні інформаційні технології та інноваційні методики навчання в підготовці фахівців: методологія, теорія, досвід, проблеми: збірник наукових праць*. Київ-Вінниця: ТОВ Фірма «Планер», 2020.
7. Гриб'юк О.О. Комп'ютерне моделювання та робототехніка в навчально-виховному процесі сучасного навчального закладу. // *Сьома міжнародна науково-практична конференція FOSS Lviv-2017: Збірник наукових праць*, 27-30 квітня 2017 року, м. Львів. С. 38-43. ISBN 978-966-2598-86-5.

УДК 004.942:[327

Ключище І. С. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О
Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-технічний університет імені академіка Юрія Бугая» (Київ)

МАТЕМАТИЧНІ МЕТОДИ ТА ТЕХНОЛОГІЇ ШТУЧНОГО ІНТЕЛЕКТУ В ІНФОРМАЦІЙНО-АНАЛІТИЧНИХ СИСТЕМАХ ПРОГНОЗУВАННЯ РЕСУРСІВ ПІДПРИЄМСТВА

У сучасних умовах стрімкої цифровізації бізнесу наявність якісного веб представництва є критичною необхідністю. Проте процес традиційної веб

розробки залишається тривалим та фінансово затратним для малого та середнього бізнесу. Існуючі на ринку класичні системи управління контентом (CMS) та візуальні No-Code платформи дозволяють пришвидшити цей процес, але створюють нові проблеми: вони генерують надлишковий код (деградацію DOM), обмежують масштабування бізнес-логіки та формують жорстку залежність від постачальника (vendor lock-in) через використання спільного серверного середовища. З іншого боку, поява новітніх генеративних інструментів на базі штучного інтелекту відкрила нові можливості для автоматизації написання коду [1]. Однак ці інструменти орієнтовані переважно на інженерів, оскільки великі мовні моделі (LLM) часто генерують критичні помилки та не вирішують питань інфраструктурного розгортання і контейнеризації веб додатків. З огляду на це, створення гібридної інформаційно-аналітичної системи, яка б поєднувала простоту ШІ-генерації з надійністю жорстких декларативних контрактів та автоматизованою ізольованою інфраструктурою, є актуальним науково-практичним завданням.

Мета дослідження полягає у розробленні та програмній реалізації інформаційно-аналітичної системи (SaaS-платформи TouchCraft AI) для автоматизованого створення, розгортання та оркестрації веб додатків на базі технологій штучного інтелекту та концепції *Server-Driven UI* [2]. Для досягнення поставленої мети було визначено та вирішено такі завдання дослідження: Провести системний аналіз існуючих підходів до автоматизації веб розробки, виявити архітектурні обмеження No-Code платформ та AI-генераторів коду. Сформулювати функціональні та нефункціональні вимоги до інформаційної системи та спроектувати її логічну і фізичну архітектуру бази даних. Розробити серверну підсистему (*AIBuilderEngine*), що реалізує механізм жорсткої Pydantic-валідації для усунення «галюцинацій» ШІ та перетворення неструктурованих запитів у суворі декларативні JSON-контракти. Створити клієнтський односторінковий

додаток (*FrontendBuilder*) на базі фреймворку *Vue 3* для візуального управління інфраструктурою та динамічного рендерингу *еталонних UI-компонентів*. Розробити інфраструктурний модуль (*Deployer*) для автоматизованої збірки, ізольованої *Docker-контейнеризації* [3] клієнтських проєктів та динамічної маршрутизації через *Nginx Proxy Manager*. Провести комплексну верифікацію, модульне та інтеграційне тестування розробленої системи, а також оцінити продуктивність згенерованих артефактів.

На базі спроектованої архітектури здійснюється перехід від інженерного концепту до розгортання працездатного B2B-продукту (SaaS-платформи). Процес розробки розділено на два паралельні напрямки. Перший – створення реактивного клієнтського інтерфейсу (*FrontendBuilder*) за допомогою фреймворку *Vue 3*. Другий – розробка асинхронного серверного ядра (*AIBuilderEngine*) на базі *FastAPI*. Головний акцент у розділі зроблено на програмній реалізації унікальних алгоритмів платформи [4]. До них належать: інтеграція з мовними моделями (*Gemini API*) для побудови декларативного *UI*, механізми суворої *Pydantic-валідації* та низькорівнева генерація вихідного коду (*артефактів*). Окремо висвітлено інфраструктурний код – автоматизовану оркестрацію ізольованих *Docker-контейнерів* та налаштування *HTTP-маршрутизації* через *Nginx Proxy Manager*.

Завершальним етапом є технічна верифікація системи. Вона включає навантажувальне тестування інфраструктури та профілювання продуктивності згенерованих веб додатків за ключовими метриками *Google Core Web Vitals* (зокрема *LCP, CLS та TTFB*) [5]. Отримані результати на практиці підтверджують відповідність розробленого рішення заявленим функціональним та нефункціональним вимогам.

Програмна реалізація клієнтської підсистеми *TouchCraft AI* виконана як *Single Page Application (SPA)* на базі фреймворку *Vue 3*. Використання

Composition API та архітектури SFC (Single File Components) дозволяє декомпонувати складну бізнес-логіку на незалежні реактивні модулі [6]. Клієнтська частина виступає інтелектуальним прошарком, що абстрагує користувача від складності хмарної інфраструктури та ШІ-оркестрації. Життєвий цикл взаємодії користувача з інтерфейсом системи (рис. 1) відображає переходи від етапу авторизації до фінального розгортання та управління ресурсами [7].

Автентифікація та механізм контролю доступу (Waiting Room). Сесія ініціюється через Auth0 (OAuth 2.0). Після верифікації JWT-токена [7] спрацьовує модуль WaitingRoom.vue, який виконує роль Gatekeeper-фільтра. Це інженерне рішення запобігає нецільовому використанню ресурсів LLM неавторизованими або неактивованими користувачами.

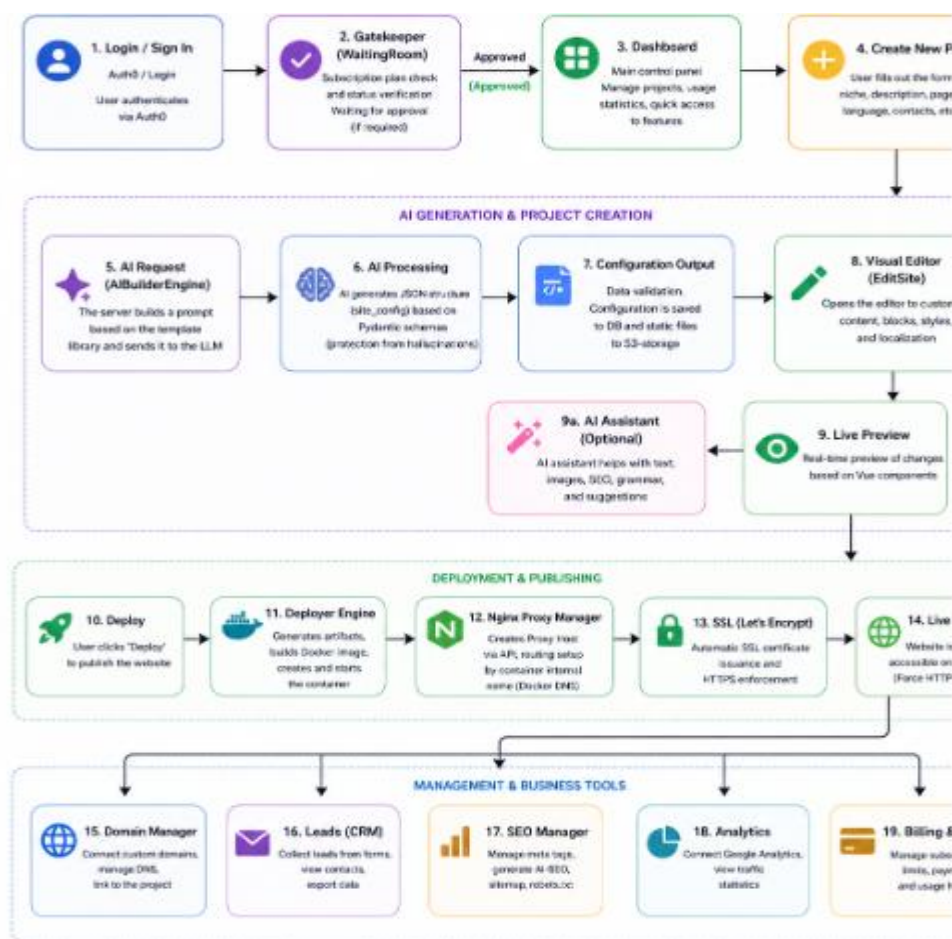


Рис. 1. Діаграма процесів інтерфейсу користувача платформи
Джерело: власне опрацювання

Логіка перевірки статусу доступу в WaitingRoom.vue

```
const checkStatus = async () => {
  try {
    const response = await api.get('/user/me');
    // Якщо користувач активований адміністратором – перехід на Dashboard
    if (response.data.is_active) {
      router.push('/dashboard');
    }
  } catch (e) {
    console.error("Access denied or server error", e);
  }
};
```

Ініціалізація та III-генерація проєкту (CreateSite.vue). Після активації користувач отримує доступ до майстра створення проєктів. Модуль CreateSite.vue реалізує Wizard-інтерфейс, де ключовим завданням є формування структурованого промпту для III-ядра. Інтерфейс налаштування параметрів генерації, що включає вибір ніші та мов локалізації наведено нижче (рис. 2).

Рис. 2. Інтерфейс ініціалізації AI-генерації вебдодатка
Джерело: власне опрацювання

Ядро візуального редагування та Server-Driven UI (EditSite). Модуль EditSite.vue є центральним вузлом системи, що реалізує концепцію Server-Driven UI (SDUI) [2]. Структура редактора розділена на EditorSidebar.vue (управління станом) та EditorPreview.vue (візуалізація). Програмно рендеринг реалізовано через ітерацію по об'єкту site_config. Така архітектура дозволяє змінювати

структуру сайту (додавати блоки, змінювати порядок) без модифікації коду фронтенду, оскільки логіка рендерингу цілком залежить від даних, отриманих від сервера.

Управління життєвим циклом та інфраструктурою (Dashboard & DomainManager). Модуль Dashboard.vue забезпечує контроль станів Docker-контейнерів. Кожна операція (Start/Stop) є асинхронним викликом, що взаємодіє з системним Deployer-модулем бекенду. *Перехід до повноцінної багатосторінковості:* трансформація поточної архітектури Landing Page у систему підтримки необмеженої кількості сторінок. Це дозволить як ШІ, так і користувачу створювати складні ієрархічні структури (окремі сторінки «Про нас», «Послуги», «Блог», «Контакти» тощо), виходячи за межі поточного формату односторінкового додатка з каталогом.

Впровадження системи глибокої кастомізації стилів: розробка інструментарію для детального керування візуальною ідентичністю сайту. Це передбачає надання як ШІ-асистенту, так і людині можливості тонкого налаштування CSS-змінних, розширених палітр, сіток та типографіки для кожного окремого блоку чи сторінки. *Міграція інфраструктури на кластери Kubernetes:* перехід від монолітного Docker-хостингу до оркестрації в K8s для забезпечення автоматичного масштабування, високої доступності та глобальної відмовостійкості системи.

Розширення бібліотеки компонентів та E-commerce модулів: впровадження повноцінного функціоналу інтернет-магазинів із кошиком, кабінетом покупця та інтеграцією платіжних шлюзів (Stripe, PayPal). *Досягнення еталонної якості коду (Lighthouse 100/100):* подальша оптимізація шаблонів для усунення зауважень щодо валідності robots.txt та повної підтримки стандартів доступності WCAG, що гарантує інклюзивність згенерованих ресурсів. *Інтелектуальна аналітика та кешування:* впровадження механізмів глибокої AI-аналітики поведінки

відвідувачів для автоматичного надання порад щодо покращення конверсії, а також використання багаторівневого кешування для миттєвого завантаження контенту в будь-якій точці світу.

Практична реалізація підтвердила життєздатність обраного декларативно-генеративного підходу, а створена платформа TouchCraft AI заклала потужний інженерний фундамент для подальшого масштабування у комерційному сегменті веб технологій.

Анотація. У дослідженні пропонується вирішення актуального науково-практичного завдання – розроблення інформаційно-аналітичної системи автоматизованого створення та оркестрації вебдодатків з використанням технологій штучного інтелекту (TouchCraft AI). Розроблена платформа концептуально змінює підхід до веброзробки, дозволяючи отримувати готові, багатомовні та інфраструктурно ізольовані сайти за лічені хвилини замість тижнів класичної розробки. У ході виконання роботи було отримано наступні результати: Проведено системний аналіз недоліків існуючих конструкторів сайтів та AI-генераторів, на основі якого обґрунтовано вибір концепції *Server-Driven UI (SDUI)*. Спроектовано та реалізовано архітектуру, що складається з клієнтського SPA-додатка (*Vue 3*) та високопродуктивного асинхронного бекенду (*FastAPI*). Розроблено механізм оркестрації інфраструктури, який упаковує клієнтські проєкти в ізольовані *Docker*-контейнери та здійснює автоматичну маршрутизацію трафіку з випуском *SSL*-сертифікатів. Успішно верифіковано роботу системи шляхом модульного тестування (*11/11* успішних тестів *Pydantic*-контрактів) та аудиту *Google Lighthouse* (продуктивність понад 95 балів).

Ключові слова: геймдизайн орієнтований підхід, дослідницьке навчання, варіативні моделі, комп'ютерне моделювання, КОМСДН, інтелектуальна система, аналітична система, математичне моделювання, *SaaS*, TouchCraft AI.

Література

1. Hrybiuk O. Problems of expert evaluation in terms of the use of variative models of a computer-oriented learning environment of mathematical and natural science disciplines in schools, [w:] *Zeszyty Naukowe Politechniki Poznańskiej. Seria: Organizacja i Zarządzanie*, Zeszyt Nr 79, Poznań: Wydawnictwo Politechniki Poznańskiej (WPP), 2019. P. 101-119.
2. Hrybiuk, O., Kant, G.S. CleverCOMSRL: Implementation of an AI Computer-Aided Design System in the Context of the Cognitive Science Paradigm for the Research Training Process. In: Machado, J., et al. *Innovations in Mechatronics Engineering III. icieng 2024. Lecture Notes in Mechanical Engineering*. Springer, Cham. 2024. P. 351-362. https://doi.org/10.1007/978-3-031-61575-7_32.
3. Гриб'юк О.О. Комп'ютерне моделювання та робототехніка в навчально-виховному процесі сучасного навчального закладу. // *Сьома міжнародна науково-практична*

конференція FOSS Lviv-2017: Збірник наукових праць, 27-30 квітня 2017 року, м. Львів. С. 38-43. ISBN 978-966-2598-86-5.

4. Hrybiuk Olena. Engineering in Educational Institutions: Standards for Arduino Robots as an Opportunity to Occupy an Important Niche in Educational Robotics in the Context of Manufacturing 4.0, in: *Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*. Volume 27-32, 2020. P. 770-785.

5. Hrybiuk, O. Improvement of the educational process by the creation of centers for intellectual development and scientific and technical creativity. In: Hamrol, A., Kujawińska, A., Barraza, M.F.S. (eds.) *MANUFACTURING 2019*. LNME, 2019. P. 370–382. Springer, Cham. https://doi.org/10.1007/978-3-030-18789-7_31

6. Hrybiuk, O.O. Research learning of the natural science and mathematics cycle using computer-oriented methodological systems. Monograph. Drahomanov NPU, Kyiv, 2019. P. 277–325.

7. Khang A., Abdullayev V., Hrybiuk O., & Shukla A.K. (Eds.). *Computer Vision and AI-Integrated IoT Technologies in the Medical Ecosystem* (1st ed.). CRC Press. 2024. <https://doi.org/10.1201/9781003429609>.

УДК 004.942:[327

Корнілов Р. В (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О
Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-технічний університет імені академіка Юрія Бугая» (Київ)

МАТЕМАТИЧНІ ОСНОВИ ПРОЄКТУВАННЯ І РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТРАНСПОРТНОЇ КОМПАНІЇ

Актуальність дослідження. Розвиток веб-технологій, хмарних сервісів і клієнт–серверних архітектур відкриває можливості для створення комплексних інформаційних систем транспортних компаній, які забезпечують централізоване зберігання даних, автоматизацію бізнес-процесів, інтеграцію з зовнішніми сервісами картографії, платіжними системами та аналітичними платформами. Такі системи дають змогу автоматизувати приймання й обробку замовлень, планування маршрутів, керування автопарком і персоналом, контроль виконання рейсів і формування управлінської звітності.

Готові рішення, що присутні на ринку, не завжди повною мірою враховують специфіку конкретної транспортної компанії: географію обслуговування,

структуру автопарку, внутрішні бізнес-правила, особливості тарифікації та вимоги до звітності. Часто такі системи є закритими, складними для адаптації, вимагають значних фінансових витрат на впровадження та супровід. Це створює актуальну задачу розроблення власної інформаційної системи, орієнтованої на потреби конкретного перевізника та побудованої з урахуванням принципів модульності, масштабованості й відкритості.

Актуальність роботи зумовлена необхідністю підвищення ефективності обробки замовлень на перевезення, зменшення кількості ручних операцій, оперативного доступу до даних про клієнтів, транспортні засоби та виконані рейси. *Мета дослідження* полягає у здійсненні проектування і розроблення веб-орієнтованої інформаційної системи транспортної компанії (ІСТК), яка автоматизує основні бізнес-процеси (приймання замовлень, планування та виконання поїздок, облік клієнтів, транспортних засобів і водіїв), забезпечує прозорість даних і підтримує формування аналітичної звітності для керівництва.

У ході дослідження було вивчено актуальне питання вибору архітектури, інструментів та алгоритмів для побудови систем. У дослідженні проведено аналіз предметної області та існуючих програмних рішень для транспортних компаній, визначено їхні переваги та недоліки [1]. Сформульовано вимоги до інформаційної системи, обґрунтовано вибір архітектури клієнт–серверного веб-застосування та технологічного стеку реалізації. Розроблено структурну схему системи, здійснено проектування бази даних для зберігання відомостей про користувачів, замовлення, транспортні засоби та статуси поїздок.

Для розробленої системи (рис. 1) застосовано підхід «тонкий клієнт – потужний сервер». Основна логіка обробки даних зосереджена на стороні сервера, а клієнт переважно відображає інформацію та ініціює запити. Це спрощує оновлення програмного забезпечення, підвищує єдність бізнес-логіки та полегшує централізований контроль безпеки.

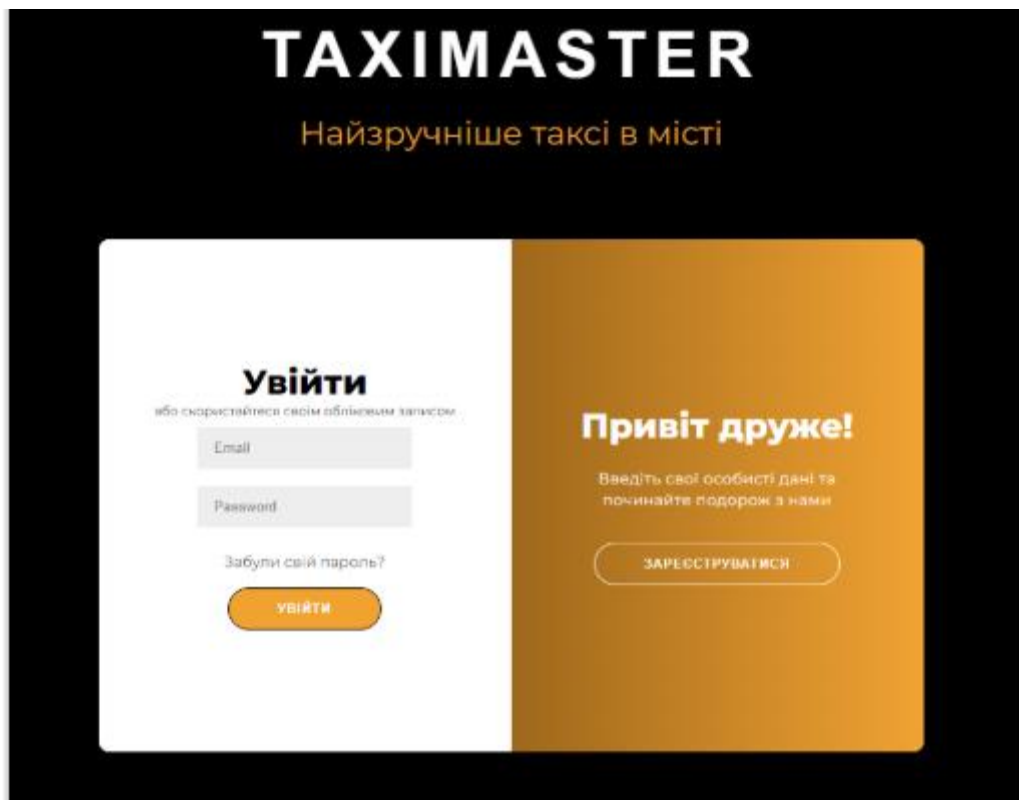


Рис. 1. Інтерфейс головної сторінки ІСТК
Джерело: опрацювання власне

Обмін даними між клієнтською та серверною частинами здійснюється через протокол *HTTPS*, який забезпечує шифрування трафіку та захист від перехоплення облікових даних [2]. Для реалізації прикладного інтерфейсу використовується REST-підхід: кожна операція системи (створення замовлення, перегляд історії поїздки, зміна статусу замовлення, авторизація користувача) відповідає певному *HTTP-методу (GET, POST, PUT, DELETE)* та *URL-ресурсу*. Дані передаються у форматі *JSON*, що є компактним, людино- та машиночитним [3].

Серверна частина інформаційної системи реалізується на платформі *Node.js* з використанням фреймворку *Express*. Такий вибір зумовлено: неблокувальною, подійно-орієнтованою моделлю введення/виведення в *Node.js*, що дає змогу обробляти велику кількість коротких запитів (створення та оновлення замовлень, розрахунок вартості тощо); наявністю широкої екосистеми бібліотек для роботи з *HTTP-запитами*, кодуванням і декодуванням *JSON*, авторизацією користувачів,

підключенням до баз даних; можливістю використання однієї мови програмування (*JavaScript*) як на клієнті, так і на сервері, що зменшує трудомісткість супроводу коду [4]. Для реалізації функцій автентифікації, роботи з cookies та захисту сесій у серверній частині можуть використовуватися спеціалізовані пакети (наприклад, для хешування паролів, формування *JSON Web Token*, надсилання електронних листів підтвердження).

Клієнтську частину побудовано на стандартних веб-технологіях *HTML5*, *CSS* та *JavaScript*. *HTML5* використовується для структуризації сторінок (форми реєстрації, сторінка замовлення, особистий кабінет), *CSS* застосовується для стилізації та адаптивного відображення інтерфейсу на різних пристроях, *JavaScript* використовується для організації динамічної взаємодії з сервером без повного перезавантаження сторінки (відправлення форм, оновлення списку замовлень, відображення розрахованої вартості поїздки тощо).

Рівень даних реалізовано на основі реляційної *СКБД MySQL*. Вона забезпечує: зберігання даних у взаємопов'язаних таблицях (користувачі, замовлення, транспортні засоби, тарифи); підтримку первинних та зовнішніх ключів для забезпечення цілісності зв'язків; можливість використання індексів для прискорення пошуку за ключовими полями (ID користувача, дата створення замовлення тощо); транзакційну обробку критичних операцій (створення або скасування замовлення, зміна статусу, оновлення балансу/оплати). Логічна модель бази даних використовується так, щоб уникати надлишкового дублювання інформації, дотримуючись нормалізації до не нижче третьої нормальної форми [5]. Це дозволяє зменшити обсяг збережених даних, спростити підтримку й оновлення довідників та забезпечити узгодженість відомостей у різних підсистемах.

Анотація. На основі проведених досліджень здійснено проектування і розроблено веб-застосування, що включає модулі реєстрації та авторизації користувачів, створення та перегляду замовлень, відстеження їхнього стану, роботу з історією поїздок. Серверна частина побудована на основі платформи *Node.js* з використанням фреймворку *Express*, для зберігання даних застосовано

реляційну базу даних MySQL. Проведено тестування основних функціональних можливостей системи та проаналізовано результати її роботи. Запропонована інформаційна система забезпечує централізоване зберігання даних, підвищує прозорість та керованість процесів у транспортній компанії, скорочує час обробки замовлень і може бути використана як основа для подальшого розширення функціональності (додавання аналітичних модулів, інтеграції з платіжними сервісами). Перспективним напрямом розвитку є інтеграція з окремим інтелектуальним модулем, який може реалізовувати алгоритми машинного навчання для прогнозування часу прибуття автомобіля, оцінки ймовірної тривалості поїздки або адаптивного налаштування коефіцієнта Кр залежно від попиту. Такий модуль доцільно реалізувати як автономний сервіс, доступний через REST-API. Серверна частина системи звертається до нього за прогнозом, а отриманий результат використовується у процесі розрахунку тарифу та відображенні інформації клієнту.

Ключові слова: дослідницьке навчання, варіативні моделі, CleverCOMSRL, комп'ютерне моделювання, КОМСДН, інформаційна система, математичне моделювання, транспортна компанія, AI.

Література

1. Hrybiuk, O., Kant, G.S. CleverCOMSRL: Implementation of an AI Computer-Aided Design System in the Context of the Cognitive Science Paradigm for the Research Training Process. In: Machado, J., et al. *Innovations in Mechatronics Engineering III*. icieng 2024. *Lecture Notes in Mechanical Engineering*. Springer, Cham. 2024. P. 351-362. https://doi.org/10.1007/978-3-031-61575-7_32.
2. Hrybiuk, O. Improvement of the educational process by the creation of centers for intellectual development and scientific and technical creativity. In: Hamrol, A., Kujawińska, A., Barraza, M.F.S. (eds.) *MANUFACTURING 2019*. LNME, 2019. P. 370–382. Springer, Cham. https://doi.org/10.1007/978-3-030-18789-7_31.
3. Hrybiuk, O.O. Research learning of the natural science and mathematics cycle using computer-oriented methodological systems. Monograph. Drahomanov NPU, Kyiv, 2019. P. 277–325.
4. Khang A., Abdullayev V., Hrybiuk O., & Shukla A.K. (Eds.). *Computer Vision and AI-Integrated IoT Technologies in the Medical Ecosystem* (1st ed.). CRC Press. 2024. <https://doi.org/10.1201/9781003429609>.
5. Hwei-Jen Lin, Shu-Yi Wang, Shwu-Huey Yen and Yang-Ta Kao, "Face Detection Based on Skin Color Segmentation and Neural Network," *2005 International Conference on Neural Networks and Brain*, Beijing, China, 2005. P. 1144-1149. <https://doi.org/10.1109/ICNNB.2005.1614818>.

Костенко Є.О. (здобувач бакалаврату, 1 курс),
Старіков Н.Є. (здобувач бакалаврату, 1 курс),
Науковий керівник – ст. викл. Нестеренко В. О.
Харківський національний автомобільно-дорожній університет

ПРОСТІШИЙ МЕТОД ПРОГНОЗУ ТА КОРИГУВАННЯ ПРИ НАБЛИЖЕНОМУ ІНТЕГРУВАННЮ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ ПЕРШОГО ПОРЯДКУ

Більшість задач моделювання динамічних систем зводяться до розв’язання задачі Коші: $y' = f(x, y), \quad y(x, y) = y_0$

Серед широкого спектру чисельних методів особливе місце займають методи прогнозу та корегування (predictor-corrector), які поєднують простоту явних схем зі стійкістю та точністю неявних.

Однак класична реалізація цих методів часто передбачає ітераційне уточнення на етапі коректора, що збільшує час обчислень. Метою даної роботи є дослідження найпростішої реалізації методу прогнозу-корекції, де корекція виконується лише один раз без ітерацій.

Розглянемо інтегрування на кроці $[x_n, x_{n+1}]$ з кроком h .

Етап прогнозу (предиктор)

Використовуємо явний метод Ейлера (перший порядок точності) для отримання початкового наближення: $y_{n+1}^0 = y_n + h \cdot f(x_n, y_n)$

Етап корекції (коректор)

Застосовуємо неявну формулу трапецій (метод Кранка-Ніколсона), яка має другий порядок точності:

$$y_n = y_n + \frac{h}{2} [f(x_{n+1}, y_{n+1}) + f(x_{n+1}, y_{n+1}^0)].$$

Спрощення: Замість розв’язування нелінійного рівняння відносно y_{n+1} (що вимагало б ітерацій), ми підставляємо значення y_{n+1}^0 у праву частину. Таким чином, коректор стає явним і обчислюється за одну формулу.

Локальна похибка. Розклад в ряд Тейлора показує, що локальна похибка методу становить $O(h^3)$, що відповідає другому порядку точності. Це суттєво вище, ніж у класичного методу Ейлера ($O(h^2)$), і досягається без ускладнення алгоритму.

Стійкість. Застосування методу до тестового рівняння Далквіста $y' = \lambda y$ (де λ — комплексне число) дає перехідний множник: $Q(z) = 1 + z + \frac{z^2}{2}$, $z = \lambda h$.

Область абсолютної стійкості методу є ширшою, ніж у явного методу Ейлера, і включає частину лівої півплощини, що робить метод придатним для розв'язання помірно жорстких задач.

ОБЧИСЛЮВАЛЬНА ЕФЕКТИВНІСТЬ

Порівняльна характеристика методів (на один крок інтегрування):

Метод	Кількість обчислень $f(x,y)$	Порядок точності
Явний метод Ейлера	1	1
Неявний метод Ейлера	$1+k$ (інтеграції)	1
Метод прогноз-корекція (запропонований)	2	2
Класичний метод Рунге-Кутти 4-го порядку	4	4

Запропонований метод є оптимальним з точки зору співвідношення "точність / обчислювальна вартість" для задач, де не потрібен високий (вище другого) порядок точності.

Розглянемо задачу: $y' = y - x^2 + 1$, $y(0) = 0.5$
на проміжку $[0, 1]$ з кроком $h = 0.1$. Точний розв'язок:

$$y(x) = (x + 1)^2 - 0.5e^x.$$

Результати обчислень:

x	Точний розв'язок Метод Ейлера	Метод прогноз- корекція (похибка)
0,5	1,8155	0,0283
1,0	3,6409	0,1126

Похибка методу прогноз-корекції на порядок менша за похибку методу Ейлера при однаковому кроці, що підтверджує теоретичні висновки.

Приклад 1. Задача : Маємо рівняння з початковою умовою :

$$y' = y, \quad y(0) = 1$$

Крок: $h = 0.1$

Точний розв'язок: $y = e^x$

Розв'язання. Метод Ейлера: $y_1 = 1 + 0.1 \cdot 1 = 1.1$

Метод прогноз-корекції:

Прогноз: $y_1^{(0)} = 1 + 0.1 \cdot 1 = 1.1$

Корекція: $y_1 = 1 + 0.05(1 + 1.1) = 1.105$

Точне значення: $e^{0.1} \approx 1.10517$

Приклад 2. Задача : Маємо рівняння $y' = x + y$ з початковою умовою:

$$y' = x + y, \quad y(0) = 1$$

Крок: $h = 0.1$

Розв'язання. Метод Ейлера: $y_1 = 1 + 0.1 \cdot (0 + 1) = 1.1$

Метод прогноз-корекції:

Прогноз: $y_1^{(0)} = 1 + 0.1 \cdot 1 = 1.1$

Корекція: $y_1 = 1 + 0.05[(1) + (0.1 + 1.1)]$

$$y_1 = 1 + 0.05 \cdot 2.2 = 1.11$$

ПРАКТИЧНІ РЕКОМЕНДАЦІЇ ТА ВИСНОВКИ

1. Простота реалізації. Алгоритм вимагає лише збереження попереднього значення функції та двох викликів $f(x, y)$ на кроці, що робить його ідеальним для вбудованих систем (мікроконтролерів) та навчальних лабораторних робіт.

2. Підвищення точності. У порівнянні з методом Ейлера, точність зростає в $O(h)$ разів без збільшення кількості обчислень похідної більш ніж удвічі.
3. Обмеження. Метод не рекомендується для дуже жорстких систем (stiff systems), де потрібна А-стійкість, яку забезпечує неявний метод Ейлера або методи з великою областю стійкості.
4. Подальше узагальнення. Даний підхід легко розширюється на системи диференціальних рівнянь першого порядку та на рівняння вищих порядків після зведення до нормальної форми Коші.

Висновки. Запропонований спрощений метод прогнозу-корекції є ефективним інструментом для швидкого та достатньо точного інтегрування диференціальних рівнянь першого порядку. Він займає проміжне положення між простими явними методами першого порядку та складними багатоетапними схемами, забезпечуючи баланс точності, стійкості та обчислювальних витрат. Метод може бути рекомендований для інженерних розрахунків реального часу, де критичними є швидкодія та простота програмної реалізації.

Анотація. У доповіді розглядається спрощений варіант методів прогнозу-корекції (предиктор-коректор) для розв'язання задачі Коші для звичайних диференціальних рівнянь (ЗДР) першого порядку. Запропонований підхід базується на комбінації явного методу Ейлера (прогноз) та неявного методу трапецій (корекція) без використання ітераційного процесу на кроці корекції. Наведено порівняльний аналіз точності, обчислювальної складності та умов стійкості у порівнянні з класичними однокроковими методами.

Продemonстровано, що даний метод забезпечує другий порядок точності при мінімальних обчислювальних витратах, що робить його ефективним для інженерних розрахунків та освітніх цілей.

Література

1. Бахвалов Н.С., Чисельні методи./ Бахвалов Н.С., Жидков Н.П., Кобельков Г.М., - К.: Вища школа, 2011. – 636 с.
2. Форсайт Дж., Машинні методи математичних обчислень /Форсайт Дж., Мальком М., Моулер К., - К.: - Наукова думка, 1980. – 408 с.

Краснокутський Б.В (здобувач бакалаврату, 2 курс),
Науковий керівник – проф. Ємельянова Т. В.
Харківський національний автомобільно-дорожній університет

ЕЛЕМЕНТАРНІ ВИПАДКОВІ ФУНКЦІЇ ЯК СКЛАДОВІ ВИПАДКОВИХ ПРОЦЕСІВ З НЕПЕРЕРВНИМ ЧАСОМ І НЕПЕРЕРВНИМИ СТАНАМИ

Відомо, що в більшості задач технологічної галузі слід визначати фактори, які мають випадковий характер. Випадкові фактори присутні в прикладних задачах як випадкові функції. Одже, при моделюванні будь-якого явища треба враховувати вплив випадкового фактору. Цей вплив описується випадковими функціями.

Побудова математиної моделі явища, що знаходиться під впливом випадкового фактору, потребує визначення його випадкового характеру, вивчення динаміки розвитку та закономірності випадкового явища.

Далі розглядаємо випадкові явища з неперервним часом і неперервними станами. Прикладом такого випадкового явища є зміна напруги електромережи $U(t)$ в виробничому приміщенні.

В більшості випадків випадкові явища характеризуються складною структурою. Одже, випадковий процес можна розглядати як результат дії кілької випадкових збурень. В ряді завдань випадкові процеси зручно уявляти випадкове явище за допомогою елементарних випадкових функцій.

Елементарними випадковими функціями визначаються функції, в яких залежність від аргументу t є не випадковою функцією, випадкові величини входять у вигляді параметрів, що є незалежними від аргументу t .

Приклад 1. Елементарна випадкова функція має вигляд

$X(t) = e^{-Vt}$, ($t > 0$), де V – випадкова величина, розподілена за показниковим законом розподілу з щільністю $f(V) = \lambda e^{-\lambda V}$ ($V > 0$, $\lambda > 0$).

Реалізації функції $X(t)$ представлені на рис.1.

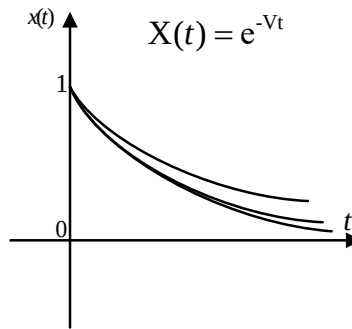


Рис. 1. Реалізації випадкової функції $X(t) = e^{-\lambda t}$

Математичне сподівання випадкового процесу $X(t) = e^{-\lambda t}$

$$M[X(t)] = \int_{-\infty}^{\infty} e^{-\lambda t} f(v) dv = \int_0^{\infty} e^{-\lambda t} \lambda e^{-v\lambda} dv = \frac{\lambda}{\lambda + t}.$$

$$M[X^2(t)] = \int_{-\infty}^{\infty} e^{-2\lambda t} f(v) dv = \int_0^{\infty} e^{-2\lambda t} \lambda e^{-v\lambda} dv = \frac{\lambda}{\lambda + 2t}.$$

$$M[X(t_1)X(t_2)] = \int_{-\infty}^{\infty} e^{-\lambda t_1} e^{-\lambda t_2} f(v) dv = \int_0^{\infty} e^{-\lambda(t_1+t_2)} \lambda e^{-v\lambda} dv = \frac{\lambda}{\lambda + t_1 + t_2}.$$

Кореляційна функція $K_X(t_1, t_2)$ обчислюється за формулою

$$\begin{aligned} K_X(t_1, t_2) &= M[X(t_1)X(t_2)] - m_X(t_1)m_X(t_2) = \\ &= \frac{\lambda}{\lambda + t_1 + t_2} - \frac{\lambda}{\lambda + t_1} \frac{\lambda}{\lambda + t_2} = \frac{\lambda(\lambda + t_1)(\lambda + t_2) - \lambda^2(\lambda + t_1 + t_2)}{(\lambda + t_1 + t_2)(\lambda + t_1)(\lambda + t_2)} = \\ &= \frac{\lambda^3 + \lambda^2(t_1 + t_2) + \lambda(t_1 t_2) - \lambda^3 - \lambda^2(t_1 + t_2)}{(\lambda + t_1 + t_2)(\lambda + t_1)(\lambda + t_2)} = \frac{\lambda t_1 t_2}{(\lambda + t_1 + t_2)(\lambda + t_1)(\lambda + t_2)}. \end{aligned}$$

Дисперсія випадкової функції $X(t)$ пов'язане з кореляційною функцією рівністю

$$D[X(t)] = K[X(t), X(t)].$$

Тому, дисперсія випадкового процесу $X(t)$ визначається кореляційною функцією при $t_1 = t_2 = t$

$$D[X(t)] = K[X(t), X(t)] = \frac{\lambda t^2}{(\lambda + 2t)(\lambda + t)^2}, t > 0.$$

Відомо, що випадкова функція, яка визначає випадкові збурення системи, в більшості випадків залежить від кількох випадкових величин. Ця залежність може бути достатньо складною. Далі приводимо приклад елементарної випадкової функції, залежність якої від випадкових величин є достатньо складною.

Приклад 2. Елементарна випадкова функція має вигляд $X(t) = We^{-Vt}$ ($t > 0$), де W – випадкова величина, розподілена за нормальним розподілом з параметрами m_W, σ_W , випадкова величина V розподілена рівномірно в інтервалі $(0, a)$ ($a > 0$); випадкові величини W, V незалежні.

Щільність розподілу випадкової величини V в інтервалі $(0, a)$ дорівнює $\frac{1}{a}$.

Математичне сподівання для випадкової величини W , розподіленої за нормальним розподілом, дорівнює m_W .

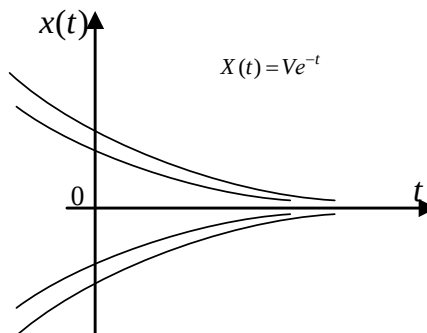


Рис. 2. Реалізації випадкової функції $X(t) = We^{-Vt}$

Математичне сподівання випадкового процесу $X(t) = We^{-Vt}$

$$M[X(t)] = M[We^{-Vt}] = M[W]M[e^{-Vt}] = m_W \frac{1}{a} \int_0^a e^{-vt} dv,$$

$$M[X(t)] = \frac{m_W}{a} \frac{1 - e^{-at}}{t} = \frac{m_W(1 - e^{-at})}{at}, t > 0.$$

Обчислюємо кореляційну функцію $K_X(t_1, t_2)$ з урахуванням незалежності випадкових величин W та V

$$\begin{aligned} K_X(t_1, t_2) &= M[X(t_1)X(t_2)] - m_X(t_1)m_X(t_2) = \\ &= M[We^{-Vt_1}We^{-Vt_2}] - \frac{m_W(1 - e^{-at_1})}{at_1} \frac{m_W(1 - e^{-at_2})}{at_2} = \\ &= M[W^2] M[e^{-V(t_1+t_2)}] - \frac{m_W(1 - e^{-at_1})}{at_1} \frac{m_W(1 - e^{-at_2})}{at_2} = \end{aligned}$$

$$M[W^2] = DW + (MW)^2 = \sigma_W^2 + m_W^2;$$

$$M[e^{-V(t_1+t_2)}] = \frac{1}{a} \int_0^a e^{-v(t_1+t_2)} dv = \frac{1 - e^{-a(t_1+t_2)}}{a(t_1+t_2)};$$

Тому, кореляційна функція $K_X(t_1, t_2)$ може бути записана у вигляді

$$K_X(t_1, t_2) = (\sigma_W^2 + m_W^2) \frac{1 - e^{-a(t_1+t_2)}}{a(t_1+t_2)} - m_W^2 \frac{(1 - e^{-at_1})(1 - e^{-at_2})}{a^2 t_1 t_2}.$$

Визначення кореляційної функції $K_X(t_1, t_2)$ елементарного випадкового процесу $X(t)$ дозволяє записати його дисперсію $D_X(t)$, оскільки виконується рівність

$$D_X(t) = K_X(t, t)$$

$$D_X(t) = K_X(t, t) = (\sigma_W^2 + m_W^2) \frac{1 - e^{-a(t+t)}}{a(t+t)} - m_W^2 \frac{(1 - e^{-at})(1 - e^{-at})}{a^2 tt}.$$

$$D_X(t) = (\sigma_W^2 + m_W^2) \frac{1 - e^{-a^2 t}}{2at} - m_W^2 \frac{(1 - e^{-at})^2}{a^2 t^2}.$$

Примітка. Поведінка елементарної випадкової функції $X(t) = We^{-Vt}$ в різні моменти часу визначається різними випадковими величинами. Так, в околі точки $t=0$ випадкова функція перетворюється у випадкову величину W , розподілену за нормальним законом з параметрами m_W, σ_W . При $t \gg \frac{1}{a}$ вплив випадкової функції $X(t) = We^{-Vt}$ на систему стає несуттєвим та з часом пропадає. Реалізації елементарної випадкової функції $X(t) = We^{-Vt}$ представлені на рис.2.

Анотація. При побудові математичної моделі будь якого явища слід урахувати можливі впливи випадкових факторів, що призводять до випадкових процесів у досліджуємой системі. В ряді задач випадкові процеси можна виразити за допомогою елементарних випадкових функцій, характеристики яких достатньо просто обчислити. В роботі наведено приклади елементарних випадкових функцій, в яких залежність від аргументу t є не випадковою функцією, випадкові величини входять у вигляді параметрів, що є незалежними від аргументу t .

Ключові слова: елементарна випадкова функція, рівномірний закон розподілу випадкової величини, показниковий закон розподілу випадкової величини.

Література

1. Несвіт М.І., Поклонський Є.В.. Випадкові процеси: навчально-методичний посібник. – Х. : ХНУБА, 2013 – 51 с. . Теорія випадкових процесів:
2. Задачі для самостійної роботи [Електронний ресурс] : навч. посіб. для студ. спеціальності 171 «Електроніка»/ КПІ ім. Ігоря Сікорського ; уклад.: О.В. Гармаш. – Електронні текстові дані (1 файл: 1,771 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 44 с.

Красько К. В (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О
*Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-
технічний університет імені академіка Юрія Бугая» (Київ)*

МАТЕМАТИЧНІ МОДЕЛІ ТА ТЕХНОЛОГІЇ ПРОЕКТУВАННЯ ГОЛОВОЛОМКОВИХ КОМП'ЮТЕРНИХ ІГОР ІЗ ВИКОРИСТАННЯМ ГЕЙМДИЗАЙН-ОРІЄНТОВАНИХ ПІДХОДІВ

Сучасний етап розвитку цифрових технологій характеризується постійним зближенням ігрових механік і повсякденних процесів. Все частіше помітно як елементи, запозичені з індустрії розваг, проникають у навчальні платформи, корпоративні сервіси, додатки до здорового способу життя та навіть у звичайні інструменти для прийняття рішень. За масовим поширенням гейміфікації ховається серйозна проблема: багато розробників обмежуються поверхневим копіюванням зовнішніх атрибутів ігор, ігноруючи фундаментальні принципи геймдизайну. Такий механічний підхід призводить до створення ігрових обгорток, які швидко надоїдають користувачам, замість того ,щоб приносити задоволення [1].

Актуальність дослідження підтверджується кількома ключовими факторами. По-перше, ринок гейміфікованих рішень продовжує стрімко зростати, але також зростає і кількість провальних проектів , де ігрові механіки відштовхують користувачів замість того , щоб їх утримувати. По-друге, в українському сегменті розробки програмного забезпечення все частіше виникає потреба у створенні не просто функціональних , а і візуально приємних продуктів, які здатні конкурувати на глобальному ринку. Люди використовують такі застосунки не лише для розваги, вони допомагають подолати прокрастинацію, зняти відповідальність коли важко зробити вибір.

Мета дослідження полягає у розробленні комп'ютерної гри жанру головоломки у відповідності до принципів візуалізації і конструювання ігрового середовища з урахуванням геймдизайн орієнтованих підходів.

У рамках дослідження проведено опитування щодо створених в цій роботі інтерфейсів. Було взято групу людей, які тестували інтерфейс та відповідали на питання. На основі відповідей була проведена ітерація дизайну та проведено останнє опитування, ціль якого дійти до максимально комфортного для гравця інтерфейсу [2].

Дослідження стануть важливим інструментом та посібником для створення UX орієнтованих ігор. Проаналізувавши існуючі матеріали по цій темі, було виявлено, що більшість посібників мають поверхневе описання правил та методологій створення таких ігор, які захоплять гравців [3]. На зараз, дуже мало матеріалів описують як проводити UX орієнтовані дослідження, як проводити опитування саме в ігровій сфері та як аналізуючи результати досліджень та опитувань створити гру, яка буде створена на основі потреб гравців і правил геймдизайну [4].

У рамках дослідження ґрунтовно прописано правила геймдизайну, розглянуто нові методи створення ігрового досвіду на прикладі сучасних ігор [5]. Було аналізовано як створюється такий левел дизайн, який не дає гравцю засумувати, але і не проводить його за руку через рівні, давая йому свободу вибору. Також, було розглянуто процедуру створення дизайну ігрового інтерфейсу на основі проведення тривалих експериментів (рис. 1).

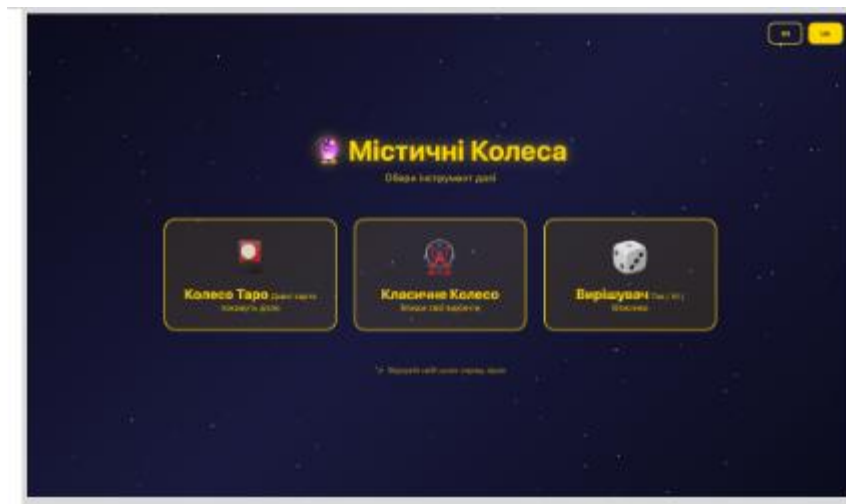


Рис. 1. Прототип головної екранної форми
Джерело: опрацювання власне

Розроблено методологію створення гри за допомогою системи нодів, де детально описується створення варіативних моделей в самому двигуні і методологія створення дизайну в ігровому середовищі, проведено створення ігрового інтерфейсу для обох ігор за правилами UI/UX [6].

Генерація випадкового індексу здійснюється за допомогою вбудованого PRNG мови JavaScript, що забезпечує рівномірний розподіл ймовірностей для кожного варіанту [7]. Функція `spinLotteryDrum()`:

```
function spinLotteryDrum() {
  // 1. Перевірка стану обертання
  if (isLotterySpinning) return;

  // 2. Фільтрація активних варіантів
  const enabledOptions = lotteryOptions.filter(opt => opt.enabled);
  if (enabledOptions.length < 2) { /* помилка */ return; }

  // 3. Запуск анімації
  drumSpinner.classList.add('spinning');

  // 4. Випадковий вибір переможця
  const winner = enabledOptions[Math.floor(Math.random() *
  enabledOptions.length)];

  // 5. Після зупинки: показ результату + відключення переможця
  setTimeout(() => {
    showWinnerBall(winner);
    lotteryOptions[realIndex].enabled = false; // РЕЖИМ НА ВИБУВАННЯ
    localStorage.setItem('lotteryOptions', JSON.stringify(lotteryOptions));
    renderLotteryDrum(); // Оновлення візуалізації
  }, 4000);
}
```

У рамках дослідження пропонуються результати опитування користувачів у контексті створених ігрових інтерфейсів пропонованих ігор. Результати опитування збираються та аналізуються, а на їх основі проводиться етап ітерації – зміни в дизайні базуючись на відповідях та потребах користувачів. Після закінчення ітерації проводиться повторне опитування з новим дизайном і аналізуються результати до і після ітерації.

Основні результати роботи: 1. Здійснено ґрунтовний аналіз правил використання гейм дизайну в процесі створення ігор. 2. Розглянуто інструментарій ігрових двигунів. 3. Розроблено методичні аспекти проектування і створення комп'ютерних ігор. 4. Здійснено опитування користувачів щодо оптимізації створених ігор з урахуванням геймдизайн орієнтованого підходу. 5. Проведено ітерації в геймдизайні з урахуванням відповідей користувачів. Результати опитування стануть у нагоді при створенні ігрового інтерфейсу для комп'ютерних ігор, адже мають верифіковані дані ігрового досвіду користувачів.

***Анотація.** Дотепер елементи, запозичені з індустрії розваг, проникають у навчальні платформи, корпоративні сервіси, додатки до здорового способу життя та навіть у звичайні інструменти для прийняття рішень. За масовим поширенням гейміфікації ховається серйозна проблема: багато розробників обмежуються поверхневим копіюванням зовнішніх атрибутів ігор, ігноруючи фундаментальні принципи геймдизайну. Такий механічний підхід призводить до створення ігрових обгорт, які швидко надходять користувачам, замість того, щоб приносити ігровий ефект. Мета дослідження полягає у розробленні комп'ютерної гри жанру головоломки у відповідності до принципів візуалізації і конструювання ігрового середовища з урахуванням геймдизайн орієнтованих підходів. У рамках дослідження пропонуються результати опитування користувачів у контексті створених ігрових інтерфейсів пропонованих ігор. Результати опитування збираються та аналізуються, а на їх основі проводиться етап ітерації – зміни в дизайні базуючись на потребах користувачів. Після закінчення ітерації проводилося повторне опитування з новим дизайном і ґрунтовно аналізувались результати тестування.*

***Ключові слова:** геймдизайн орієнтований підхід, дослідницьке навчання, варіативні моделі, комп'ютерне моделювання, КОМСДН, ігрова система, AI, математичне моделювання, ігровий інтерфейс, головоломка, UI/UX.*

Література

1. Hrybiuk O. Problems of expert evaluation in terms of the use of variative models of a computer-oriented learning environment of mathematical and natural science disciplines in schools, [w:] *Zeszyty Naukowe Politechniki Poznańskiej. Seria: Organizacja i Zarządzanie*, Zeszyt Nr 79, Poznań: Wydawnictwo Politechniki Poznańskiej (WPP), 2019. P. 101-119.
2. Hrybiuk, O., Kant, G.S. CleverCOMSRL: Implementation of an AI Computer-Aided Design System in the Context of the Cognitive Science Paradigm for the Research Training Process. In: Machado, J., et al. *Innovations in Mechatronics Engineering III. icieng 2024. Lecture Notes in Mechanical Engineering*. Springer, Cham. 2024. P. 351-362. https://doi.org/10.1007/978-3-031-61575-7_32.
3. Гриб'юк О.О. Комп'ютерне моделювання та робототехніка в навчально-виховному процесі сучасного навчального закладу. // *Сьома міжнародна науково-практична конференція FOSS Lviv-2017: Збірник наукових праць*, 27-30 квітня 2017 року, м. Львів. С. 38-43. ISBN 978-966-2598-86-5.
4. Hrybiuk Olena. Engineering in Educational Institutions: Standards for Arduino Robots as an Opportunity to Occupy an Important Niche in Educational Robotics in the Context of Manufacturing 4.0, in: *Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*. Volume 27-32, 2020. P. 770-785.
5. Hrybiuk, O. Improvement of the educational process by the creation of centers for intellectual development and scientific and technical creativity. In: Hamrol, A., Kujawińska, A., Barraza, M.F.S. (eds.) *MANUFACTURING 2019. LNME*, 2019. P. 370–382. Springer, Cham. https://doi.org/10.1007/978-3-030-18789-7_31
6. Hrybiuk, O.O. Research learning of the natural science and mathematics cycle using computer-oriented methodological systems. Monograph. Drahomanov NPU, Kyiv, 2019. P. 277–325.
7. Khang A., Abdullayev V., Hrybiuk O., & Shukla A.K. (Eds.). *Computer Vision and AI-Integrated IoT Technologies in the Medical Ecosystem* (1st ed.). CRC Press. 2024. <https://doi.org/10.1201/9781003429609>.

УДК 004 + 621.91:67.05

Могильницький М.В. (методист)

Навчально-методичний центр професійно-технічної освіти у Чернігівській області (Чернігів)

ШЛЯХИ ВИКЛАДАННЯ МАТЕМАТИЧНИХ ДИСЦИПЛІН ДЛЯ ПОТРЕБ РОБІТНИКІВ-ВЕРСТАТНИКІВ

Вивчаючи науково-методичну літературу можна дійти висновку, що теорія і практика вивчення математики в навчальних закладах I-II рівнів акредитації, в закладах додаткової підготовки та перепідготовки є досить маловивченою. Разом з тим, в суспільстві знизилась роль та чисельність промислових робітників,

натомість зросли потреби в підготовці економістів, аграріїв, фахівців інформаційно-облікової сфери, що дало певний тематичний зсув в методах і задачах математичної освіти [1-3;5].

В історичній ретроспективі не можна не згадати курс вищої математики для технікумів радянських часів І.Л. Зайцева [6], який видавався з 1950-х рр. та мав навіть міжнародні версії. Курс був полегшеною версією так званого втузівського курсу – для технічних ВНЗ, політехнікуму. Спочатку йшов виклад аналітичної геометрії на площині, окремо розглядалися криві другого порядку; далі елементи диференціального та інтегрального числення (теорія границь, функція та її властивості, похідна, формули диференціювання, дослідження функцій за допомоги похідної, застосування похідної в фізиці та техніці, диференціал, інтеграл, використання інтеграла до обчислення площин фігур та об'єму тіл, застосування інтеграла в механіці – що важливо для робітників-металообробників. Пізніші перевидання мали розділ «Доповнення»: диференціальні рівняння, ряди, гармонічний аналіз; теми стали в нагоді електрикам та радіотехнікам.

Не можна не враховувати й те, що сьогодні дещо ослабла роль певних розділів класичної вищої математики і посилилась роль інших математичних дисциплін: дискретної математики, чисельних методів, математичного програмування, теорії ймовірностей та математичної статистики, моделювання тощо. Ця проблема загальна для математичної освіти – вона далеко виходить за межі обговорення математичних програм для технікумів та коледжів. Відповіддю на ці об'єктивні труднощі впровадження математичних методів в справу підготовки студентів до сучасної виробничої діяльності стало застосування інформаційно-комп'ютерних засобів навчання (в якості методів активізації пізнавальних процесів: Mathcad, Maple V, Mathematica, Matlab, Derive, GRAN, MuPad) та організаційно-педагогічні методи (метод проектів, розрахунково-графічні роботи, виконання самостійних творчих завдань, тощо). Що ж стосується

фахівців робітничих спеціальностей, то можна зазначити наступне. Токар повинен правильно інтерпретувати креслення, щоб розуміти який вид обробки потребує та чи інша деталь. З елементарної математики особливо важлива тригонометрія. Робота фрезерувальника – це складні геометричні форми, креслення з багатьма «видами», розрізами, криволінійні контури та їх поєднання, робота пор координатам. Для успішної роботи мають бути компетенції з аналітичної та нарисної геометрії, фізики.

Узагальнюючи вимоги до математичної підготовки молодших фахівців, можна окреслити наступні моменти на яких концентрується сучасна навчальна програма. Функції та їх конкретизація: степенева, логарифмічна, показникові, тригонометричні функції. Рівняння. Похідна (дослідження функцій методами диференціального числення одної змінної). Інтеграл. Комбінаторика. Елементи теорії ймовірності та мат. статистика. Вектори та координати. Елементи стереометрії. Тіла та поверхні обертань. Отже, програма являє собою певне поєднання полегшеного курсу аналізу та математичної статистики з поглибленими розділами елементарної математики.

Шляхи «осучаснення» математичних курсів для коледжів йдуть двома шляхами – через розмиття та зближення навчального контенту для вищої та середньої спеціальної школи та систематичного розширення існуючої програми для потреб тієї чи іншої спеціальності. Першим шляхом йдуть автори підручника Р. Коляди та співавт. [4], які адресують свою роботу студентам навчальних закладів усіх рівнів акредитації. Посібник написаний відповідно до навчальної програми з вищої математики для підготовки фахівців інженерно-технічних та технологічних спеціальностей. Він складається з наступних розділів: елементи лінійної і векторної алгебри, елементи аналітичної геометрії, вступ до математичного аналізу, диференціальне числення функцій однієї змінної, інтегральне числення, функції багатьох змінних, диференціальні рівняння, теорія

рядів, кратні та криволінійні інтеграли. Тут викладач і студент самі оберуть потрібний контент. Другий підхід можна побачити в серії підручників М.Л. Бакланової та співавт. [1-3]. Підручник [1] охоплює такі розділи, як лінійна алгебра, векторна алгебра аналітична геометрія; плюс позапрограмні теми: комплексні числа, елементи теорії многочленів. Він складається з трьох частин: теорія + вправи, опис математичного пакету Mathcad, довідкова частина, що відсилає студента до шкільної програми та додаткової літератури. Підручник вищої математики [3] на перший погляд тематично рухається в річищі методик того ж Зайцева, але охоплює тематику значно ширше і наближається до політехнічного курсу для ВНЗ (4-8 семестри: по 130 годин лекцій та семінарів, 280 годин самостійної роботи). Достатньо згадати, що в курс входять: теореми Ферма, Ролля, Лагранжа, Коші. Правило Лопіталя. Формула Тейлора. Розклад елементарних функцій в ряд Маклорена; невластні інтеграли, диференціальне числення функції багатьох змінних, скалярне поле, градієнт, диференціальні рівняння вищих порядків тощо. На курсове проектування винесені, наприклад, алгебраїчні структури.

Висновки. Для робітничих спеціальностей важливими виявляються креслення і нарисна геометрія, компетенції в галузі тривимірних пакетів та об'ємного моделювання; навички роботи з електронним таблицями (Excel) дуже важливі: це й розрахунок параметрів деталей, складання технологічних карт і переліку матеріалів. Щодо конкретики тематичного наповнення курсів вищої математики, то скоріше тут краще слідувати в напрямку розширеної класичної традиції геометрії та аналізу, ніж уводити студента-металообробника в методики, які притаманні скоріше економетричним дисциплінам (наприклад, логіку, графі, дискретну математику скоріше слід поєднувати з теорією програмування, ніж з стандартним курсом вищої математики).

Анотація. В статті обговорюються шляхи оптимізації математичної підготовки молодших фахівців зі спеціальностей дерево- та металообробки: токарів та фрезерувальників.

Ключові слова: робітники-верстатники, навчальні заклади I-II рівнів акредитації, курс вищої математики, математична підготовка

Література

1. Бакланова М.Л. Лінійна і векторна алгебра та аналітична геометрія: методичний посібник. Черкаси: Фламінго, 2003. 208 с.
2. Бакланова М.Л., Триус Ю.В. Основи лінійного програмування з комп'ютерною підтримкою. Черкаси: Фламінго, 2006. 221 с.
3. Бакланова М.Л. Основи вищої математики. Черкаси: Фламінго, 2007. 300 с.
4. Коляда Р. В., Мельник І.О., Мельник О. М. Вища математика: Навч. посібник. Львів: Магнолія, 2025 342 с.
5. Рижій В.В. Посібник з елементів вищої математики для технікумів та коледжів . Вінниця, 2017. 146 с.
6. Záitsev I. L. Elementos de Matemáticas Superiores. Moscu-Bucarest, 1977. 422 p.

Nazar Tanasiuk (undergraduate student, 3 year),
Liubov Nemchenko (undergraduate student, 3 year),
Research Advisor - Associate Professor G. Bobrytska
Ivan Kozhedub Kharkiv National Air Force University

STABLE PAIRS IN BIPARTITE GRAPHS AND EXAMPLES OF THEIR USE IN MATHEMATICAL MODULAR ARITHMETIC

Introduction. Matching problems appear in many practical situations where two groups of agents must be paired according to their preferences. Typical examples include assigning students to universities, matching medical residents to hospitals, or pairing participants in various cooperative activities.

One of the most famous models of this type is the Stable Marriage Problem, introduced by Gale and Shapley in 1962.

The central question is whether it is always possible to construct a matching between two sets of agents in such a way that no pair would prefer to abandon their assigned partners and form a new pair. Such matchings are called stable.

Remarkably, Gale and Shapley proved that a stable matching always exists and can be found by a simple algorithm.

Matchings in Graphs. Many problems involving pairings can be naturally described using graph theory.

Let $G = (V, E)$ be an undirected graph. A matching in G is a set of edges $M \subseteq E$ such that no two edges in M share a common vertex.

In other words, every vertex is incident to at most one edge of the matching.

If every vertex of the graph is matched, the matching is called perfect.

Bipartite Graphs. A graph is called bipartite if its vertices can be divided into two disjoint sets X and Y such that every edge connects a vertex of X with a vertex of Y .

Matching problems often arise naturally in bipartite graphs. For example, one set may represent workers and the other jobs.

Theorem 1 (Hall's Marriage Theorem). *Let $G = (X, Y, E)$ be a bipartite graph. A perfect matching that covers all vertices of X exists if and only if for every subset $S \subseteq X$*

$$|N(S)| \geq |S|,$$

where $N(S)$ is the set of neighbors of vertices in S .

In other words, every group of vertices in X must have at least as many available neighbors in Y . Otherwise it would be impossible to assign distinct partners to all vertices in the group.

Hall's theorem guarantees the existence of perfect matchings in bipartite graphs under suitable conditions.

However, in many real situations participants have preferences. In this case the goal is not only to find a matching, but also to ensure that the resulting pairs are stable.

The Stable marriage Problem. Consider two sets of equal size:

$$M = \{m_1, m_2, \dots, m_n\}, \quad W = \{w_1, w_2, \dots, w_n\},$$

where M represents men and W represents women.

Each participant has a strict preference ordering over all members of the opposite set.

A matching is a bijection $\mu : M \rightarrow W$ which assigns each man to exactly one woman and vice versa.

Stable Matchings. A matching μ is called stable if there is no pair (m, w) such that m prefers w to $\mu(m)$, w prefers m to $\mu^{-1}(w)$.

Such a pair is called a blocking pair, because both participants would prefer to deviate from the assigned matching.

In other words, matching is stable if there is no connection outside of this matching that is better for both man and woman in it.

The Gale-Shapley Algorithm. The existence of stable matchings was proved by Gale and Shapley using the following algorithm.

Initially all participants are unmatched.

1. Each unmatched man proposes to the most preferred woman on his list to whom he has not yet proposed.
2. Each woman considers all proposals she receives and temporarily accepts the one she prefers most, rejecting the others.
3. Rejected men continue proposing to the next woman on their preference list.

The process continues until every participant is matched.

The Stable marriage Problem. Consider two sets of equal size:

$$M = \{m_1, m_2, \dots, m_n\}, \quad W = \{w_1, w_2, \dots, w_n\},$$

where M represents men and W represents women.

Each participant has a strict preference ordering over all members of the opposite set.

A matching is a bijection $\mu : M \rightarrow W$ which assigns each man to exactly one woman and vice versa.

Stable Matchings. A matching μ is called stable if there is no pair (m, w) such that m prefers w to $\mu(m)$, w prefers m to $\mu^{-1}(w)$.

Such a pair is called a blocking pair, because both participants would prefer to deviate from the assigned matching.

In other words, matching is stable if there is no connection outside of this matching that is better for both man and woman in it.

The Gale-Shapley Algorithm. The existence of stable matchings was proved by Gale and Shapley using the following algorithm.

Initially all participants are unmatched.

1. Each unmatched man proposes to the most preferred woman on his list to whom he has not yet proposed.
2. Each woman considers all proposals she receives and temporarily accepts the one she prefers most, rejecting the others.
3. Rejected men continue proposing to the next woman on their preference list.

The process continues until every participant is matched. **Theorem 2** (Gale–Shapley). *For every instance of the stable marriage problem, with the same number of men and women, there exists a stable matching.*

Idea of the proof. The algorithm must terminate because each man proposes to each woman at most once, and there are finitely many possible proposals.

When the algorithm stops, every participant is matched.

Suppose that a blocking pair (m, w) exists. Then m must have proposed to w before proposing to his final partner. At that time w either rejected him immediately or later replaced him with a more preferred partner. Therefore w cannot prefer m to her final partner, which contradicts the definition of a blocking pair.

The process continues until every participant is matched. **Theorem 2** (Gale–Shapley). *For every instance of the stable marriage problem, with the same number of men and women, there exists a stable matching.*

Idea of the proof. The algorithm must terminate because each man proposes to each woman at most once, and there are finitely many possible proposals.

When the algorithm stops, every participant is matched.

Suppose that a blocking pair (m, w) exists. Then m must have proposed to w before proposing to his final partner. At that time w either rejected him immediately or later replaced him with a more preferred partner.

Therefore w cannot prefer m to her final partner, which contradicts the definition of a blocking pair.

Example of the Gale-Shapley Algorithm. Consider three men m_1, m_2, m_3 and three women w_1, w_2, w_3 with the following preferences.

Men: $m_1 : w_1 \succ w_2 \succ w_3$

$m_2 : w_1 \succ w_3 \succ w_2$

$m_3 : w_2 \succ w_1 \succ w_3$

Women: $w_1 : m_2 \succ m_1 \succ m_3$

$w_2 : m_1 \succ m_3 \succ m_2$

$w_3 : m_3 \succ m_2 \succ m_1$

Execution of the Algorithm.

Step 1. m_1 proposes to w_1 , m_2 proposes to w_1 , m_3 proposes to w_2 .

Woman w_1 keeps the proposal from m_2 and rejects m_1 . Woman w_2 accepts m_3 .

Step 2. m_1 proposes to w_2 .

Woman w_2 prefers m_1 to m_3 , so she accepts m_1 and rejects m_3 .

Step 3. m_3 proposes to w_1 , but w_1 prefers m_2 and rejects him.

Step 4. m_3 proposes to w_3 , who accepts.

The final matching is $(m_1, w_2), (m_2, w_1), (m_3, w_3)$.

Optimality Property. An interesting feature of the Gale–Shapley algorithm is that it produces a matching that is optimal for the proposing side.

If men propose, the resulting matching is the best stable matching for all men and the worst stable matching for all women.

A Property of the Algorithm. An interesting property of the Gale–Shapley algorithm is that once a woman rejects a man, she will never later be matched with him.

If a woman rejects a proposal from a man during the algorithm, then they cannot be matched in the final stable matching.

Idea. When a woman rejects a man, she already holds a proposal from another man whom she prefers.

During the algorithm she can only replace her current partner with someone she prefers even more.

Therefore she will never end up matched with a man she previously rejected.

Thus every rejection in the algorithm is permanent.

Structure of Stable Matchings. Let μ and ν be two stable matchings.

We say that μ is better for men than ν if every man weakly prefers his partner in μ to his partner in ν .

Theorem 3. *The set of all stable matchings forms a lattice with respect to the preference ordering of participants.*

In particular, there exists

- a men-optimal stable matching,
- a women-optimal stable matching.

Every other stable matching lies between these two extremes.

The Gale–Shapley algorithm produces the men-optimal stable matching when men propose, and the women-optimal matching when women propose.

Thus the algorithm does not merely prove existence. It also selects a particular stable matching that is optimal for one side of the market.

The number of stable matchings may grow exponentially with the number of participants.

Applications. Stable matching theory has important real-world applications.

For example, variations of the Gale–Shapley algorithm are used in the matching of medical residents to hospitals and in school admission systems in several countries.

These applications demonstrate how abstract mathematical ideas can lead to practical solutions for complex social problems.

Conclusion. The stable marriage problem provides a remarkable example of how simple mathematical models can capture complex social interactions. Starting from the basic notion of matchings in bipartite graphs, one arrives at the problem of pairing participants who have individual preferences.

While classical results such as Hall’s theorem guarantee the existence of perfect matchings under suitable combinatorial conditions, the stable marriage problem introduces an additional requirement: the resulting matching should be stable with respect to the preferences of the participants.

The Gale–Shapley algorithm shows that such stable matchings always exist and can be constructed by a simple iterative procedure. Moreover, the set of all stable matchings possesses a rich mathematical structure, forming a lattice and containing distinguished extreme solutions that are optimal for one side of the market.

These results illustrate an important principle of modern discrete mathematics: elegant theoretical ideas can lead to efficient algorithms and practical applications in real-world matching markets such as school admissions and labor allocation.

Thus the theory of stable matchings demonstrates how mathematical reasoning can provide both conceptual insight and practical mechanisms for organizing complex systems of individual preferences.

References

1. Bondy, U.S.R. Murphy, Graph Theory/ Springer London, 2011. – 655 p.

2. A. B. Bollobas, Modern Graph Theory / Springer New York, 2011. – 394 p.
3. D. West, Introduction to Graph Theory / Pearson, 2017. – 608 p.

Огнєва Х. О (здобувач бакалаврату, 1 курс),
Науковий керівник – доц Михайленко І. В.
Харківський національний автомобільно-дорожній університет

ІНТЕГРАЛЬНИЙ ПІДХІД ДО ОЦІНЮВАННЯ СУМАРНОГО ПОПИТУ НА ТРАНСПОРТНІ ПОСЛУГИ

У сучасних умовах функціонування транспортної системи України, що характеризуються нестабільністю вантажопотоків, сезонними коливаннями та зростанням ролі електронної комерції, особливої актуальності набуває точне прогнозування попиту на перевезення. Здатність правильно оцінити сумарний обсяг транспортних послуг за певний період дозволяє підприємствам ефективно планувати ресурси: кількість транспортних засобів, персонал, паливе, оптимізувати маршрути та знижувати витрати.

Традиційно попит розглядають як дискретну величину (наприклад, кількість замовлень за годину або за день). Однак для більш точного аналізу доцільно перейти до неперервних моделей, які враховують динаміку змін протягом доби, тижня або місяця. Математичним апаратом, що дозволяє підсумовувати неперервно змінювану величину, є визначений інтеграл.

Метою даної роботи є обґрунтування застосування визначеного інтеграла для визначення сумарного попиту на транспортні послуги та ілюстрація методу на конкретному прикладі.

Нехай попит на транспортні послуги (наприклад, кількість тонн вантажу, що потребує перевезення за одиницю часу) є неперервною функцією часу $D(t)$. Тоді сумарний попит Q за проміжок часу від $t=\alpha$ до $t=\beta$ визначається за формулою:

$$\int_{\alpha}^{\beta} D(t) dt$$

Геометричний зміст цього інтеграла - це площа криволінійної трапеції, обмеженої графіком функції $D(t)$, віссю абсцис t та вертикальними прямими $t=\alpha$ і $t=\beta$.

Якщо функція попиту задана дискретно (на основі емпіричних даних), її можна апроксимувати аналітичною функцією (наприклад, методом найменших квадратів) або застосувати чисельні методи інтегрування (метод прямокутників, трапецій, Сімпсона). Це дозволяє використовувати інтегральне числення для обробки реальних статистичних даних.

Застосування інтеграла дає змогу врахувати нерівномірність попиту в часі та отримати точне значення сумарного обсягу, необхідне для планування роботи транспортного підприємства.

Наведемо практичний приклад розрахунку сумарного попиту.

Розглянемо маршрут Харків – Київ. Припустимо, що попит на перевезення вантажів (у тоннах) протягом робочого тижня (7 днів) описується квадратичною функцією:

$$D(t) = 100 + 20t - t^2,$$

де t - час у днях ($t=0$ відповідає понеділку, $t=7$ – неділі). Така модель відображає зростання попиту в середині тижня (пік припадає на четвер) і спад до вихідних, що є типовим для багатьох вантажних перевезень.

Необхідно визначити сумарний попит за тиждень.

Обчислюємо визначений інтеграл:

$$Q = \int_0^7 (100 + 20t - t^2) dt$$

Знаходимо первісну функції:

$$\int (100 + 20t - t^2)dt = 100t + 10t^2 - \frac{t^3}{3} + C$$

Застосовуємо формулу Ньютона-Лейбніца:

$$Q = \left[100t + 10t^2 - \frac{t^3}{3} \right]_0^7 = \left(100 \cdot 7 + 10 \cdot 49 - \frac{343}{3} \right) - 0$$

Обчислюємо:

$$100 \cdot 7 = 700$$

$$10 \cdot 49 = 490$$

$$\frac{343}{3} = 114\frac{1}{3} \approx 114,333$$

$$700 + 490 = 1190$$

$$1190 - 114,333 = 1075,667$$

Отже, сумарний попит за тиждень становить ****1075,7 тони**** (з округленням до десятих).

Отримане значення дозволяє спланувати необхідну кількість рейсів. Якщо вантажопідйомність одного автомобіля дорівнює 20 тонн, то за тиждень потрібно виконати:

$$N = \frac{1075,7}{20} \approx 53,8 \approx 54 \text{ рейси}$$

Наведений приклад демонструє практичну цінність використання визначеного інтеграла в транспортній логістиці. Метод дозволяє перейти від дискретних спостережень до неперервної моделі та отримати точний сумарний показник, необхідний для прийняття управлінських рішень.

У реальних умовах функція попиту може мати складніший вигляд (експоненційний, тригонометричний, кусково-заданий) або бути заданою таблично. У таких випадках застосовуються чисельні методи інтегрування, які легко реалізуються за допомогою сучасних програмних засобів (Mathcad,

MATLAB, Python). Це відкриває широкі можливості для обробки великих масивів даних і підвищення точності прогнозів.

Подальші дослідження можуть бути спрямовані на врахування не лише часової, а й просторової залежності попиту (наприклад, уздовж маршруту), що потребуватиме застосування кратних інтегралів. Це дозволить будувати більш складні та адекватні моделі транспортних процесів.

Висновки

1. Визначений інтеграл є ефективним математичним інструментом для визначення сумарного попиту на транспортні послуги за умови, що попит описано неперервною функцією часу.

2. Запропонований підхід дозволяє отримати точні значення сумарного обсягу перевезень, необхідні для планування ресурсів підприємства (кількості транспортних засобів, рейсів, персоналу).

3. Наведений приклад для маршруту Харків – Київ підтверджує практичну значущість методу.

4. Інтегральне числення є основою для подальших досліджень у галузі моделювання транспортних процесів, зокрема для задач оптимізації та прогнозування.

***Анотація.** У статті досліджено можливість застосування визначеного інтеграла для обчислення сумарного попиту на транспортні послуги. Розглянуто теоретичні основи методу, побудовано модель неперервної функції попиту, залежної від часу. Наведено чисельний приклад розрахунку сумарного попиту на маршруті Харків – Київ. Проаналізовано результати та визначено перспективи використання інтегрального числення в задачах планування перевезень. Роботу виконано в межах наукової діяльності кафедри транспортних технологій ХНАДУ.*

***Ключові слова:** визначений інтеграл, попит на транспортні послуги, математичне моделювання, логістика, планування перевезень.*

Література

1. Дубовик В. П., Юрик І. І. Вища математика : навч. посіб. Київ : Вища школа, 2018. 648 с.
2. Козик В. В., Федорук О. В. Математичне моделювання в логістиці : [монографія/навч. посіб.]. Львів : ЛНУ ім. І. Франка, 2020. 312 с.
3. Костенко О. М. Застосування інтегрального числення в економіці. *Вісник економіки транспорту і промисловості*. 2021. № 74. С. 45–52.
4. Вдовиченко В. О., Кузьмін А. А. Моделювання попиту на вантажні перевезення в регіоні. *Автомобільний транспорт*. 2023. Вип. 52. С. 112–119.
5. Бугір М. К. Математика для економістів : підручник. Тернопіль : Підручники і посібники, 2019. 520 с.

Скорик А.В. (здобувач бакалаврату, 2 курс),
Науковий керівник – проф. Ємельянова Т. В.
Харківський національний автомобільно-дорожній університет

ХАРАКТЕРИСТИКИ ТА ЗАКОН РОЗПОДІЛУ ВИПАДКОВОЇ ФУНКЦІЇ, ВИПАДКОВА ВЕЛИЧИНА ЯКОЇ ПІДПОРЯДКОВАНА РОЗПОДІЛУ РЕЛЄЯ

Відомо, що в більшості задач технологічної галузі слід визначати фактори, які мають випадковий характер. Випадкові фактори присутні в прикладних задачах як випадкові функції. Одже, при моделюванні будь-якого явища треба враховувати вплив випадкового фактору. Цей вплив описується випадковими функціями, аргументами яких є змінна t та випадкова величина X , яка незалежить від цієї змінної. Випадкова величина X визначає збудження випадкового шуму, що виникає в різних фізичних та інженерних задачах, наприклад, в радіолокаційних приладах, при обробки сигналів тощо.

Наведемо приклад випадкової функції, випадкова величина якої розподілена за законом Релєя.

Приклад 1. Випадкова функція має вигляд $X(t)=V^2 t$, ($t>0$), де V – випадкова величина, розподілена за законом Релєя

$$f_V(v) = \begin{cases} \frac{v}{\sigma^2} e^{-\frac{v^2}{2\sigma^2}}, & v > 0, \\ 0, & v < 0. \end{cases}$$

Реалізації функції $X(t)=V^2 t$, ($t>0$), представлені на рис.1.

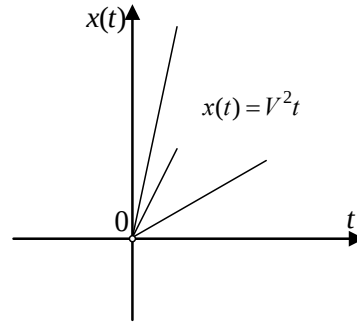


Рис. 1. Реалізації функції $X(t)=V^2 t$, ($t>0$)

Математичне сподівання випадкового процесу обчислюємо як добуток математичного сподівання випадкової величини

$$M[X(t)] = M(V^2 t) = M(V^2) t,$$

$$\begin{aligned} M[X(t)] &= M(V^2) t = t \int_0^{\infty} v^2 \frac{v}{\sigma^2} e^{-\frac{v^2}{2\sigma^2}} dv = \left\{ \begin{array}{l} v^2 = z \\ 2v dv = dz \end{array} \right\} = \frac{1}{2} t \int_0^{\infty} \frac{z}{\sigma^2} e^{-\frac{z}{2\sigma^2}} dz = \\ &= \frac{4\sigma^2 t}{2} = 2\sigma^2 t. \end{aligned}$$

Дисперсія випадкового процесу $X(t)=V^2 t$ обчислюємо, за властивістю дисперсії, як добуток дисперсії випадкової величини V^2 та множника t^2 :

$$D(X(t)) = D(V^2 t) = D(V^2) t^2,$$

$$D(V^2) = M(V^2)^2 - (M V^2)^2,$$

$$M(V^4) = \int_0^{\infty} v^4 \frac{v}{\sigma^2} e^{-\frac{v^2}{2\sigma^2}} dv = \left\{ \begin{array}{l} v^2 = z \\ 2v dv = dz \end{array} \right\} = \frac{1}{2} \int_0^{\infty} \frac{z^2}{\sigma^2} e^{-\frac{z}{2\sigma^2}} dz = 4\sigma^4.$$

$$D(V^2) = M(V^4) - (MV^2)^2 = 8\sigma^4 - (2\sigma^2)^2 = 4\sigma^4.$$

Тому, дисперсія випадкового процесу $X(t) = V^2 t$, ($t > 0$), дорівнює

$$D(X(t)) = D(V^2 t) = D(V^2) t^2 = 4\sigma^4 t^2.$$

Кореляційна функція $K_X(t_1, t_2)$

$$\begin{aligned} K_X(t_1, t_2) &= M(X(t_1)X(t_2)) - m_X(t_1)m_X(t_2) = \\ &= M(V^2 t_1 V^2 t_2) - 2\sigma^2 t_1 2\sigma^2 t_2 = M(V^4) t_1 t_2 - 4\sigma^4 t_1 t_2 = \\ &= 8\sigma^4 t_1 t_2 - 4\sigma^4 t_1 t_2 = 4\sigma^4 t_1 t_2. \end{aligned}$$

Закон розподілу випадкової функції $X(t) = V^2 t$, ($t > 0$), визначаємо за формулою, в якій випадкова величина монотонно зростає неперервно диференційована

$$f_X(x, t) = f_V(x^{-1}(x, t)) \left| \frac{d(x^{-1})}{dx} \right|,$$

де функція $x^{-1}(x, t)$ - зворотна функція до $X(t) = V^2 t$, ($t > 0$),

Зворотна функція до випадкової функції з рівності $\equiv$

$$X(t) = V^2 t \rightarrow v^2 = \frac{x}{t} \rightarrow v = \sqrt{\frac{x}{t}} = x^{-1}(x, t).$$

Похідна зворотної функції дорівнює

$$\left| \frac{d(v)}{dx} \right| = \frac{d\sqrt{\frac{x}{t}}}{dx} = \frac{1}{2\sqrt{tx}}, t > 0, x > 0.$$

Функція $f_V(x^{-1}(x, t))$ приймає вигляд

$$f_V(x^{-1}(x, t)) = \frac{\sqrt{\frac{x}{t}}}{\sigma^2} e^{-\frac{\frac{x}{t}}{2\sigma^2}}, \frac{x}{t} > 0,$$

Після множення на похідну отримаємо вираз

$$f_V(x^{-1}(x, t)) \left| \frac{d(v)}{dx} \right| = \frac{1}{2\sqrt{tx}} \frac{\sqrt{\frac{x}{t}}}{\sigma^2} e^{-\frac{\frac{x}{t}}{2\sigma^2}},$$

Тому одновимірна щільність розподілу випадкового процесу, в якому випадкова величина розподілена за законом Релея, має вигляд

$$f_X(x, t) = \begin{cases} \frac{1}{2t\sigma^2} e^{-\frac{x}{2t\sigma^2}}, & x > 0, \\ 0, & x < 0. \end{cases}$$

Відмітимо, що закон розподілу випадкової функції є показниковим законом з параметром

$$\lambda = \frac{1}{2t\sigma^2}$$

Анотація. При побудові математичної моделі будь якого явища слід урахувати можливі впливи випадкових факторів, що призводять до випадкових процесів у досліджуємой системі. В ряді задач випадкові процеси можна виразити за допомогою елементарних випадкових функцій, характеристики яких достатньо просто обчислити. В роботі наведено приклади елементарних випадкових функцій, в яких залежність від аргументу t є не випадковою функцією, випадкові величини входять у вигляді параметрів, що є незалежними від аргументу t .

Ключові слова: закон розподілу Релея, закон розподілу випадкової функції від випадкової величини за законом Релея.

Література

1. Несвіт М.І., Поклонський Є.В.. Випадкові процеси: навчально-методичний посібник. – Х. : ХНУБА, 2013 – 51 с. . Теорія випадкових процесів:
2. Задачі для самостійної роботи [Електронний ресурс] : навч. посіб. для студ. спеціальності 171 «Електроніка»/ КПІ ім. Ігоря Сікорського ; уклад.: О.В. Гармаш. – Електронні текстові дані (1 файл: 1,771 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 44 с.

Ференс Д.Ю. (магістрант, 1 курс),
Науковий керівник – доц. Рудик О. Ю.
Хмельницький національний університет, Україна

ЗАСТОСУВАННЯ SOLIDWORKS У ВИВЧЕННІ ПРИРОДНИЧО-МАТЕМАТИЧНИХ ДИСЦИПЛІН

Одним з ключових труднощів у процесі опанування природничо-математичних дисциплін виступає недостатній рівень забезпечення навчального процесу якісними навчально-методичними матеріалами. У багатьох випадках керівництво освітніх установ робить акцент переважно на розвитку матеріально-технічної складової інформатизації, розглядаючи її як пріоритетний напрям. Водночас, питання придбання сучасного програмного забезпечення або організації його власної розробки часто залишається поза належною увагою.

Грунтовне осмислення та всебічний аналіз окреслених проблем створюють передумови для визначення ефективних підходів до інтеграції інформаційних

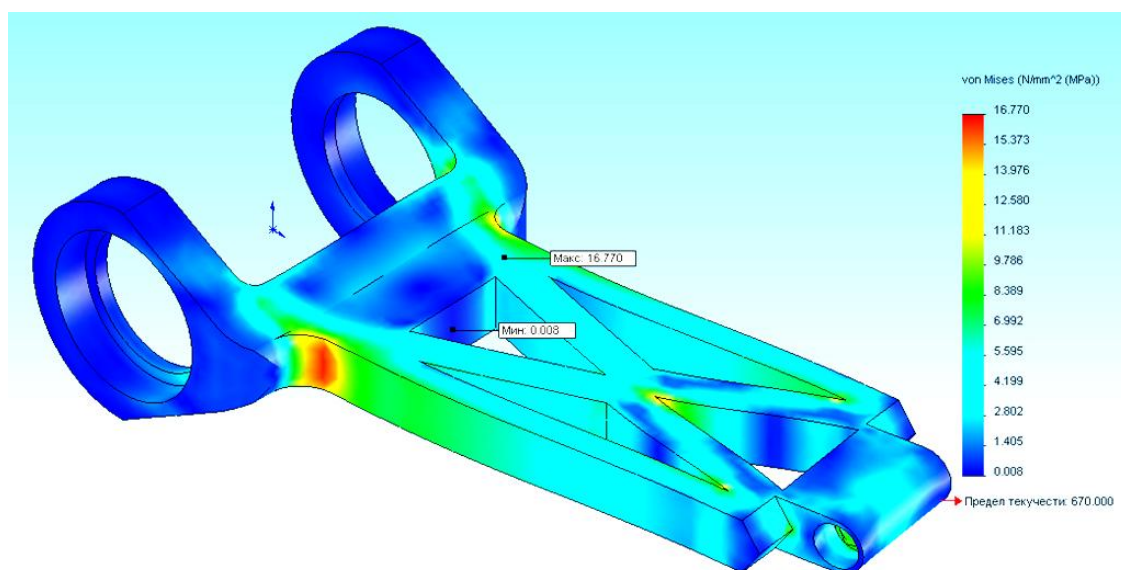
технологій (ІТ) у навчальний процес. При формуванні таких підходів доцільно зосередити увагу на низці принципових аспектів, які безпосередньо впливають на результативність їх використання.

Освітній процес із вивчення природничо-математичних дисциплін на факультеті інженерії, транспорту та архітектури ХНУ ґрунтується на активному застосуванні сучасних цифрових інструментів, зокрема 3D-системи твердотілого параметричного моделювання SOLIDWORKS [1, 2, 3]. У межах навчання значна увага приділяється дослідженню фізичних процесів, що описують напружено-деформований стан елементів автомобільної техніки. Для цього використовується спеціалізований модуль SOLIDWORKS Simulation [4, 5], який забезпечує проведення інженерного аналізу та моделювання реальних умов навантаження.

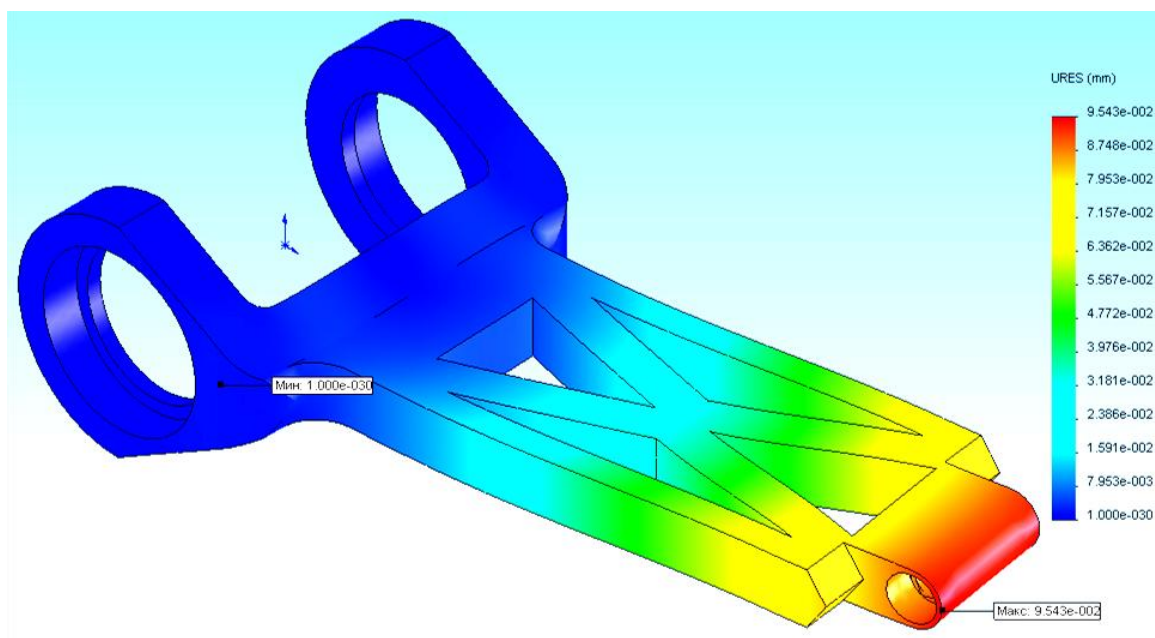
Зазначений програмний продукт функціонує на основі використання геометричної моделі деталі, на базі якої формується розрахункова модель для подальшого аналізу. Такий підхід дозволяє значно підвищити точність досліджень і забезпечує наочність отриманих результатів. Важливою перевагою є інтегрованість середовища, що дає змогу суттєво скоротити кількість операцій, пов'язаних із врахуванням специфіки побудови скінченно-елементних моделей.

Як приклад можна розглянути результати розрахунку рами поворотного кронштейна стенда, призначеного для ремонту коробок передач вантажних автомобілів (рис. 1). У представлених даних відображено розподіл еквівалентних напружень за критерієм Мізеса (von Mises), а також сумарні переміщення URES. Аналіз отриманих результатів показує, що в ділянках, позначених червоним і жовтим кольорами, концентруються максимальні значення напружень і переміщень. Це свідчить про потенційно небезпечні зони конструкції, які

потребують додаткової уваги при проєктуванні, оптимізації або підсиленні для забезпечення надійності та довговічності виробу.



а - вузлові напруження Von Mises



б – переміщення URES

Рис. 1. Результати розрахунків

Практичний досвід підтверджує, що інтеграція SOLIDWORKS у навчальний процес позитивно впливає не лише на якість засвоєння навчального матеріалу, але й на розвиток творчого мислення студентів. Застосування сучасних інженерних програмних засобів сприяє більш глибокому та системному опануванню знань, формуванню практичних умінь і навичок, необхідних для майбутньої професійної діяльності. Окрім цього, такий підхід допомагає розвивати важливі особистісні якості, зокрема аналітичне мислення, самостійність у прийнятті рішень і здатність до професійного зростання в умовах сучасного технологічного середовища.

Анотація. На прикладі CAD-системи SOLIDWORKS та її додатку – CAE-системи SOLIDWORKS Simulation застосовано математичний засіб комп'ютерного моделювання (метод скінченних елементів) для розрахунку рами поворотного кронштейна стенда, призначеного для ремонту коробок передач вантажних автомобілів

Ключові слова: SOLIDWORKS, SOLIDWORKS Simulation, метод скінченних елементів, рама поворотного кронштейна стенда.

Література

1. Rudyk O. Yu. Innovative approaches in teaching automotive disciplines / O. Yu. Rudyk, O. V. Dykha, K. E. Golenko // "System Technologies" 2 (151) 2024 «System technologies». – Pp. 144-154. – URL: <https://journals.nmetau.edu.ua/index.php/st/article/view/1717/1017>
2. Rudyk O. Yu. Modern educational resources on the example of SolidWorks / O. Yu. Rudyk, O. V. Hlyadyk, R. V. Kolisnyk, Yu. Yu. Yastremskyi // Ресурсно-орієнтоване навчання в «3D»: доступність, діалог, динаміка : збірник тез доповідей IV Міжнародної науково-практичної інтернет-конференції (м. Полтава, 22–23 лютого 2024 року). – Полтава: ФКУЕП ПДАУ, 2024. – С. 163-168. – URL: <https://elar.khmnu.edu.ua/handle/123456789/16962>
3. Rudyk O. Yu. Using SolidWorks in modern education and science [Electronic resource] / O. Yu. Rudyk, I. I. Baranov, M. M. Gereta, V. O. Dytyniuk, S. I. Fedoryshyn. – URL: <https://elar.khmnu.edu.ua/handle/123456789/14676>
4. Rudyk O. Yu. Application of SolidWorks is for professional preparation of specialists / O. Yu. Rudyk, P. V. Kaplun, V. A. Gonchar // II Міжнародна науково-практична інтернет-конференція «Ресурсно-орієнтоване навчання в «3D»: доступність, діалог, динаміка»: збірник тез доповідей (електронне видання) (м. Полтава, 22–23 лютого 2022 року). – Полтава: ПУЕТ, 2022. – С. 140-146. – Режим доступу: <http://elar.khmnu.edu.ua/jspui/handle/123456789/12794>
5. Rudyk O. Application of SolidWorks for the training of highly qualified specialists [Електронний ресурс] / O. Rudyk, P. Kaplun, V. Gonchar. – URL: <http://elar.khmnu.edu.ua/jspui/handle/123456789/12817>

ФУНДАМЕНТАЛЬНІ ОСНОВИ РОЗВ'ЯЗАННЯ ПРОФЕСІЙНО-ПРИКЛАДНИХ ЗАДАЧ

Богач І.А (здобувач бакалаврату, 2 курс),
Разумовський В.Е. (здобувач бакалаврату, 2 курс),
Науковий керівник – доц Гадецька С. В..

Харківський національний автомобільно-дорожній університет

АНАЛІЗ ТЕОРЕТИЧНИХ ОСНОВ МОДЕЛЮВАННЯ ТРАНСПОРТНИХ ПРОЦЕСІВ ЗАСОБАМИ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ

Сучасна транспортна система міста або регіону є складною динамічною системою, у якій взаємодіють транспортні засоби, дорожня інфраструктура, регулюючі пристрої, водії та зовнішні фактори. Через нерівномірність попиту на перевезення, обмеженість пропускної здатності доріг та часові коливання інтенсивності руху транспортні процеси мають яскраво виражений нестационарний характер.

У таких умовах інтуїтивних оцінок уже недостатньо. Для прогнозування завантаження вулично-дорожньої мережі, оцінювання стійкості режимів руху та пошуку ефективних рішень з організації перевезень застосовують математичне моделювання. Воно дозволяє не тільки описати спостережувані процеси, а й дослідити наслідки зміни параметрів системи: пропускної здатності, режимів світлофорного регулювання, інтенсивності прибуття транспорту, структури маршрутів.

Одним із найбільш зручних інструментів для опису динаміки транспортних систем є системи диференціальних рівнянь. Вони дають можливість представити зміну кількості автомобілів, щільності потоку, інтенсивності або черги в часі, а

також встановити умови стабілізації чи, навпаки, накопичення перевантаження. Саме тому їх використання є актуальним як для навчальних досліджень, так і для практичних задач транспортної інженерії.

Мета роботи полягає у висвітленні теоретичних основ математичного моделювання транспортних процесів за допомогою систем диференціальних рівнянь. Під транспортним потоком зазвичай розуміють сукупність транспортних засобів, що рухаються по певній ділянці мережі та характеризуються швидкістю, інтенсивністю, щільністю і часовими затримками. У макроскопічному підході потік розглядають як безперервне середовище, подібне до потоку рідини. У мікроскопічному підході аналізують поведінку окремого автомобіля та взаємодію між сусідніми транспортними засобами. Мезоскопічні моделі поєднують переваги обох підходів.

Для дослідження міських транспортних систем особливо важливо враховувати часову зміну параметрів. У години пік інтенсивність надходження транспорту різко зростає, а на окремих ділянках відбувається накопичення автомобілів. Якщо пропускна здатність нижча за попит, виникає черга, яка поширюється на суміжні елементи мережі. Саме динамічний характер цих процесів і зумовлює доцільність використання диференціальних рівнянь.

Система звичайних диференціальних рівнянь описує швидкість зміни стану системи. У транспортній інтерпретації станом може бути кількість автомобілів на ділянках, довжина черги перед вузлом, завантаження смуг руху або агрегована щільність потоку на сегментах мережі. Права частина рівнянь формується за принципом балансу: приплив мінус відтік.

Перевагою такого підходу є можливість поєднати фізичну інтерпретацію моделі з обчислювальною зручністю. Диференціальні моделі допускають аналітичне дослідження стаціонарних режимів, чисельне інтегрування, аналіз

стійкості та параметричні експерименти, що особливо важливо у задачах планування транспортної інфраструктури.

Розглянемо просту транспортну систему, що складається з трьох послідовних ділянок дороги. Нехай $x_1(t)$, $x_2(t)$, $x_3(t)$ — кількість транспортних засобів на першій, другій і третій ділянках відповідно. Тоді зміна завантаження описується такою системою рівнянь:

$$\frac{dx_1}{dt} = \lambda - k_1 x_1, \quad \frac{dx_2}{dt} = k_1 x_1 - k_2 x_2, \quad \frac{dx_3}{dt} = k_2 x_2 - k_3 x_3$$

Тут λ — інтенсивність надходження автомобілів на першу ділянку, а коефіцієнти k_1 , k_2 , k_3 характеризують інтенсивність переходу потоку між сусідніми елементами мережі. Фізично це означає, що чим більша кількість автомобілів на ділянці, тим вищий потенційний відтік з неї, якщо інфраструктурні умови залишаються незмінними.

У стаціонарному режимі похідні дорівнюють нулю. Це дає змогу перейти від динамічної системи до системи алгебраїчних рівнянь і визначити сталий рівень завантаження кожного сегмента. Однак транспортні процеси рідко є абсолютно стаціонарними, тому важливим є також аналіз перехідних процесів: як швидко система входить у сталий режим і які сегменти накопичують найбільше навантаження на проміжних етапах.

У класичній теорії транспортних потоків широко використовується модель Лайтхілла–Уїзема–Річардса, яка виражає закон збереження транспортних засобів на ділянці дороги:

$$\frac{\partial \rho(x, t)}{\partial t} + \frac{\partial q(x, t)}{\partial x} = 0$$

$$q = \rho v, \quad v = v_f \left(1 - \frac{\rho}{\rho_{\max}} \right)$$

У цій моделі $\rho(x,t)$ — щільність транспортного потоку, $q(x,t)$ — інтенсивність, v_f — швидкість вільного руху, а $\rho_{\max}$ — максимальна можлива щільність.

Модель LWR дозволяє пояснювати поширення хвиль ущільнення і розвантаження в потоці, що особливо корисно для дослідження виникнення заторів. Хоча LWR формально є рівнянням у частинних похідних, на практиці після просторової дискретизації дороги її можна звести до системи звичайних диференціальних рівнянь для окремих сегментів. Для вузлів, перехресть або коротких перевантажених ділянок зручно використовувати балансну модель накопичення черги:

$$\frac{dN}{dt} = q_{in}(t) - q_{out}(t)$$

Якщо інтенсивність прибуття q_{in} упродовж певного інтервалу часу перевищує інтенсивність вибуття q_{out} , то кількість автомобілів $N(t)$ на ділянці зростає. Це найпростіша модель утворення заторів. Вона зручна для оцінювання довжини черги, часу очищення вузла після завершення години пік і порівняння варіантів організації руху.

Анотація. Транспортна система є динамічною, тому її природно описувати диференціальними рівняннями, заснованими на балансовому принципі «приплив мінус відтік». В роботі проаналізовано основні класичні моделі теорії транспортних потоків.

Ключові слова: диференціальні рівняння, моделі теорії транспортних потоків, балансна модель.

Література

1. Helbing D. Traffic and related self-driven many-particle systems // Reviews of Modern Physics. 2001. Vol. 73. P. 1067–1141.
2. Homburger, Kell and Perkins, Fundamentals of Traffic Engineering, 13th Edition, Institute of Transportation Studies, University of California (Berkeley), 1992.

Гаркуша Я.О. (здобувач бакалаврату, 2 курс),

Богдан А.О. (здобувач бакалаврату, 2 курс),

Науковий керівник – доц Гадецька С. В.

Харківський національний автомобільно-дорожній університет

ВИКОРИСТАННЯ МЕТОДІВ ЛІНІЙНОЇ АЛГЕБРИ ДЛЯ АНАЛІЗУ СТРУКТУРИ ТРАНСПОРТНИХ МЕРЕЖ

Відомо, що матричне числення та системи лінійних рівнянь дозволяють описувати складні транспортні вузли, спираючись виключно на алгебраїчний апарат.

Транспортна мережа — це не просто набір доріг, це динамічна система розподілу потоків. Традиційно її розглядають через теорію графів, проте саме лінійна алгебра надає обчислювальний інструментарій для розв'язання задач високої розмірності. Основна ідея полягає в тому, що кожен транспортний вузол (перехрестя) можна розглядати як точку балансу. Згідно з законом збереження потоку, кількість транспорту, що входить у вузол, має дорівнювати кількості транспорту, що виходить з нього (з урахуванням накопичення або витоку).

Якщо ми позначимо інтенсивність потоку на ділянках мережі як змінні $x_1, x_2, \dots, x_n$, то структуру мережі можна описати системою лінійних алгебраїчних рівнянь (СЛАР):

$$Ax=b,$$

де A — матриця структури (коефіцієнти розподілу потоків), x - вектор інтенсивностей на ділянках, b — вектор зовнішніх входів/виходів (наприклад, потік із житлових масивів).

У класичній лінійній алгебрі для опису зв'язків використовуються матриці суміжності. Проте для аналізу транспортної структури це поняття розширюється до матриць перехідних імовірностей. Якщо ми знаємо ймовірність того, що водій

на перехресті обере певний напрямок, ми можемо скласти таку відповідну матрицю P , що стан системи через k кроків (наприклад, через 15-хвилинні інтервали) визначається як:

$$X^{(k)} = P^k X^{(0)}.$$

Це дозволяє прогнозувати "затори" та критичні точки мережі через аналіз власних чисел матриці. Якщо найбільше власне число $\lambda \approx 1$, система перебуває у стабільному (хоча, можливо, і перевантаженому) стані.

Розглянемо фрагмент міської мережі з чотирьох ключових точок (магістралей). Нехай x_1, x_2, x_3, x_4 — інтенсивності руху (авто/год) на відповідних ділянках. Наведемо умови балансу у вузлах: 1) вузол А: $500 + x_4 = x_1 + x_2$ (вхід 500 та потік з 4-ї ділянки дорівнює виходу на 1 та 2);

2) вузол В: $x_1 + 300 = x_3 + 200$ (вхід з 1-ї та зовнішній 300 дорівнює виходу на 3-ю та 200); 3) Вузол С: $x_2 + x_3 = x_4 + 600$.

Формуємо СЛАР:

$$\begin{cases} x_1 + x_2 - x_4 = 500 \\ x_1 - x_3 = -100 \\ x_2 + x_3 - x_4 = 600 \end{cases}$$

що в матричному представленні має вигляд:

$$\begin{pmatrix} 1 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 500 \\ -100 \\ 600 \end{pmatrix}$$

Розв'язавши систему методом Гаусса, маємо наступнимий результат. Нехай $x_4 = c$ (вільний потік на об'їзній). Тоді $x_3 = x_2 - 600 + c$, $x_1 = x_3 - 100 = x_2 - 700 + c$. Проаналізуємо отриманий результат: якщо ми знаємо, що мінімальна пропускна здатність x_4 , має бути 200 км/год, ми можемо точно розрахувати необхідні

параметри для інших ділянок, щоб уникнути колапсу. Це демонструє, як лінійна алгебра визначає реальні можливості транспортної системи.

Наведемо ще один приклад застосування теорії матриць до аналізу ситуації на дорогах. Припустимо, у нас є вузол, де сходяться три дороги. Датчики показали: вхід $x_1 = 1005$ авто/год, вхід $x_2 = 490$ авто/год, вихід $x_3 = 1510$ авто/год. Однак у реальному житті датчики часто помиляються. І, дійсно, є проблема у розрахунках: $1005 + 490 = 1495 \neq 1510$. Маємо похибку в 15 одиниць. Як знайти найбільш імовірні "істинні" значення? Для побудови математичної моделі мінімізуємо функцію помилки:

$$S = (x_1 - 1005)^2 + (x_2 - 490)^2 + (x_3 - 1510)^2$$

за умови лінійного обмеження $x_1 + x_2 - x_3 = 0$. При цьому вектор спостережень: $y = [1005, 490, 1510]^T$, матриця обмежень $C = [1, 1, -1]$. Використовуючи метод множників Лагранжа, отримуємо скориговані значення: $x_1^* = 1000$, $x_2^* = 485$, $x_3^* = 1515$ (зауважимо, що розрахунок залежить від напрямку вектора помилки, але основна ідея полягає в тому, що лінійна алгебра "підтягує" дані до фізичного закону збереження).

***Анотація.** Проведений аналіз використання методів лінійної алгебри для аналізу структури транспортних мереж показав, що використання лінійної алгебри, зокрема, дозволяє формалізувати структуру, перетворивши фізичні дороги на математичні об'єкти, виявляти дефіцит пропускної здатності (аналізуючи ранг відповідних матриць і вивчаючи отримані СЛАР на сумісність), оптимізувати потоки, використовуючи метод найменших квадратів для мінімізації часу затримки в системі.*

***Ключові слова:** лінійна алгебра, структура транспортних мереж, пропускна здатність.*

Список літератури

1. Helbing D. Traffic and related self-driven many-particle systems // Reviews of Modern Physics. 2001. Vol. 73. P. 1067–1141.
2. Горбачов П. Ф., Любий Є. В. Транспортні системи. — Харків: ХНАДУ, 2014. (Аналіз формування транспортних потоків у містах).

УДК 004.942:[327

Кукульчук В. О. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О

Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-технічний університет імені академіка Юрія Бугая» (Київ)

ІНФОРМАЦІЙНА ОСВІТНЯ ІНФРАСТРУКТУРА НОВОГО ПОКОЛІННЯ: МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ І ШТУЧНИЙ ІНТЕЛЕКТ

Актуальність дослідження полягає у необхідності розвитку освітніх технологій, та враховують індивідуальні потреби учнів. Інтеграція локальних П-моделей дозволяє уникнути залежності від хмарних сервісів, забезпечує конфіденційність користувачів і дозволяє системі функціонувати в умовах обмеженого інтернету [1]. Вивчення іноземних мов залишається однією з найбільш затребуваних навичок у всьому світі, яке відкриває двері різних можливостей, таких як працевлаштування, кар'єрного зростання, та спілкування з людьми.

Однак традиційні методи навчання часто мають обмежені можливості адаптації та персоналізації під кожну людину та її розуміння. Існуючі системи управління навчанням (LMS) мають низку недоліків: вони не враховують індивідуальні темпи та стилі навчання кожного учня, не забезпечують персоналізований зворотний зв'язок у реальному часі, не роблять правильну гейміфікацію та часто вимагають постійного доступу до інтернету та хмарних сервісів.

Учні з різними рівнями підготовки часто відчують брак індивідуалізованої допомоги, що призводить до зниження мотивації та якості засвоєння матеріалу. Проблема всіх додатків вивчення іноземних мов особливо гостра, де є залежність

від централізованих хмарних сервісів створює проблеми конфіденційності та недоступна в умовах обмеженого інтернет-з'єднання.

Мета дослідження полягає у розробленні освітньої інформаційної системи з використанням технологій штучного інтелекту.

У рамках дослідження здійснено ґрунтовний аналіз існуючих можливостей програмних систем задля визначення високорівневих потреб і розроблення зручної, ефективної та простої програмної системи. Розроблено освітню інформаційну систему з використанням локальних ШІ-моделей, яка забезпечує (рис. 1): адаптивне навчання іноземним мовам на основі персоналізованого контенту; автоматичну генерацію словникових слів та вправ на основі поточного рівня учня та його інтересів; діалог із віртуальним ШІ-помічником для практики розмовних навичок; повну приватність даних користувача через локальне розташування ШІ-моделей та локальну базу даних і можливість роботи без постійного підключення до інтернету.

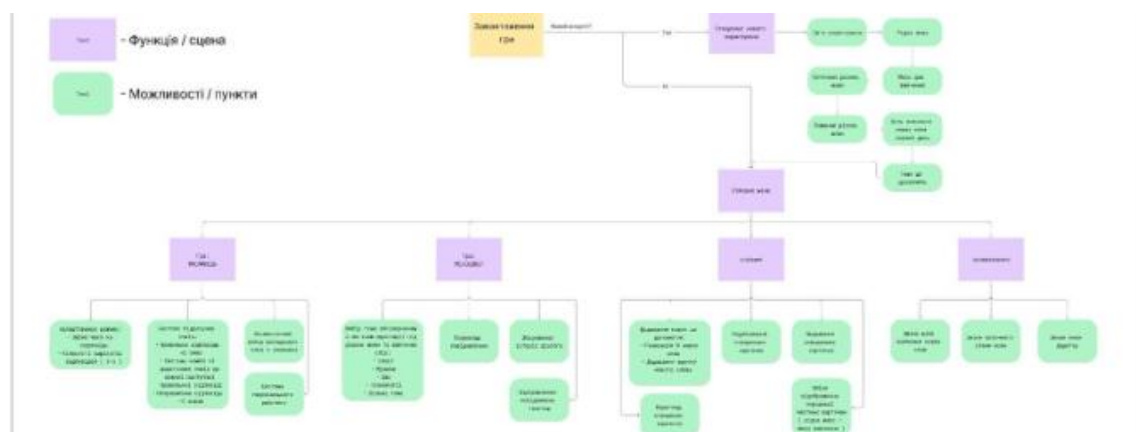


Рис. 1. Функціональна схема освітньої інформаційної системи
Джерело: опрацювання власне

Програмна система реалізована у вигляді локального клієнтського застосунку, що функціонує за рахунок ресурсів пристрою. Архітектура побудована за модульним принципом [2], що забезпечує гнучкість та

масштабованість без необхідності змін основних компонентів. До складу системи входять такі основні компоненти: користувацький інтерфейс, модуль управління профілем користувача, модуль адаптивного навчального контенту, модуль ігрової практики, модуль діалогу з ШІ-помічником, модуль управління прогресом та локальна база даних у форматі JSON. Взаємодія між компонентами здійснюється через базу даних. Локальна ШІ-модель Ollama використовується для генерації навчального контенту та забезпечення інтерактивного діалогу з користувачем. Модель працює без звернення до зовнішніх хмарних сервісів та використовує обчислювальні ресурси пристрою користувача, що забезпечує повну конфіденційність даних та незалежність від інтернет-з'єднання. Система використовує бібліотеку `neocortex-unity-sdk` для взаємодії з локальним сервісом Ollama. Бібліотека надає зручне використання відправлення запитів та обробки відповідей для локальних П-моделей. ШІ-модель обробляє текстові запити, враховує контекст попередніх повідомлень та параметри профілю користувача.

На основі цього формується індивідуалізована відповідь, яка використовується як для навчального процесу, так і для підтримки спілкування іноземною мовою з користувачем. Модуль словника відповідає за накопичення та управління вивчених слів користувача. На основі цих даних формується промпт для Ollama з вимогою генерувати N слів у JSON форматі. Структура кожного слова включає: оригінальне слово, переклад, речення з цим словом, переклад речення. Система валідує отримані дані на коректність структури та перевіряє на дублювання перед збереженням у базу даних [3]. Всі операції з файлом реалізовані як асинхронні. Після успішного збереження користувачеві виводиться сповіщення про додавання нових слів з можливістю одразу їх переглянути. Користувач може переглядати слова, обирати сортування карточок, та позначати слова як вивчені або не освоєні.

Модуль Мисливця реалізує ігровий формат закріплення слів через систему вибору слів із декількох варіантів. При запуску гри система обирає випадкові слова зі Словника користувача та підставляє їх. Система формує питання в випадковому порядку використання перекладу слова або на рідній мові, також є можливість використання речень. Після завершення сесії система розраховує метрики: кількість питань, загальне забране очків та середній час відповіді. Результати зберігаються у базу даних з детальною інформацією про сесію та метриками. Взаємодія з ШІ-помічником реалізована двома способами (рис. 2): внутрішня комунікація за допомогою промт-повідомлень, яка не видна користувачеві та використовується для автоматичних операцій, таких як генерація слів або аналіз вправ; пряме спілкування користувача з ШІ-помічником через текстовий інтерфейс, побудований за принципом месенджера. Даний підхід забезпечує інтуїтивно зрозумілий формат спілкування та створює процес навчання, наближений до реального діалогу.

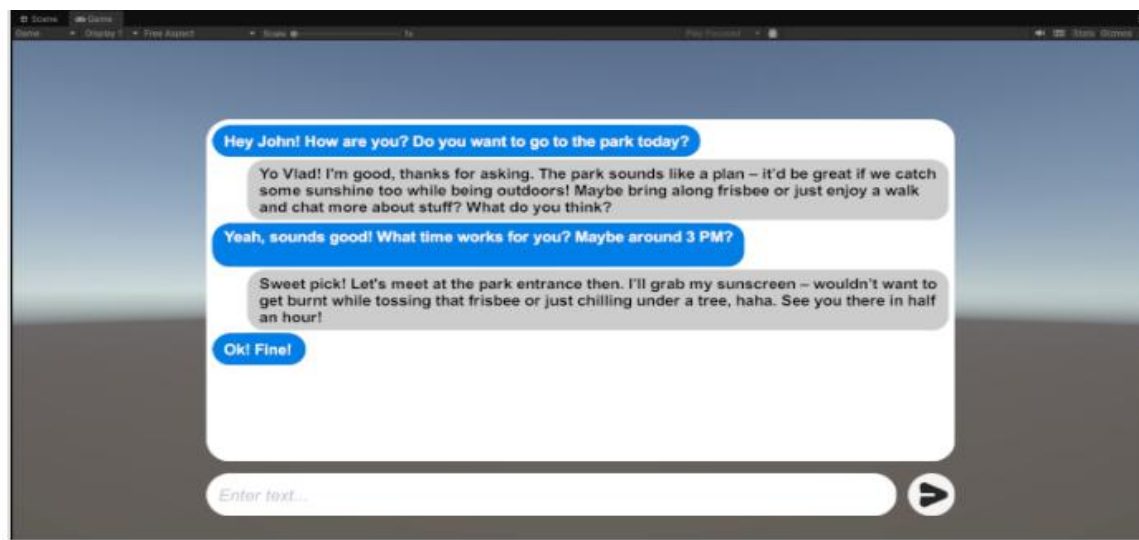


Рис. 2. Прототип екранної форми взаємодії користувача з ШІ помічником освітньої інформаційної системи
Джерело: опрацювання власне

Модуль управління профілем відповідає за збереження та обробку персональної інформації користувача. При першому запуску користувач проходить процес реєстрації, в якому вказує: ім'я, рідну мову, цільову мову, поточний рівень володіння мовою (A1-C2), бажаний рівень, список інтересів та щоденну ціль кількості нових слів. Інформація зберігається у локальну базу даних. Користувач може редагувати параметри профілю у будь-який момент через модуль налаштувань: змінити мову інтерфейсу, інтереси, щоденну ціль слів. При зміні рівня знань система автоматично при наступному використанні адаптує генерацію контенту. При зміні інтересів система додає нові слова категорії для генерації нових слів [4]. Для збереження інформації використовується локальна база даних в JSON форматі. Вся інформація зберігається з окремими файлами: users.json (профілі користувачів), dictionary.json (словник слів), messages.json (історія діалогів), game_sessions.json (результати ігор).

Анотація. У рамках дослідження запропоновано практичне вирішення фундаментальної проблеми інтеграції розробленої інформаційної системи з використанням технологій штучного інтелекту в освітній процес. Інтерфейс користувача реалізований у вигляді набору екранів з використанням Unity та C#. Особлива увага приділяється зручності навігації та адаптації інтерфейсу під різні розміри екранів. Всі елементи інтерфейсу масштабуються автоматично, що дозволяє використовувати застосунок на різних пристроях. Система використовує гнучкі макети та умовне розташування елементів залежно від ширини екрану та пропорцій. Всі шрифти, кнопки та зображення масштабуються автоматично, забезпечуючи коректне відображення на різних розмірах екранів. Розвиток технологій штучного інтелекту та великих мовних моделей (LLM) відкриває нові можливості для автоматизації та персоналізації навчального процесу. Локальні ШІ-моделі дозволяють створювати інтелектуальні системи, які працюють повністю оффлайн, забезпечують приватність користувача та можуть адаптуватися до потреб кожного учня в реальному часі.

Ключові слова: CleverCOMSRL, дослідницьке навчання, варіативні моделі, математичне моделювання, КОМСДН, інформаційна система, графічний інтерфейс, комп'ютерне моделювання, штучний інтелект, LLM, AI.

Література

1. Khang A., Abdullayev V., Hrybiuk O., & Shukla A.K. (Eds.). Computer Vision and AI-Integrated IoT Technologies in the Medical Ecosystem (1st ed.). CRC Press. 2024. <https://doi.org/10.1201/9781003429609>.
2. Hrybiuk, O.O. Research learning of the natural science and mathematics cycle using computer-oriented methodological systems. Monograph. Drahomanov NPU, Kyiv, 2019. P. 179–185.
3. Hrybiuk, O., Kant, G.S. CleverCOMSRL: Implementation of an AI Computer-Aided Design System in the Context of the Cognitive Science Paradigm for the Research Training Process. In: Machado, J., et al. *Innovations in Mechatronics Engineering III. icieng 2024. Lecture Notes in Mechanical Engineering*. Springer, Cham. 2024. P. 351-362. https://doi.org/10.1007/978-3-031-61575-7_32.
4. Hrybiuk, O. Improvement of the educational process by the creation of centers for intellectual development and scientific and technical creativity. In: Hamrol, A., Kujawińska, A., Barraza, M.F.S. (eds.) *MANUFACTURING 2019*. LNME, 2019. P. 370–382. Springer, Cham. https://doi.org/10.1007/978-3-030-18789-7_31.

Локтіонова А.О (здобувач бакалаврату, 2 курс),

Троян Є.В. (здобувач бакалаврату, 2 курс),

Науковий керівник – доц Гадецька С. В..

Харківський національний автомобільно-дорожній університет

ДОСЛІДЖЕННЯ МОДЕЛІ НАКОПИЧЕННЯ ТРАНСПОРТНОЇ ЧЕРГИ З ВИКОРИСТАННЯМ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ

Розглянемо балансну модель накопичення черги, яку використовують для вузлів, перехресть або коротких перевантажених ділянок. Балансне рівняння накопичення транспорту на перевантаженій ділянці має вигляд:

$$\frac{dN}{dt} = q_{in}(t) - q_{out}(t)$$

В цьому диференціальному рівнянні q_{in} - інтенсивність прибуття упродовж певного інтервалу часу. Якщо воно перевищує інтенсивність вибуття q_{out} , то кількість автомобілів $N(t)$ на ділянці зростає. Наведена модель застосовується для оцінювання довжини черги, часу очищення вузла після завершення години пік і порівняння варіантів організації руху. Перевага цієї моделі полягає у зрозумілій

інтерпретації параметрів та можливості швидкого числового аналізу. Для більш глибокого розуміння процесу застосування наведеної моделі розглянемо кілька прикладів.

Приклад 1: стаціонарне навантаження трьох ділянок. Нехай інтенсивність надходження транспорту $\lambda = 40$ авт/хв, а коефіцієнти пропускної здатності дорівнюють $k_1 = 0,4$; $k_2 = 0,3$; $k_3 = 0,2$. Знайдемо стаціонарний режим роботи системи. Поклавши похідні рівними нулю, одержимо: $40 - 0,4x_1 = 0$; $0,4x_1 - 0,3x_2 = 0$; $0,3x_2 - 0,2x_3 = 0$. Із першого рівняння маємо $x_1 = 100$. Підставляючи це значення у друге, дістаємо $x_2 = 133,3$. Далі з третього рівняння отримуємо $x_3 = 200$. Отримані результати означають, що за однакового вхідного потоку навантаження на наступних ділянках зростає через нижчі коефіцієнти відтоку. Найбільш напруженим елементом є третя ділянка. Саме такі сегменти доцільно розглядати як кандидати для пріоритетної оптимізації: розширення, корекції режимів регулювання або перерозподілу маршрутів.

Приклад 2: транспортний вузол з двома напрямками. Нехай $x_1(t)$ — кількість автомобілів на основному напрямку, а $x_2(t)$ — на другорядному. Модель має вигляд: $dx_1/dt = 30 - 0,5x_1 + 0,2x_2$; $dx_2/dt = 20 + 0,5x_1 - 0,4x_2$. У стаціонарному режимі отримуємо систему $30 - 0,5x_1 + 0,2x_2 = 0$ та $20 + 0,5x_1 - 0,4x_2 = 0$. Помножимо друге рівняння на 0,5 і додамо до першого так, щоб виключити x_1 . Після перетворень отримуємо $x_2 = 80$, а потім $x_1 = 92$.

Цей приклад демонструє, що навіть для невеликого вузла взаємний перерозподіл потоків між напрямками суттєво впливає на стаціонарні значення. Такі моделі корисні для оцінювання конфліктних точок у міських мережах.

Приклад 3 (у матричній формі). Розглянемо систему з трьох послідовно пов'язаних ділянок, записану в матричному вигляді. Нехай вектор стану $x(t) =$

$[x_1(t), x_2(t), x_3(t)]^T$. Тоді модель можна подати у формі загального матричного запису системи: $\dot{x} = Ax + b$.

Візьмемо такі параметри: $\lambda = 36$ авт/хв, $k_1 = 0,35$, $k_2 = 0,25$, $k_3 = 0,20$. Тоді матриця системи і вектор зовнішнього впливу мають вигляд:

$$A = \begin{pmatrix} -0,35 & 0 & 0 \\ 0,35 & -0,25 & 0 \\ 0 & 0,25 & -0,2 \end{pmatrix}, \quad b = \begin{pmatrix} 36 \\ 0 \\ 0 \end{pmatrix}.$$

Стаціонарний режим визначається з умови $Ax^* + b = 0$. Отже, потрібно розв'язати систему лінійних алгебраїчних рівнянь. У розгорнутому вигляді вона має вигляд: $-0,35x_1 + 36 = 0$; $0,35x_1 - 0,25x_2 = 0$; $0,25x_2 - 0,20x_3 = 0$. З першого рівняння одержуємо $x_1 = 36 / 0,35 = 102,86$ автомобіля. Із другого рівняння знаходимо $x_2 = 0,35 \cdot 102,86 / 0,25 = 144,00$ автомобіля. З третього рівняння маємо $x_3 = 0,25 \cdot 144 / 0,20 = 180,00$ автомобіля. Запишемо отриманий стаціонарний вектор у матричному вигляді: $x^* = [102,86; 144,00; 180,00]^T$. Це означає, що при сталому вхідному потоці 36 авт/хв система прагне до такого розподілу транспортних засобів по трьох ділянках. Уже з цих значень видно, що третя ділянка знову виявляється найбільш завантаженою. Тепер виконаємо перевірку балансу. Для першої ділянки відтік становить $0,35 \cdot 102,86 \approx 36$ авт/хв, тобто він врівноважує вхідний потік. Для другої ділянки відтік $0,25 \cdot 144 = 36$ авт/хв дорівнює притоку з першої. Для третьої ділянки відтік $0,20 \cdot 180 = 36$ авт/хв, що також збігається з притоком. Отже, стаціонарне рішення знайдено правильно.

Для ілюстрації нестационарного режиму розглянемо умовний приклад зміни щільності транспортного потоку на міській магістралі (Рис. 1). На початку інтервалу спостереження щільність є порівняно невисокою, далі зростає внаслідок збільшення попиту в години пік, а потім поступово зменшується завдяки розвантаженню мережі.

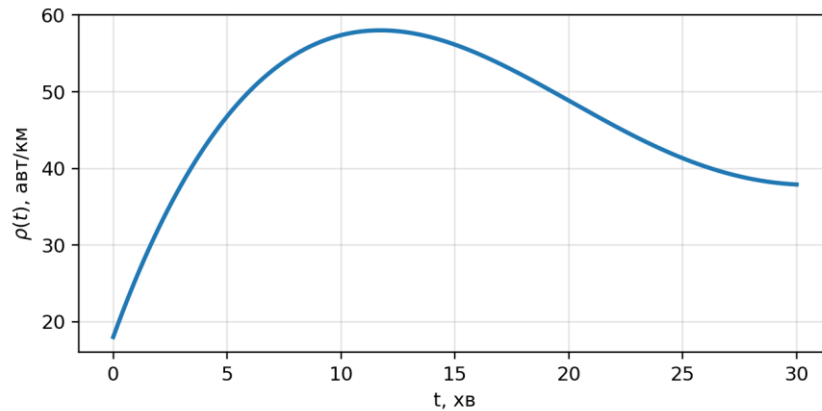


Рис. 1. Приклад графіка зміни щільності транспортного потоку

За графіком видно, що на початковому етапі система працює поблизу режиму вільного руху. Далі щільність збільшується, що призводить до зниження швидкості та підвищення ймовірності утворення локальних заторів. Після досягнення максимуму починається фаза розвантаження. Практична цінність таких графіків полягає в тому, що вони допомагають визначити критичні часові інтервали, коли необхідне активне керування: зміна циклів світлофорів, обмеження поворотів, пріоритет громадського транспорту або інформування водіїв про альтернативні маршрути.

Системи диференціальних рівнянь використовують при оцінюванні пропускної здатності доріг, моделюванні черг на підходах до перехресть, розрахунку навантаження на транспортні вузли та аналізі стабільності режимів руху. Диференціальні моделі є особливо цікавими, тому що вони поєднують класичні розділи вищої математики — диференціальні рівняння і матриці — з прикладною проблематикою транспортних систем. Водночас слід пам'ятати, що будь-яка модель є спрощенням реальності. Для підвищення адекватності необхідно враховувати нерівномірність прибуття потоку, вплив світлофорних фаз, різні типи транспортних засобів, поведінку водіїв, погодні умови.

***Анотація.** У роботі розглянуто підхід до математичного моделювання транспортних процесів за допомогою звичайних диференціальних рівнянь.*

Показано, що такий апарат дозволяє описувати динаміку завантаження окремих ділянок дороги, аналізувати стаціонарні режими й виявляти потенційні вузькі місця мережі. На прикладах продемонстровано, що аналіз стаціонарних станів і графіків зміни щільності дозволяє виявляти перевантажені ділянки та приймати інженерні рішення.

Ключові слова: звичайні диференціальні рівняння, моделювання транспортних процесів, пропускна здатність доріг.

Література

1. Каменєв Д. О. Математичні моделі аналізу транспортних мереж і характеристик руху. Харків: ХНАДУ, 2020.
2. Treiber M., Kesting A. Traffic Flow Dynamics: Data, Models and Simulation. Berlin : Springer, 2013.

УДК 004.942:[327

Лунчев Г. О. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О
Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-технічний університет імені академіка Юрія Бугая» (Київ)

ІНФОРМАЦІЙНО-МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ КАЗУАЛЬНИХ ГОЛОВОЛОМОК

В умовах постійного зростання інформаційного навантаження та дефіциту концентрації уваги, зумовленого надмірним використанням гаджетів чи нестачею сну, люди все частіше стикаються з проблемою зниження швидкості їхньої реакції, нестачею зосередженості та концентрації [1]. Серед ефективних способів подолання цієї проблеми є взаємодія з простими ігровими системами різних жанрів, що вимагають швидкого аналізу ситуації та прийняття блискавичних рішень.

Розроблення ігрової системи жанру казуальних головоломок має практичну актуальність як створення інтерактивного інструменту для розвитку когнітивних здібностей, а також сенсомоторних навичок користувача, що надзвичайно

важливо в сучасних умовах зростаючої цифрової залежності та пасивного споживання інформації.

Мета дослідження полягає у розробленні ігрової системи жанру казуальних головоломок типу «Змійка», яка демонструє принципи побудови гнучко конфігурованої та реактивної програмної системи, здатної опрацьовувати події в реальному часі з мінімальними затримками.

Архітектура програмного забезпечення розроблена з урахуванням принципів об'єктно-орієнтованого програмування (ООП) [2], використанням патернів проектування та принципів *SOLID* [3]. Відповідальність за елементи гри розподілена між різними класами (рис. 1), побудованими з урахуванням можливості легкої інтеграції нових можливостей в майбутньому, без зміни загальної логіки, що значно підвищує гнучкість ігрової програмної системи.

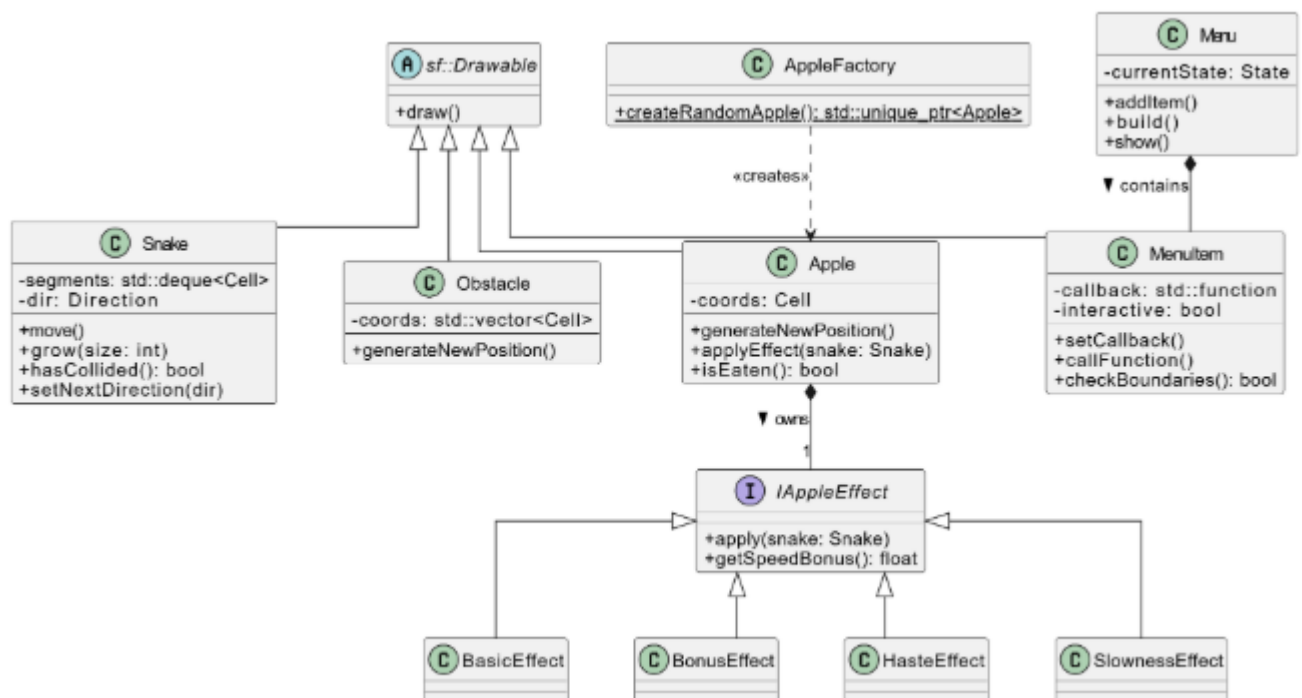


Рис. 1. Діаграма основних класів ігрової програмної системи «Змійка»
Джерело: опрацювання власне

Реалізовані компоненти розділені на дві категорії: *складові рушії*, що складають більшість реалізованих компонентів й представляють як і конкретні функціональні об'єкти, менеджери інших класів, так і структури даних, що використовуються для створення складніших сутностей; *ігрові сутності*, що представляють собою об'єкти, з якими гравець безпосередньо взаємодіє на ігровому полі. Побудова візуальних ефектів з інтеграцією *шейдерів* спростила підготовку до відмалювання сутностей та перенесла частину обчислень з центрального процесора на *графічний*, забезпечивши вищу стабільність ІПС. Візуальна архітектура системи побудована за принципом, де шейдери повністю контролюють візуальний стан об'єктів, в той час як процесор лише надсилає необхідні дані й використовує конкретний шейдер при запиті на відмалювання.

Для кожної сутності та модулів, що реалізують самостійні візуальні ефекти, розроблено *фрагментний шейдер* [4]. Клас *Snake* реалізує надсилання даних в шейдер двома шляхами: надсилання даних всередині окремої вершини, використовуючи її текстурні координати; надсилання різних даних використовуючи *shader uniforms*.

Графічний користувацький інтерфейс (GUI) ІСЖКГ «Змійка» виконаний мінімалістично, є інтуїтивним в навігації та не містить зайвих елементів чи опцій, мінімізуючи стороннє когнітивне навантаження та концентруючи увагу користувача на початку нової ігрової сесії. GUI не передбачає конкретного кольорового оформлення, надаючи користувачу вибір з кількох контрастних кольорових тем, побудованих для якомога кращого сприйняття інформації. Інтерактивне меню виступає головною точкою взаємодії з системою в GUI, через яку здійснюється конфігурація системи й ініціалізація ігрового циклу.

Фон ігрового поля стилізований у вигляді шахової дошки, візуально розділяючи клітинки *grid-based* поля, покращуючи просторову орієнтацію

користувача. Для концентрації уваги гравця на подіях в межах ігрового поля, інформація щодо рахунку гри з'являється виключно на короткий проміжок часу (< 1 секунди) поруч із ціллю в момент реєстрації зіткнення (рис. 2). Для елементів класу *Apple* та *Obstacle* передбачено ефект плавної появи, оскільки різка поява може спровокувати перемикання уваги гравця з його поточної цілі.

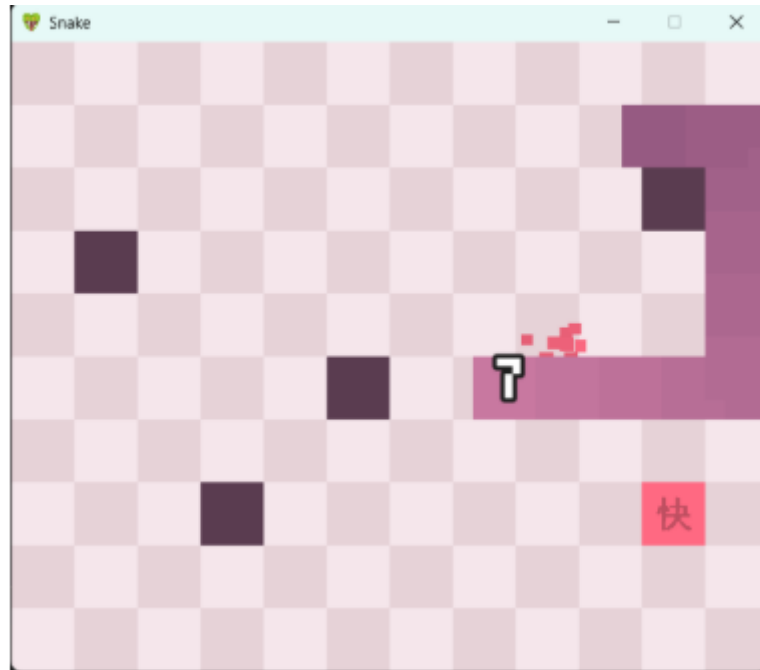


Рис. 2. Ігрове поле ігрової програмної системи «Змійка». Демонстрація появи рахунку на місці зіткнення з яблуком
Джерело: опрацювання власне

Забезпечення конфігурованості системи делеговано класу *Configuration*, що інкапсулює в собі змінювані значення. Він реалізує методи доступу до параметрів конфігурації та їхнього циклічного перемикання, абстрагуючи доступ до динамічних значень. Використання глобальних змінних або патерну «Одинак» [5] значно ускладнює тестування системи, унеможливлюючи модульне тестування системи. Архітектура системи розроблена застосовуючи принцип *впровадження залежностей* (англ. *Dependency Injection*) [6]. Життєвий цикл єдиного екземпляру *Configuration* контролюється розумним вказівником *std::unique_ptr* [7] на вищому

рівні системи, гарантуючи його безпечне знищення після завершення програми. Залежним модулям системи, та їх дочірнім об'єктам, передається *неволодіючий вказівник*, унеможливаючи випадкове знищення, забезпечуючи зручний доступ до даних всередині єдиного екземпляру.

Реалізація системи збереження даних. Архітектура системи запису та читання файлів збереження ІПС реалізована у тісній співпраці трьох елементів: абстрактного інтерфейсу *Serializable*, допоміжного класу *Archive* та класу *SaveManager*, який реалізує методи збереження й запису файлів, а також імплементує патерн «*Registry*» [3]. *SaveManager*, використовуючи метод *bind()*, зберігає *вказівники* на об'єкти *конкретних класів* у асоціативному контейнері, де ключем виступає заголовок конкретного об'єкту, що *інкапсульований* всередині методу *getHeader()* інтерфейсу *Serializable*. Зареєстровані об'єкти пізніше використовуються для збереження або завантаження їхніх даних шляхом *серіалізації*.

Процес серіалізації делегований класу *Archive*, в той час як *Serializable* лише абстрагує цей процес в методі *serialize()*. Цей метод використовує посилання на екземпляр *Archive*, та використовує *перевантажений класом Archive оператор &* для серіалізації або десеріалізації даних всередину нього. Напрямок операції визначається при створенні об'єкту класу *Archive*.

Для розділення даних різних реалізацій *Serializable* всередині файлу збереження, використовуються раніше згадані заголовки. Такий підхід дозволяє зрозуміти функції які дані відносяться до якого об'єкту, а також зробити файл більш зрозумілим у разі потреби в ручній модифікації. Задля безпеки, у файл також записується номер версії збереження, аби уникнути помилоку в разі завантаження застарілого формату файлу збереження.

Для здійснення операцій над файлом збереження, клас *SaveManager* реалізує методи *save()* та *load()*. Метод *save()* записує дані всередину файлу, ітерууючись крізь всі зареєстровані об'єкти, формуючи на виході структурований файл. Метод *load()* завантажує дані з файлу збереження, зчитуючи файл у пошуку збігу з заголовками зареєстрованих об'єктів. Знайшовши збіг, він використовує *Archive* у режимі десеріалізації для передачі даних з потоку в зміні відповідного зареєстрованого об'єкту.

Знаходження валідного шляху для зберігання файлу на різних операційних системах (ОС) делеговане класу *PathUtils*. Він утилізує директиви препроцесора та можливості C++17 бібліотеки *std::filesystem*. Метод *getSaveFilePath()* визначає шлях до цільової директорії залежно від ОС. Для визначення шляху в *Windows*, використовуючи функцію *_dupenv_s*, зчитується зміна середовища *APPDATA*. В свою чергу на *UNIX-подібних* [7] системах, як *Linux* чи *macOS* зчитується зміна *Home*. У разі відсутності цільової директорії, новий каталог створюється функцією *fs::create_directories()*.

Останнім етапом реалізації системи є комплексне тестування з метою верифікації коректності роботи розроблених алгоритмів, профілювання продуктивності та перевірки стабільності системи. Було проведено аналіз показників якості з вибірки з користувачів з різним досвідом в ігрових системах жанру казуальних головоломок.

Здійснено профілювання системи інструментами *Visual Studio*. Для навантажувального тестування системи перевірки зіткнень створено конфігурацію з ігровим полем у розмірі 432 тисяч клітинок. Величезна кількість даних не вплинула на час виконання операцій, навантаження на центральний процес залишилось передбачуваним, підтвердивши велику ефективність системи до масштабування ігрового поля. Споживання оперативної пам'яті під час

профілювання лишалось стабільним та передбачуваним. Також проведено навантажувальне тестування шляхом випромінювання одночасно 100 тисяч частинок методом *emit()* класу *ParticleManager*. Спостереження верифікують ефективність реалізованого методу. Система працює стабільно, використання оперативної пам'яті лишається передбачуваним незважаючи на використання 60 тисяч вершин, гарантуючи безперешкодний перехід між станами ігрової системи. Верифіковано кросплатформність системи використовуючи підсистему *WSL*. Підтверджена коректна генерація нативних файлів збірки системою *CMake* та успішну компіляцію під ОС *Linux*. Підтверджено коректність роботи інструменту *PathUtils* на інших платформах, який успішно знайшов цільову директорію *HOME* згідно зі стандартами *UNIX*.

В результаті аналізу показників якості та опитування респондентів, верифіковано високий рівень залученості користувачів. Новачкам легко адаптуватись до керування, графічного інтерфейсу системи та до ігрових подій завдяки адаптивній складності системи. Водночас досвідчені гравці не втрачають зацікавленості та залишаються в зоні найближчого розвитку.

Анотація. У рамках дослідження здійснено реалізацію спроектованої ігрової програмної системи з урахуванням виокремлених вимог. Використання патернів проектування й дотримання принципів *SOLID* створює гнучку архітектуру гарантуючи масштабованість кодової бази. Реалізована динамічна модель нестатичної конфігурованості забезпечує адаптивне коригування складності на основі порівняльного аналізу агрегованих даних ефективності користувача. Модель гарантує плавне коригування для кожної з передбачених конфігурацій ігрового циклу, утримуючи гравця в зоні найближчого розвитку. Основні модулі системи реалізовано, утилізуючи інструментарій бібліотеки *SFML* та ефективних нативних структур *STL*, що дозволило досягти константної асимптотичної складності $O(1)$ для критично важливих алгоритмів. Графічну підсистему розширено, а розрахунок візуальних ефектів оптимізовано інтеграцією *GLSL* фрагментних шейдерів. Використання цих інструментів забезпечує раціональне управління ресурсами, високу швидкодію, якісний аудіовізуальний відгук та стабільність ігрового процесу. Комплексне

тестування, а також профілювання системи інструментами Visual Studio, підтвердили стабільність системи за умов надмірного масштабування розміру ігрового поля та використання величезної кількості візуальних ефектів у вікні. Залучення респондентів верифікувало зручність графічного інтерфейсу системи та інтуїтивності керування. Використання системи збірки CMake й підсистеми WSL довело коректну та стабільну роботу системи на операційних системах.

Ключові слова: об'єктно орієнтоване програмування, дослідницьке навчання, варіативні моделі, математичне моделювання, змійка, казуальні головоломки, комп'ютерне моделювання, КОМСДН, ігрова система, AI.

Література

1. Hrybiuk Olena. Engineering in Educational Institutions: Standards for Arduino Robots as an Opportunity to Occupy an Important Niche in Educational Robotics in the Context of Manufacturing 4.0, in: *Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*. Volume 27-32, 2020. P. 770-785.
2. Hrybiuk O. Experience in Implementing Computer-Oriented Methodological Systems of Natural Science and Mathematics Research Learning in Ukrainian Educational Institutions. In: Machado J., Soares F. (eds) *Innovations in Mechatronics Engineering. Lecture Notes in Mechanical Engineering*, 2022. P. 55-68. Springer, Cham Online.
3. Hrybiuk O. Problems of expert evaluation in terms of the use of variative models of a computer-oriented learning environment of mathematical and natural science disciplines in schools, [w:] *Zeszyty Naukowe Politechniki Poznańskiej. Seria: Organizacja i Zarządzanie*, Zeszyt Nr 79, Poznań: Wydawnictwo Politechniki Poznańskiej (WPP), 2019. P. 101-119.
4. Гриб'юк О.О. Дослідницьке навчання учнів предметів природничо-математичного циклу з використанням комп'ютерно орієнтованих методичних систем / О. О. Гриб'юк. Монографія. Київ: НПУ імені М. П. Драгоманова, 2019.

УДК 004.942:[327

Максименко В. М. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О
Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-технічний університет імені академіка Юрія Бугая» (Київ)

МАТЕМАТИЧНІ ОСНОВИ МОДУЛЬНОЇ МЕДИЧНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБРОБКИ БІОМЕДИЧНИХ ДАНИХ ПАЦІЄНТІВ

Актуальність дослідження. Сучасна медична діагностика характеризується стрімким зростанням обсягів досліджень та підвищенням вимог

до їх точності й швидкості. В умовах високого навантаження ключовим завданням стає мінімізація впливу людського фактора на рутинні процеси. Головна ідея автоматизації полягає в тому, що кваліфікований фахівець повинен витратити свій ресурс на складні аналітичні завдання, а не на механічне перенесення даних.

Використання автоматичних аналізаторів дозволяє позбавити персонал від ресурсозатратної роботи за мікроскопом, звільняючи час для ретельного дослідження проблемних зразків. Проте, незважаючи на наявність високотехнологічного обладнання, рівень цифрової інтеграції в багатьох лабораторіях залишається недостатнім [1].

Сучасна лабораторія представлена у вигляді складного програмно-апаратного комплексу, де функціонують десятки аналізаторів різних виробників. Це породжує фундаментальну проблему інтеграції: хаос протоколів та відсутність єдиного стандарту взаємодії.

Мета дослідження полягає у розробленні модульної медичної інформаційної системи обробки біомедичних даних пацієнтів

У рамках дослідження здійснено ґрунтовний аналіз існуючих можливостей програмних систем задля визначення високорівневих потреб і розроблення зручної, ефективної та простої програмної системи. На основі проведеного аналізу предметної області та виявлених недоліків існуючих рішень, метою магістерської роботи є створення комплексної екосистеми *Logos*, яка забезпечує наскрізний потік медичних даних: від аналізатора до пацієнта. Для досягнення мети реалізовано три взаємопов'язані програмні модулі (рис. 1).

Універсальний драйвер (Connection Driver) - Автономний desktop-додаток для інтеграції будь-якого лабораторного обладнання. *Ключова особливість*: Локальна буферизація (SQLite) для захисту від збоїв мережі та гнучкий конструктор протоколів, що вирішує проблему "зоопарку" стандартів. *Сервіс*

інтелектуального аналізу (AI-Core) - API-шлюз для обробки та інтерпретації результатів. *Ключова особливість*: Використання LLM для генерації пояснень ("що це означає?"), з можливістю донавчання моделі/корегування інтерпретацій професійними, мотивованими фахівцями та персоналізованих рекомендацій, з обов'язковою анонімізацією персональних даних перед обробкою.

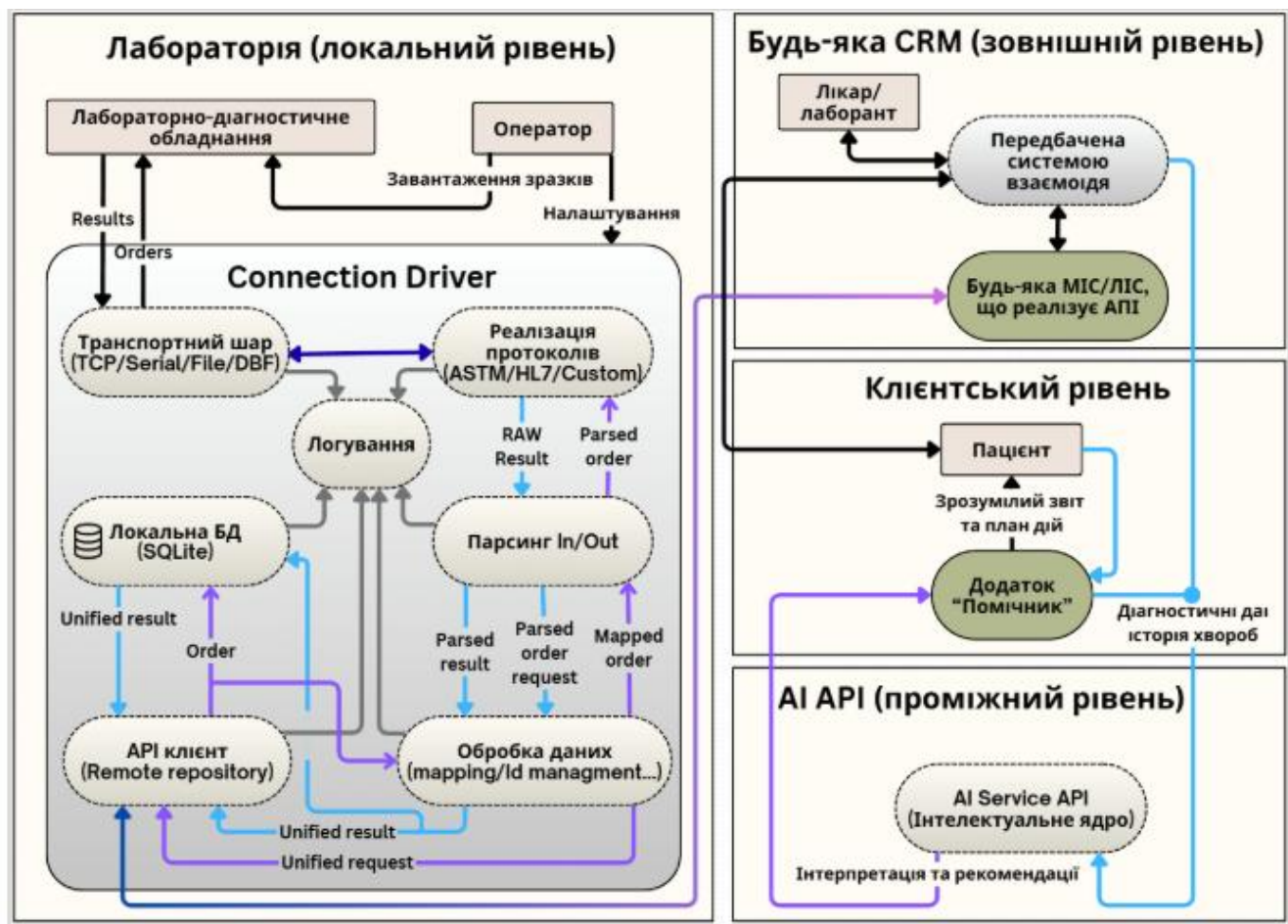


Рис. 1. Схема взаємодії модулів медичної інформаційної системи обробки біомедичних даних пацієнтів (TO-BE)
Джерело: опрацювання власне

Мобільний додаток пацієнта (Logos.Care) - персональний асистент для зберігання історії та навігації по здоров'ю. *Ключова особливість*: Централізація даних з різних джерел та проактивна підтримка (нагадування, трекери) замість пасивного архіву. Запропонована архітектура (TO-BE) вирішує проблему

фрагментації, об'єднуючи лабораторний процес та досвід пацієнта в єдиний інформаційний контур.

Враховуючи високу складність предметної області (лабораторна діагностика), різноманіття термінології та критичні вимоги до надійності даних, для проектування системи було обрано синтез трьох методологічних підходів [2]: *предметно-орієнтованого проектування (DDD), емпіричного аналізу та принципів чистої архітектури. Предметно-орієнтоване проектування (Domain-Driven Design, DDD)* - ключова методологія даної роботи. Вибір DDD зумовлений необхідністю точно відобразити складні бізнес-процеси лабораторії у програмному коді. Ниження архітектурних ризиків: у складних доменах (як-от ЛІС) найбільшою проблемою є невірні припущення розробників щодо логіки процесів, що призводить до дороговартісних переробок. DDD дозволяє створити точну модель домену, де структура коду відповідає реальним фізичним сутностям (*"Аналізатор", "Протокол передачі", "Результат"*). Єдина мова (Ubiquitous Language): Використання єдиного глосарію термінів як у комунікації з медичним персоналом, так і в назвах класів/методів. Це усуває "труднощі перекладу" між лікарем та розробником.

Регуляторна відповідність та безпека: Точна модель домену допомагає чітко розмежувати зони відповідальності (Bounded Contexts), що є критичним для відповідності вимогам безпеки даних. Наприклад, контекст *"Драйвер"* ізольований від контексту *"Персональні дані пацієнта"*, що зменшує ризик клінічних помилок. Проектування базується не лише на теоретичних викладках, а й на методі *включеного спостереження*. У рамках дослідження проведено глибоке занурення у процеси (досвід роботи медичним інженером та досвід інтеграції сторонніх ЛІС), що дозволило виявити приховані "болі" користувачів, які неможливо побачити ззовні (наприклад, реальні сценарії помилок при скануванні штрих-кодів

або проблеми з підключенням). Це дозволило сформулювати вимоги до *UX*, орієнтовані на реальні умови експлуатації, а не на ідеалізовані сценарії [3].

Менеджер штрих-кодів медичної інформаційної системи обробки біомедичних даних пацієнтів (рис. 2) дозволяє задати послідовність кодів, які будуть використовуватися для результатів, по принципу FIFO. Менеджер має можливість вказати кольорову схему, для візуальної зручності. Наприклад, якщо на одному робочому місці відкрито 2+ менеджерів для різних приладів, а відповідна кольорова схема дозволяє легко знайти потрібний менеджер [4].

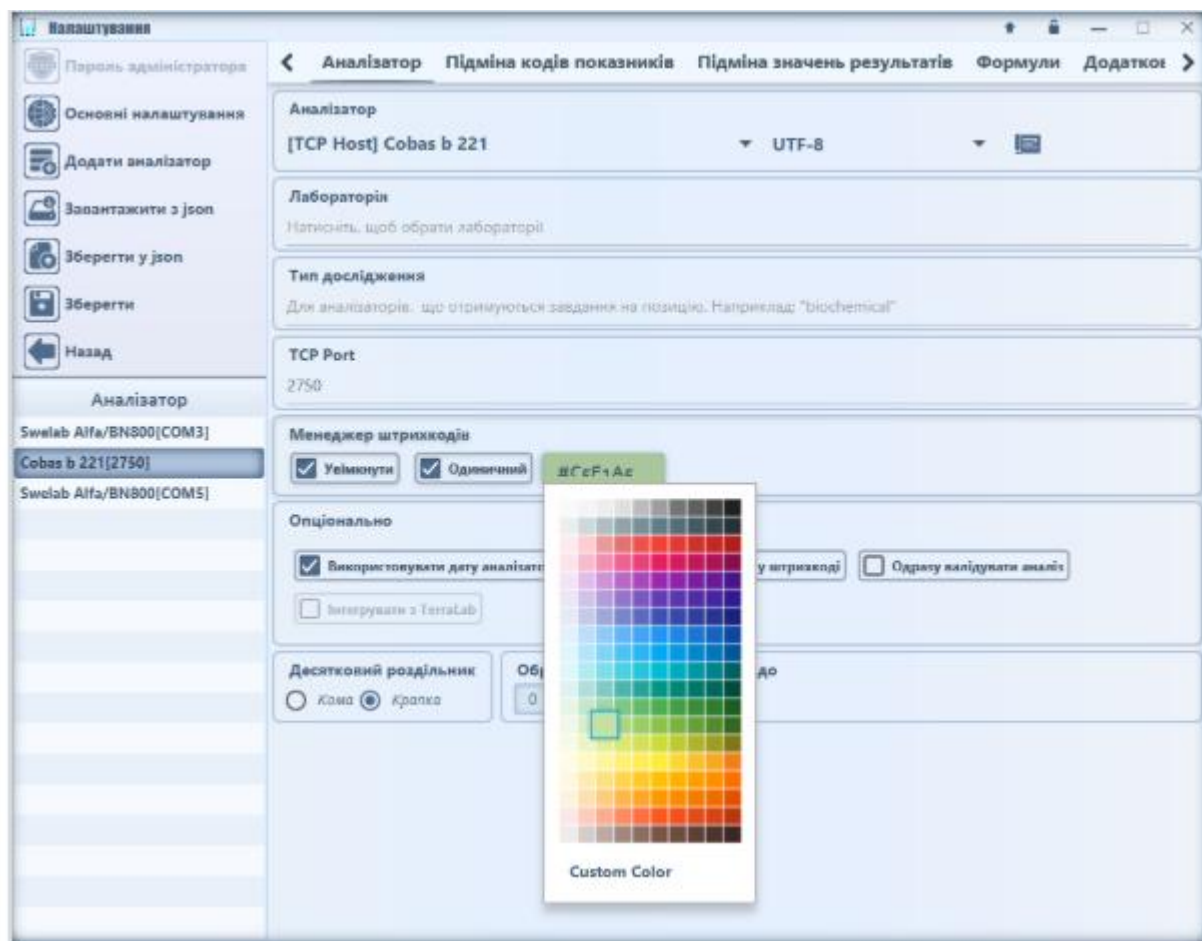


Рис. 2. Менеджер штрих-кодів медичної інформаційної системи обробки біомедичних даних пацієнтів
Джерело: опрацювання власне

Для технічної реалізації обрано комбінацію підходів, що не виключають, а доповнюють *DDD: Clean Architecture* забезпечує поділ коду на незалежні шари, що дозволяє змінювати зовнішні інтерфейси (UI) чи бази даних без втручання в бізнес-логіку. *Розподілена модульна архітектура*: Система реалізується як набір трьох автономних модулів (Драйвер, API, Мобільний додаток), кожен з яких може розроблятися та масштабуватися незалежно.

Анотація. У рамках дослідження запропоновано практичне вирішення фундаментальної проблеми інтеграції в сучасних лабораторіях: хаосу протоколів та залежності від постачальника (*vendor-lock*). Проект створює універсальний, гнучкий та надійний проміжний драйвер, який виступає "перекладачем" між будь-яким лабораторним обладнанням та будь-якою медичною інформаційною системою (МІС/ЛІС). Драйвер звільняє лабораторії від архітектурних обмежень, які нав'язують виробники обладнання та розробники ЛІС. Концепція ґрунтується на основних принципах: Інкапсуляція складності: драйвер бере на себе складну логіку взаємодії з численними специфічними протоколами (*Serial, TCP, File, ASTM, HL7*). Стандартизація взаємодії для зовнішньої системи (МІС/ЛІС) драйвер надає єдине, просте та стандартизоване *REST API*. Замість підтримки сотень протоколів, МІС потрібно реалізувати лише два ендпойнти: *SendResults* і *GetData*. Надійність та гнучкість: На відміну від простих драйверів, що інтегровані під конкретну ЛІС, *Connection Driver* є автономним додатком. Він забезпечує буферизацію даних у локальній *SQLite* базі (гарантуючи доставку при збоях мережі) та надає потужний графічний інтерфейс (UI) для гнучкого налаштування мапінгу кодів, формул та бізнес-логіки. Таким чином, драйвер перетворює хаотичну взаємодію "багато-до-багатьох" на керовану та надійну архітектуру "багато-до-одного".

Ключові слова: модульна медична система, дослідницьке навчання, варіативні моделі, *CleverCOMSRL*, КОМСДН, інформаційна система, інтелектуальний аналіз даних, математичне моделювання, біомедичні дані, AI.

Література

1. Hrybiuk, O., Kant, G.S. *CleverCOMSRL: Implementation of an AI Computer-Aided Design System in the Context of the Cognitive Science Paradigm for the Research Training Process*. In: Machado, J., et al. *Innovations in Mechatronics Engineering III*. *icieng* 2024. *Lecture Notes in Mechanical Engineering*. Springer, Cham. 2024. P. 351-362. https://doi.org/10.1007/978-3-031-61575-7_32.
2. Khang A., Abdullayev V., Hrybiuk O., & Shukla A.K. (Eds.). *Computer Vision and AI-Integrated IoT Technologies in the Medical Ecosystem* (1st ed.). CRC Press. 2024. <https://doi.org/10.1201/9781003429609>.

3. Hrybiuk, O.O. Research learning of the natural science and mathematics cycle using computer-oriented methodological systems. Monograph. Drahomanov NPU, Kyiv, 2019. P. 179–185.
4. Hrybiuk, O. Improvement of the educational process by the creation of centers for intellectual development and scientific and technical creativity. In: Hamrol, A., Kujawińska, A., Barraza, M.F.S. (eds.) *MANUFACTURING 2019*. LNME, 2019. P. 370–382. Springer, Cham. https://doi.org/10.1007/978-3-030-18789-7_31.

Maxim Ptashniy (Master's student)
Jagiellonian University, Krakow, Poland,
 Research Advisor - Associate Professor O. Ptashnyi
Kharkiv National automobile and Highway University
 Maksim Yatsyna (undergraduate student, 3year),
 Yurii Dubovskyi (undergraduate student, 3year),
 Research Advisor - Associate Professor Bobrytskay G. S.
Ivan Kozhedub Kharkiv National Air Force University

SEARCHING FOR MUTUALLY EXCLUSIVE PROPERTIES IN SELECTION PROBLEMS USING ULTRAFILTERS

Introduction. The mathematical theory of voting studies how collective decisions can be de-ri-ved from individual preferences. Given a set of voters and a set of possible alternatives, the goal is to design a rule which aggregates the preferences of individuals into a single collective decision.

Although the theory is often motivated by political elections, the same mathematical ideas apply in many everyday situations. For example, a family deciding which movie to watch in the evening may also use some form of voting rule based on individual preferences.

Formally, suppose that a set V of voters must choose an element from a finite set A of alternatives. Each voter has a preference ordering on A . A voting rule is a procedure that assigns a social outcome based on the collection of these preferences.

Examples of Voting Systems. Different societies may use different voting procedures.

Dictatorship. In a dictatorship the outcome of the election always coincides with the preference of a single distinguished voter.

Monarchy. A related system is monarchy, where a single alternative is always chosen regardless of the preferences of voters.

Majority Rule. A common democratic rule is the majority system: an alternative wins if it receives more votes than any other alternative.

Absolute Majority. In some systems a candidate must obtain more than 50% of the votes in order to win.

Block Voting. In block voting the electorate may be divided into blocks. Within each block voters choose a preferred candidate, and the candidate who wins a block receives all the votes of that block.

This system is sometimes described as a “winner takes all” mechanism inside each block.

The Borda Count. Another well known system is the Borda count. Each voter ranks the alternatives. Points are assigned according to position in the ranking. For example, if there are n candidates, the first place receives $n - 1$ points, the second place $n - 2$ points, and so on.

The candidate with the largest total number of points is declared the winner.

For example, suppose three candidates A, B, C and three voters:

Voter 1	Voter 2	Voter 3
A	B	B
B	C	A
C	A	C

Using the Borda rule we assign 2 points for first place, 1 point for second place and 0 points for third place. The resulting scores determine the winner.

Desirable Properties of Voting Rules. Many properties of voting systems appear natural from an intuitive point of view.

Anonymity. A voting rule is anonymous if the outcome does not depend on the identity of voters.

Neutrality. A rule is neutral if all alternatives are treated symmetrically.

Majority Criterion. If a candidate is ranked first by a strict majority of voters, that candidate should win the election.

Weak Pareto Principle. If every voter prefers alternative A to alternative B , then the social outcome should not rank B above A .

Independence of Irrelevant Alternatives. The relative ranking of two alternatives should depend only on the individual preferences between these two alternatives. In other words if relation between A and B did not change for anyone between two models, the relative result of voting between A and B cant change either.

Condorcet Criterion. If a candidate defeats every other candidate in pairwise comparisons, that candidate should be the winner. At first sight these conditions seem extremely reasonable. However, it quickly becomes clear that no voting system can satisfy all of them simultaneously.

For example, dictatorship violates anonymity, while monarchy violates neutrality.

The Condorcet Paradox. Even very simple voting situations may lead to surprising results.

Consider three candidates A, B, C and three voters:

Voter 1	Voter 2	Voter 3
A	B	C
B	C	A
C	A	B

Now compare candidates pairwise.

- A is preferred to B by voters 1 and 3.
- B is preferred to C by voters 1 and 2.
- C is preferred to A by voters 2 and 3.

Thus we obtain

$$A \succ B, \quad B \succ C, \quad C \succ A$$

which forms a cycle.

This phenomenon is known as the Condorcet paradox and demonstrates that collective preferences may become inconsistent even when individual preferences are perfectly rational.

4. Arrow's Impossibility Theorem. One of the most famous results in social choice theory is the impossibility theorem proved by Kenneth Arrow.

Roughly speaking, Arrow showed that when there are at least three alternatives, any voting rule satisfying several natural conditions must necessarily be a dictatorship.

The proof proceeds through several steps.

First, one proves that for every pair of alternatives there exists a decisive voter whose preference determines the social comparison between these two alternatives.

Then one shows that if a voter is decisive for a certain pair of alternatives, that voter must also control other comparisons.

Finally it follows that a single voter determines the entire social ordering. In other words, the system becomes a dictatorship.

Theorem 1 (Arrow's Impossibility Theorem). *Suppose that*

- *the set of alternatives satisfies $|A| \geq 3$,*
- *the set of voters satisfies $|V| \geq 2$. Let a social welfare function satisfy*
- *the weak Pareto principle,*
- *independence of irrelevant alternatives. Then the rule must be a dictatorship.*

A dictator is a voter whose preferences always determine the social ranking of alternatives.

Ultrafilters and Voting.

Social Welfare Functions. Let V be a finite set of voters and A a finite set of alternatives.

Each voter has a preference ordering on A . A voting rule that produces a complete social ranking of alternatives is called a social welfare function.

Formally, a social welfare function assigns to every profile of individual preference orderings a collective ordering on A .

Earlier we considered voting rules that output a set of winners. Such rules are often called social choice functions.

A social welfare function is a stronger object: it produces a complete ranking of all alternatives. It appears that Arrow's theorem can be extended on these systems.

Stronger Arrow's theorem. There is a more abstract formulation of Arrow's theorem using ultrafilters.

Consider the set of voters V . A subset $S \subseteq V$ is called decisive if whenever all voters in S prefer A to B , the social outcome also ranks A above B .

The collection of all decisive sets has remarkable structure. It turns out that these sets form an ultrafilter.

A family $\mathcal{U}$ of subsets of V is called an ultrafilter if it satisfies

- $\emptyset \notin \mathcal{U}$,
- if $A, B \in \mathcal{U}$ then $A \cap B \in \mathcal{U}$,
- if $A \in \mathcal{U}$ and $A \subseteq B$ then $B \in \mathcal{U}$,
- for every $A \subseteq V$ either $A \in \mathcal{U}$ or $V \setminus A \in \mathcal{U}$.

Theorem 2. *Every ultrafilter on a finite set is principal.*

Proof. Let V be finite and let U be an ultrafilter on V .

Assume first that there exists $x \in V$ such that $\{x\} \in U$. Then every set in U must contain x . Otherwise the intersection of this set with $\{x\}$ would be empty, which contradicts the fact that $\emptyset \notin U$.

Thus U consists precisely of all subsets containing x .

Now suppose that no singleton belongs to U . Then for every $x \in V$ the complement $V \setminus \{x\}$ must belong to U .

Taking intersections of all these sets we obtain

$$\bigcap_{x \in V} (V \setminus \{x\}) = \emptyset$$

which contradicts the property that the intersection of elements of an ultra-filter must also belong to the ultrafilter. Therefore some singleton must belong to U .

In the language of voting theory this means that decisive coalitions must contain a particular voter. Therefore the voting rule is determined by a single individual, which corresponds to dictatorship.

Conclusion. The mathematical theory of voting reveals a striking phenomenon. Many natural conditions that appear essential for fairness can not be satisfied simultaneously. Although no perfect voting system exists, studying the trade-offs between different properties provides valuable insight into the structure of collective decision making.

References

1. J. K. Hodge, R. E. Klima, The mathematics of voting and elections, American Mathematical Society, 2005
2. E.A. Robinson, D. H. Ullman, A mathematical look at politics, CRC Press, 2011.
3. D. G. Saari, Chaotic elections. A mathematicians looks at voting, American Mathematical Society, 2001.

Міщенко М.С (здобувач бакалаврату, 1 курс),
Кравчик Д.В. (здобувач бакалаврату, 1 курс),
Науковий керівник – проф. Ємельянова Т. В.
Харківський національний автомобільно-дорожній університет

ХАРАКТЕРИСТИКИ ТА ЗАКОНИ РОЗПОДІЛУ ЕЛЕМЕНТАРНИХ ВИПАДКОВИХ ФУНКЦІЙ, ЯК СКЛАДОВИХ ВИПАДКОВИХ ПРОЦЕСІВ

Відомо, що в більшості задач технологічної галузі слід визначати фактори, які мають випадковий характер. Випадкові фактори присутні в прикладних задачах як випадкові функції. Одже, при моделюванні будь-якого явища треба враховувати вплив випадкового фактору. Цей вплив описується випадковими функціями. Наведемо кілька прикладів впливу випадкових факторів на технологічний процес:

вібраційний процес, як випадковий процес, якому піддаються колеса автомобіля при русі по ґрунтовій дорозі; вібрації опор моста в напрямку перпендикулярному їх перетину при русі вантажного автотранспорту; відхилення закріпленого вантажу від заданого положення під дією випадкових збурень через «старіння» рельсового покриття при русі залізничного потягу; ордината профілю обробленої поверхні, яка випадковим чином змінюється вздовж даної траси; вплив випадкових збурень в електричних ланцюгах на роботу автоматизованих пристроїв тощо.

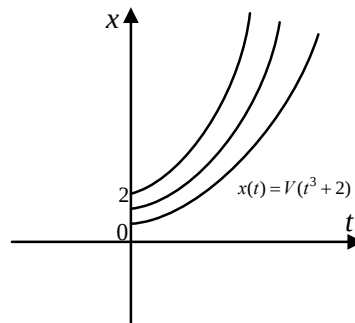
Побудова математичної моделі явища, що знаходиться під впливом випадкового фактору, потребує визначення його випадкового характеру, вивчення динаміки розвитку та закономірності випадкового явища.

В більшості випадків випадкові явища характеризуються складною структурою. Одже випадковий процес можна розглядати як результат дії кількох випадкових збурень. В ряді завдань випадкові процеси зручно уявляти випадкове явище за допомогою елементарних випадкових функцій.

Елементарними випадковими функціями визначаються функції, в яких залежність від аргументу t є не випадковою функцією, випадкові величини входять у вигляді параметрів, що є незалежними від аргументу t .

Приклад 1. Елементарна випадкова функція має вигляд $X(t) = V(t^3 + 2)$, ($t > 0$), де V – випадкова величина, розподілена за рівномірним розподілом на відріжку $[0; 1]$.

Реалізації функції $X(t)$ представлені на рис. 1.



Закон розподілу випадкової величини V визначається за умовами задачі

$$f_V(v) = \begin{cases} 1, & v \in [0; 1], \\ 0, & v \notin [0; 1]. \end{cases}$$

Математичне сподівання $M[V]$ для випадкової величини V , рівномірно розподіленої на відріжку $[0; 1]$, дорівнює

$$M[V] = \frac{1}{2}.$$

Математичне сподівання випадкового процесу $X(t) = V(t^3 + 2)$

$$M[X(t)] = M[V(t^3 + 2)] = M[V](t^3 + 2) = \frac{1}{2}(t^3 + 2)$$

Дисперсія випадкової величини X , розподіленої за рівномірним законом в $[0; 1]$ дорівнює

$$D[V] = \frac{1}{12}.$$

Дисперсія випадкового процесу

$$D[X(t)] = D[V(t^3 + 2)] = D[V](t^3 + 2)^2 = \frac{1}{12}(t^3 + 2)^2.$$

Кореляційна функція $K_X(t_1, t_2)$

$$\begin{aligned} K_X(t_1, t_2) &= M[(X(t_1) - m_X(t_1))(X(t_2) - m_X(t_2))] = \\ &= M[(V(t_1^3 + 2) - \frac{1}{2}(t_1^3 + 2))(V(t_2^3 + 2) - \frac{1}{2}(t_2^3 + 2))] = \\ &= M[V - \frac{1}{2}]^2(t_1^3 + 2)(t_2^3 + 2) = \frac{1}{12}(t_1^3 + 2)(t_2^3 + 2). \end{aligned}$$

де

$$M[V - \frac{1}{2}]^2 = \int_0^1 (V - \frac{1}{2})^2 dV = \frac{1}{3}(V - \frac{1}{2})^3 \Big|_0^1 = \frac{1}{12}$$

Закон розподілу елементарної випадкової функції $X(t)$ має рівномірний характер на відрітку $[0; t^3 + 2]$

$$f_X(x, t) = \begin{cases} \frac{1}{t^3 + 2}, & x \in [0; t^3 + 2], \\ 0, & x \notin [0; t^3 + 2]. \end{cases}$$

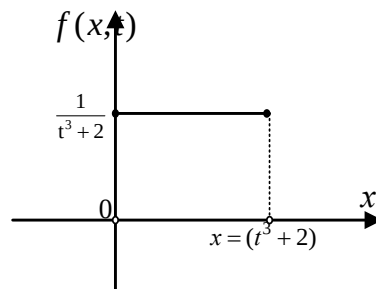


Рис. 1. Одновімірний закон розподілу елементарної випадкової функції $X(t) = V(t^3 + 2)$

Приклад 2. Елементарна випадкова функція має вигляд $X(t) = Ve^{-t}$, ($t > 0$), де V – випадкова величина, розподілена за нормальним розподілом з параметрами m , σ .

Реалізації функції $X(t)$ представлені на рис.2.

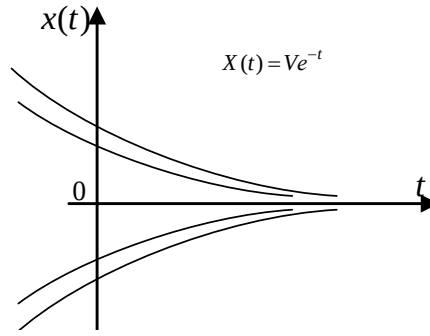


Рис. 2. Реалізації випадкової функції $X(t)$

Математичне сподівання $M[V]$ для випадкової величини V , розподілена за нормальним розподілом, дорівнює m . σ .

Математичне сподівання випадкового процесу $X(t) = Ve^{-t}$

$$M[X(t)] = M[Ve^{-t}] = M[V]e^{-t} = me^{-t}.$$

Дисперсія випадкової величини X , розподіленої за нормальним розподілом, дорівнює σ^2

$$D[V] = \sigma^2.$$

Дисперсія випадкового процесу

$$D[X(t)] = D[Ve^{-t}] = D[V](e^{-t})^2 = \sigma^2 e^{-2t}.$$

Кореляційна функція $K_X(t_1, t_2)$

$$\begin{aligned} K_X(t_1, t_2) &= M[(X(t_1) - m_X(t_1))(X(t_2) - m_X(t_2))] = \\ &= M[(Ve^{-t_1} - me^{-t_1})(Ve^{-t_2} - me^{-t_2})] = \\ &= M[V - m]^2 e^{-t_2} = (MV^2 - m^2)e^{-(t_1+t_2)} = \sigma^2 e^{-(t_1+t_2)}. \end{aligned}$$

Для будь-якого фіксованого моменту часу t випадкова функція $X(t)$, як лінійна функція розподілу нормальної випадкової величини, має нормальний розподіл з характеристиками $m_X(t) = me^{-t}$, $\sigma_X(t) = \sigma^2 e^{-2t}$. Одновимірний закон розподілу елементарної випадкової функції $X(t) = Ve^{-t}$, ($t > 0$), нормальний і має вигляд

$$f_X(x, t) = \frac{1}{\sqrt{2\pi\sigma}e^{-t}} \exp\left[-\frac{(x - me^{-t})^2}{2\sigma^2 e^{-2t}}\right].$$

Анотація. При побудові математичної моделі будь якого явища слід ураховувати можливі впливи випадкових факторів, що призводять до випадкових процесів у досліджуємой системі. В ряді задач випадкові процеси можна виразити за допомогою елементарних випадкових функцій, характеристики яких достатньо просто обчислити. В роботі наведено приклади елементарних випадкових функцій, в яких залежність від аргументу t є не випадковою функцією, випадкові величини входять у вигляді параметрів, що є незалежними від аргументу t .

Ключові слова: випадкові фактори, елементарні випадкові функції, реалізації випадкової функції, одновимірний закон розподілу елементарної випадкової функції.

Література

1. Несвіт М.І., Поклонський Є.В.. Випадкові процеси: навчально-методичний посібник. – Х. : ХНУБА, 2013 – 51 с. . Теорія випадкових процесів:
2. Задачі для самостійної роботи [Електронний ресурс] : навч. посіб. для студ. спеціальності 171 «Електроніка»/ КПІ ім. Ігоря Сікорського ; уклад.: О.В. Гармаш. – Електронні текстові дані (1 файл: 1,771 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 44 с.

УДК 004.942:[327

Миколіук П. О. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О
Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-технічний університет імені академіка Юрія Бугая» (Київ)

РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ СТРУКТУРНОГО МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ

Актуальність дослідження. У сучасних умовах цифрової трансформації бізнесу організації стикаються з необхідністю швидкої адаптації до змін конкурентного середовища, посилення вимог до якості управлінських рішень та оптимізації ресурсів. Значне ускладнення бізнес-процесів, використання розподілених інформаційних систем, впровадження стандартів корпоративного управління та інтеграція інноваційних технологій створюють потребу у системах, що дозволяють формально, точно та візуально описувати діяльність організацій [1]. Структурне моделювання бізнес-процесів стало одним із ключових інструментів сучасного менеджменту та цифрової трансформації.

Попри наявність великої кількості інструментів, існує проблема їх фрагментарності та складності використання для користувачів без технічної підготовки. Частина рішень є надлишковими та орієнтованими на корпорації, що робить їх непридатними для малого та середнього бізнесу. Інші мають обмежені функції моделювання або не підтримують адаптивне редагування процесів. Також відчувається нестача локалізованих інструментів, орієнтованих на український ринок.

Актуальною є розробка легкої, інтуїтивної та функціональної інформаційної системи структурного моделювання бізнес-процесів, що поєднуватиме переваги візуального моделювання, гнучкість нотацій та можливості модернізації, й водночас відповідатиме потребам сучасних організацій. *Мета дослідження* полягає у здійсненні проектування і розроблення інформаційної системи структурного математичного моделювання бізнес-процесів.

У ході дослідження було вивчено актуальне питання вибору архітектури, інструментів та алгоритмів для побудови інформаційної системи структурного моделювання бізнес-процесів. Розроблено прототип веб-орієнтованої інформаційної системи структурного моделювання бізнес-процесів. Основним призначенням системи є забезпечення користувачів інструментами для побудови, редагування та аналізу моделей бізнес-процесів із використанням нотації BPMN 2.0 (рис. 1).

Прототип реалізовано на етапі проектування з метою візуалізації функціональних можливостей системи, перевірки логіки взаємодії користувача з основними модулями та формування уявлення про архітектуру майбутнього програмного продукту. Для розроблення прототипу використано середовище UI/UX-проектування Figma із застосуванням засобів автоматизованої генерації інтерфейсів [2]. Розроблений прототип інформаційної системи складається з таких основних функціональних модулів: модуль аутентифікації користувачів; модуль керування бізнес-процесами; модуль створення моделей бізнес-

процесів; модуль структурного моделювання; модуль налаштування параметрів BPMN-елементів; модуль перегляду ієрархічної структури процесу.

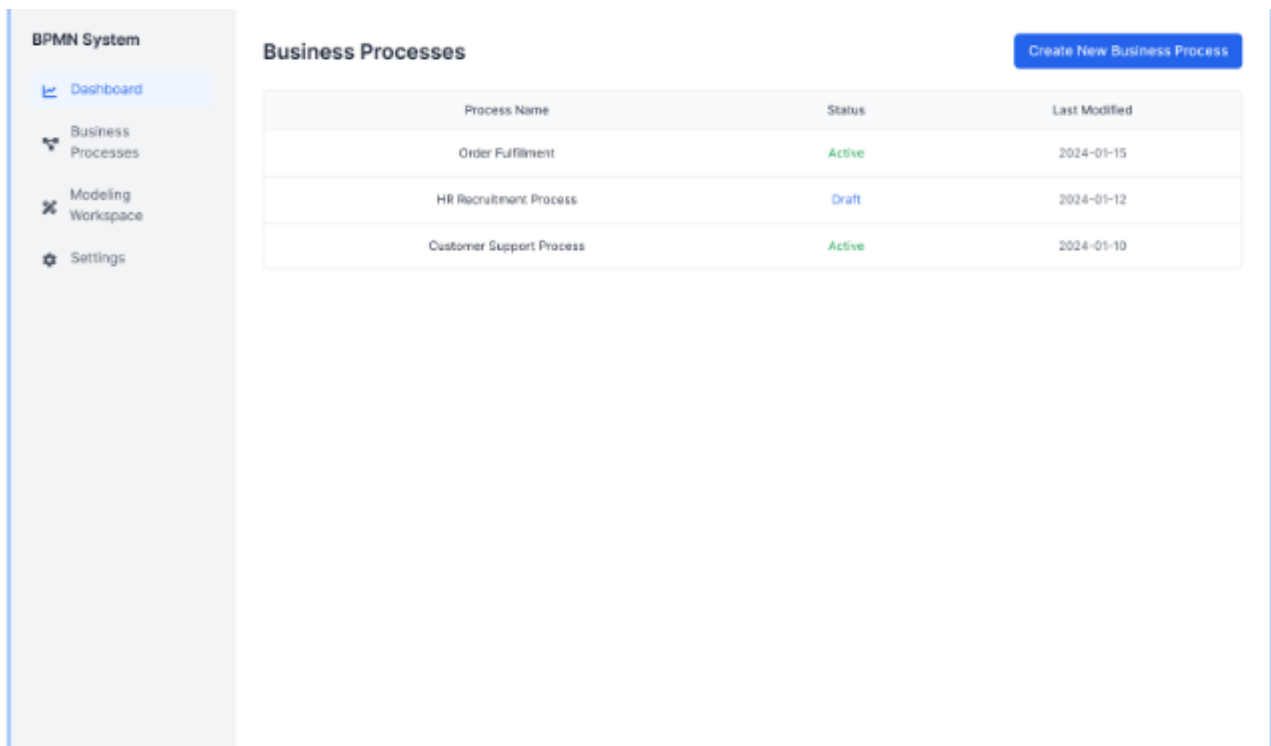


Рис. 1. Інтерфейс головної сторінки інформаційної системи структурного моделювання бізнес-процесів
Джерело: опрацювання власне

Кожен із модулів реалізовано у вигляді окремого інтерфейсного екрана, що забезпечує логічну цілісність та зручність використання системи. Інтерфейс створення нової моделі бізнес-процесу дозволяє користувачу задати назву процесу, його опис та обрати нотацію моделювання.

У рамках дослідження використовувався стандарт BPMN 2.0 як найбільш поширений та формалізований засіб опису бізнес-процесів [3]. Ключовим компонентом системи є робоча область структурного моделювання бізнес-процесів (рис. 2), що містить панель елементів BPMN, центральне полотно для побудови процесу та інструменти керування моделлю, зокрема збереження та експорту.

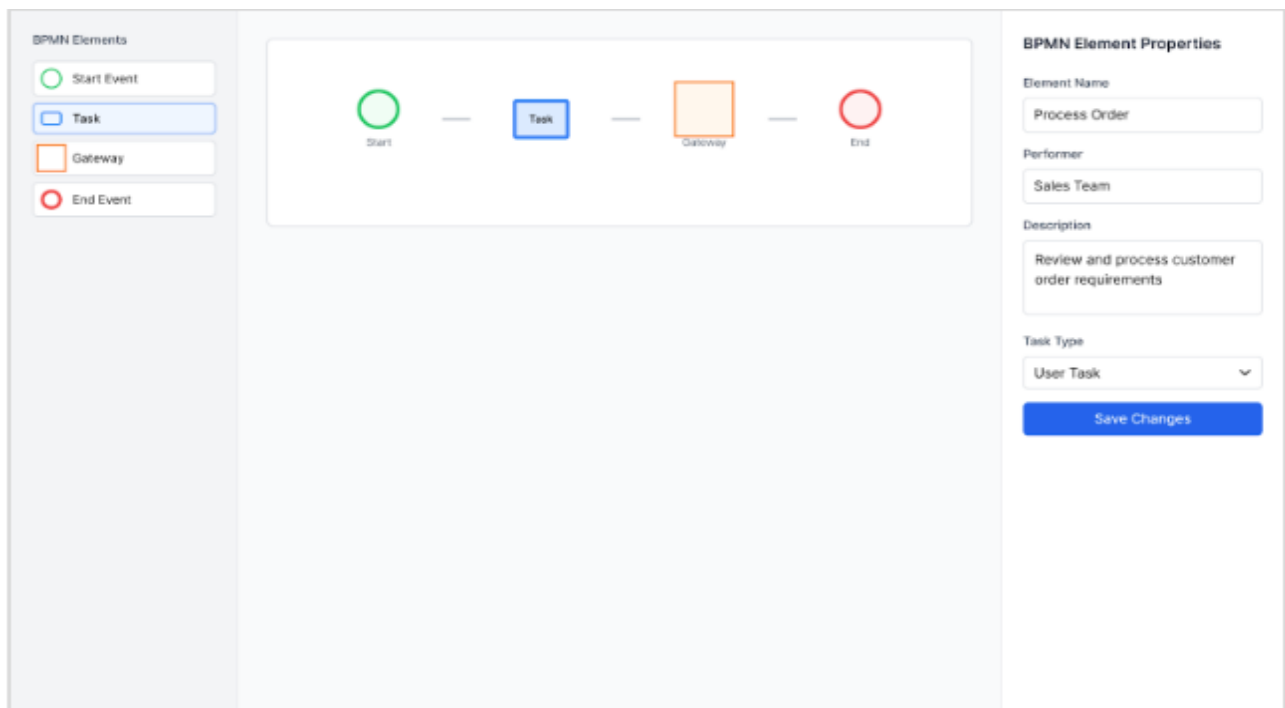


Рис. 2. Інтерфейс головної сторінки інформаційної системи структурного моделювання бізнес-процесів
Джерело: опрацювання власне

Розроблений модуль забезпечує редагування атрибутів BPMN-елементів, таких як назва, виконавець, опис та тип задачі. Використовувалося поєднання кількох підходів структурного моделювання (DFD, IDEF0) в одному програмному середовищі [4]. Здійснено проектування уніфікованої структури даних, із використанням якої забезпечується перенесення моделей між різними платформами.

Анотація. На основі проведених досліджень здійснено проектування і розроблено прототип інформаційної системи структурного моделювання бізнес-процесів. У рамках дослідження сформовано цілісне уявлення про функціональні можливості та інтерфейсну структуру системи в контексті структурного моделювання бізнес-процесів. Розроблено адаптивну модель представлення бізнес-процесів із можливістю швидкої трансформації та автоматичного аналізу структури. Отримані результати можуть бути використані як основа для подальшої програмної реалізації та розширення функціоналу інформаційної системи структурного моделювання бізнес-процесів.

Ключові слова: CleverCOMSRL, структурне моделювання, дослідницьке навчання, варіативні моделі, бізнес-процеси, комп'ютерне моделювання, КОМСДН, інформаційна система, математичне моделювання, AI.

Література

1. Hrybiuk, O. Improvement of the educational process by the creation of centers for intellectual development and scientific and technical creativity. In: Hamrol, A., Kujawińska, A., Barraza, M.F.S. (eds.) *MANUFACTURING 2019*. LNME, 2019. P. 370–382. Springer, Cham. https://doi.org/10.1007/978-3-030-18789-7_31.
2. Hrybiuk, O., Kant, G.S. CleverCOMSRL: Implementation of an AI Computer-Aided Design System in the Context of the Cognitive Science Paradigm for the Research Training Process. In: Machado, J., et al. *Innovations in Mechatronics Engineering III. icieng 2024. Lecture Notes in Mechanical Engineering*. Springer, Cham. 2024. P. 351-362. https://doi.org/10.1007/978-3-031-61575-7_32.
3. Hrybiuk, O.O. Research learning of the natural science and mathematics cycle using computer-oriented methodological systems. Monograph. Drahomanov NPU, Kyiv, 2019. P. 277–325.
4. Khang A., Abdullayev V., Hrybiuk O., & Shukla A.K. (Eds.). *Computer Vision and AI-Integrated IoT Technologies in the Medical Ecosystem* (1st ed.). CRC Press. 2024. <https://doi.org/10.1201/9781003429609>.

УДК 004.942:[327

Миронюк С. О. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О
Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-технічний університет імені академіка Юрія Бугая» (Київ)

ДОСЛІДЖЕННЯ І РЕАЛІЗАЦІЯ ПРИКЛАДНИХ МАТЕМАТИЧНИХ МЕТОДІВ РОЗПІЗНАВАННЯ СИМВОЛЬНО-ЦИФРОВИХ ЗНАКІВ

Актуальність дослідження. У процесі підготовки до реалізації проєкту було проведено систематичний підбір матеріалів, що охоплюють алгоритми обробки цифрових зображень, бібліотеки Python для комп'ютерного зору та приклади застосування хеш-функцій у реальних задачах. Основну увагу зосереджено на перцептивному хешуванні (*pHash*), методах виокремлення ключових точок (*ORB*, *SIFT*, *SURF*), способах збереження дескрипторів у БД, а також підходах до візуалізації результатів у користувацькому інтерфейсі.

Мета дослідження полягає у методично виваженому доборі та доцільній реалізації методів розпізнавання символно-цифрових знаків, відповідному розробленні ефективного алгоритму ідентифікації зображень на основі візуального хешування та виявлення ключових точок, з урахуванням продуктивності, точності та зручності використання.

У рамках дослідження здійснено ґрунтовний аналіз існуючих можливостей програмних систем задля визначення високорівневих потреб і розроблення зручної, ефективної та простої програмної системи. Проведено аналіз методів перцептивного хешування (*aHash*, *pHash*, *dHash*) та алгоритмів *ORB*, *SIFT*, *SURF*, обґрунтовано доцільність вибору технологій обробки зображень. Здійснено проєктування архітектури ПС для автоматизованого визначення схожих зображень [1]. Для реалізації алгоритмів обрано мову програмування Python, що має широку підтримку бібліотек: *OpenCV*, *imagehash*, *Pillow*, *NumPy*, *CustomTkinter*, *sqlite3*. Обрані інструменти відповідають вимогам завдання – висока швидкість розробки, кросплатформеність, візуальна підтримка, простота в навчанні. Під час підбору матеріалу були вивчені наукові джерела, документація до бібліотек, open-source реалізації, приклади з репозиторіїв GitHub, що дозволило сформулювати повне уявлення про можливості та обмеження методів, які застосовуються. Розроблено програмну систему за допомогою *Python*, *OpenCV*, *CustomTkinter*. Реалізовано інтерфейс з інтуїтивно зрозумілим керуванням. Проведено тестування на реальних зображеннях для валідації ефективності.

Аналіз існуючих рішень (вивчено три ключові підходи): *perceptual Hashing* (*pHash*) генерує унікальний «відбиток» зображення, зберігаючи інваріантність до масштабування та незначних змін. *Ключові точки* (*ORB*, *SIFT*) дозволяють зіставляти деталі зображень навіть після трансформацій. *Глибоке навчання* (*CNN*) не використовується через складність реалізації, потребу в потужному обладнанні та великих даних. Обране рішення: *комбінування pHash і ORB* дозволяє досягти балансу між точністю, швидкістю та простотою реалізації.

Архітектура програмного продукту (програма реалізована за модульним принципом): модуль інтерфейсу користувача – побудований на бібліотеці *CustomTkinter*, відповідає за завантаження зображення, виведення результатів, вибір дій. Модуль попередньої обробки зображення виконує зменшення розміру, переведення в *grayscale*, видалення шумів. Модуль ідентифікації

реалізує генерацію *pHash*, виявлення ключових точок *ORB*, розрахунок метрик подібності [2]. Модуль БД (*SQLite*) зберігає результати аналізу, хеші, дескриптори та назви зображень.

У рамках дослідження здійснено імплементацію локальної адаптивної бінаризації. Актуалізувати стратегію сусідської універсальної бінаризації на основі стратегії Соволи та Пієтікайнена, яка може бути відомим вибором для таких завдань. Перетворення у відтінки сірого (рис. 1) гарантує, що зображення в відтінках сірого, оскільки локальна порогова обробка зазвичай працює з одно канальними даними.

```
private Bitmap ApplyAdaptiveThresholding(Bitmap image)
{
    AForge.Imaging.Filters.BradleyLocalThresholding threshold = new AForge.Imaging.Filters.BradleyLocalThresholding();
    // Ці параметри можна налаштовувати
    threshold.WindowSize = 41;
    threshold.PixelBrightnessDifferenceLimit = 0.1f;
    threshold.ApplyInPlace(image);
    return image;
}

private Bitmap LocalAdaptiveThresholding(Bitmap original, int windowSize, float k) // Змініть double на float тут
{
    // Конвертуємо зображення у відтінки сірого, якщо воно ще не у такому форматі
    Bitmap grayScaleBitmap = ConvertToGrayscale(original);

    // Застосування локального адаптивного порогу
    AForge.Imaging.Filters.BradleyLocalThresholding threshold = new AForge.Imaging.Filters.BradleyLocalThresholding();
    threshold.WindowSize = windowSize;
    threshold.PixelBrightnessDifferenceLimit = k; // Тут немає потреби в зміні, оскільки k уже буде float типу

    // Застосування порогу
    threshold.ApplyInPlace(grayScaleBitmap);

    return grayScaleBitmap;
}
```

Рис. 1. Імплементація локальної адаптивної бінаризації

Налаштування порогу використовується клас *BradleyLocalThresholding* з *AForge.NET*. Це реалізація техніки локального порогу, дуже ефективна для документальних зображень. Ви можете налаштувати *WindowSize* для локальних варіацій яскравості та *PixelBrightnessDifferenceLimit*, який впливає на чутливість порогу. Метод застосовує поріг безпосередньо до зображення, змінюючи оригінальні дані зображення [3]. Інтегруємо відповідні бібліотеки

слід переконатись, що у нашому прєкті додано посилання на бібліотеку AForge.NET. Її можна отримати через NuGet. Замінити або додати виклик LocalAdaptiveThresholding у розділ попередньої обробки зображень перед виконанням OCR.

Анотація. В рамках дослідження розроблено програмну систему ідентифікації цифрових зображень у контексті реалізації методів розпізнавання символно-цифрових знаків. Опрацьовано сучасні методи хешування та аналізу візуальних ознак. Система протестована, оптимізована, продемонструвала точність 94%. Побудовано зручний інтерфейс користувача. Сформовано технічну документацію, графічні схеми, відповідні інструкції. У перспективі програмна система має потенціал для подальшого розвитку, включаючи розширення функціоналу шляхом забезпечення нейромережевої класифікації, інтеграцію з мобільними платформами та розроблення нових інструментів для професійного використання, у т.ч. шляхом здійснення пакетного аналізу зображень.

Ключові слова: нейронна мережа, дослідницьке навчання, варіативні моделі, математичне моделювання, КОМСДН, інформаційна система, комп'ютерне моделювання, розпізнавання символно-цифрових знаків, AI.

Література

1. Khang A., Abdullayev V., Hrybiuk O., & Shukla A.K. (Eds.). Computer Vision and AI-Integrated IoT Technologies in the Medical Ecosystem (1st ed.). CRC Press. 2024. <https://doi.org/10.1201/9781003429609>.
2. Hrybiuk, O.O. Research learning of the natural science and mathematics cycle using computer-oriented methodological systems. Monograph. Drahomanov NPU, Kyiv, 2019. P. 179–185.
3. Hrybiuk, O., Kant, G.S. CleverCOMSRL: Implementation of an AI Computer-Aided Design System in the Context of the Cognitive Science Paradigm for the Research Training Process. In: Machado, J., et al. *Innovations in Mechatronics Engineering III. icieng 2024. Lecture Notes in Mechanical Engineering*. Springer, Cham. 2024. P. 351-362. https://doi.org/10.1007/978-3-031-61575-7_32.
4. Hrybiuk, O. Improvement of the educational process by the creation of centers for intellectual development and scientific and technical creativity. In: Hamrol, A., Kujawińska, A., Barraza, M.F.S. (eds.) *MANUFACTURING 2019. LNME*, 2019. P. 370–382. Springer, Cham. https://doi.org/10.1007/978-3-030-18789-7_31.

Новіцький В. В. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О
*Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-
технічний університет імені академіка Юрія Бугая» (Київ)*

РОЗРОБЛЕННЯ ГРИ «БАСКЕТБОЛ» ІЗ ВИКОРИСТАННЯМ МАТЕМАТИЧНИХ ОСНОВ ТЕХНОЛОГІЇ ОБ'ЄКТНО ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ

Жанр спортивних ігор є одним із найбільш затребуваних у гейм-індустрії. Зокрема, симулятори баскетболу вимагають точного моделювання траєкторій, швидкості реакції та логіки поведінки суперника. Розробка такої системи в межах курсової роботи дозволяє глибоко дослідити взаємодію фізичних об'єктів у віртуальному просторі [1].

Мета дослідження полягає у проєктуванні та розробленні ігрової системи “Баскетбол” з використанням технології об'єктно орієнтованого програмування.

Завдання: здійснення аналізу предметної області, добір існуючих технологій і засобів для здійснення проєктування та програмування ігрової системи; розробка клієнт-серверної архітектури; створення інтуїтивного та зручного інтерфейсу користувача; реалізація функціоналу ігрового процесу, включаючи моделювання кидків, руху гравців, підрахунку очок та логіки матчів; створення системи керування ігровим контентом [2]; тестування і моніторинг роботи модулів взаємодії; реалізація механізмів ідентифікації гравців і збереження прогресу; забезпечення коректної клієнтської взаємодії; здійснення тестування та верифікації модулів інформаційної системи. Для розробки обрано мову Python та бібліотеку Pygame.

У рамках дослідження здійснено проєктування системи на основі ієрархії класів [3]. Центральним елементом є базовий клас `GameObject`, який забезпечує методи малювання та завантаження текстур для всіх сутностей. Основні файли та їх призначення: `main.py` та `game.py`: Точка входу та основний ігровий цикл. Керують зміною станів гри (меню, ігровий процес, фінал). `game_manager.py`:

«Мозок» гри. Відповідає за підрахунок очок, контроль часу (60 секунд за замовчуванням) та збереження статистики. `game_object.py`: Базовий клас для всіх графічних об'єктів, що реалізує спільні методи малювання та позиціонування [4] `ball.py` та `ball_manager.py`: Реалізація фізики м'яча та керування пулом доступних м'ячів (6 штук). `bot.py`: Клас інтелектуального опонента, що використовує математичні алгоритми для розрахунку траєкторії кидка.

Фізична модель у грі реалізована за допомогою методів чисельного інтегрування (рис. 1). Основним класом є `Ball` (файл `ball.py`), який обробляє гравітацію та колізії.

Фізичні параметри: гравітація: 800.0 пікселів/с². Еластичність: коефіцієнт 0.7 (Bouncing), що дозволяє м'ячу реалістично відскакувати від підлоги та стін (рис. 2). Бот імітує поведінку реального гравця: аналіз: бот визначає поточну позицію кошика, що рухається (Moving Basket). Розрахунок: вираховується необхідна сила та кут кидка для влучання. Похибка: для реалістичності додано параметр `ACCURACY_VARIATION` (5%), завдяки чому бот іноді промахується. Затримка: бот «думає» від 2 до 4 секунд перед кидком, щоб гравець мав шанс на перемогу.

```
114
115     # Apply gravity
116     self.velocity_y += GRAVITY * dt
117
118     # Calculate new position
119     new_x = self.center_x + self.velocity_x * dt
120     new_y = self.center_y + self.velocity_y * dt
121
122     # Left wall bounce
123     if new_x - self.radius <= 0:
124         new_x = self.radius
125         self.velocity_x = abs(self.velocity_x) * BOUNCE_DAMPING
126     # Right wall bounce
127     elif new_x + self.radius >= SCREEN_WIDTH:
128         new_x = SCREEN_WIDTH - self.radius
129         self.velocity_x = -abs(self.velocity_x) * BOUNCE_DAMPING
130
131     # Ceiling bounce
132     if new_y - self.radius <= 0:
133         new_y = self.radius
134         self.velocity_y = abs(self.velocity_y) * BOUNCE_DAMPING
135
136     # Floor bounce
137     elif new_y + self.radius >= SCREEN_HEIGHT:
138         new_y = SCREEN_HEIGHT - self.radius
139         self.velocity_y = -abs(self.velocity_y) * BOUNCE_DAMPING
140         self.velocity_x *= BOUNCE_DAMPING # Also slow down horizontal movement on floor
141
142     # Reset if ball has almost stopped bouncing
143     if abs(self.velocity_y) < 50:
144         self.reset()
```

Рис. 1. Реалізація фізичного рушія
Джерело: опрацювання власне

```

def _get_random_delay(self) -> float:
    """Get random delay between throws"""
    return random.uniform(BOT_MIN_THROW_DELAY, BOT_MAX_THROW_DELAY)

def _calculate_trajectory(self, ball: "Ball", basket: "Basket") -> tuple[float, float]:
    """
    Calculate exact velocity needed to hit the basket.
    Uses physics to predict basket position and compute required trajectory.
    Returns (velocity_x, velocity_y).
    """
    # Ball starting position
    ball_x = ball.center_x
    ball_y = ball.center_y

    # Target position (basket rim center)
    target_y = basket.y + 15 # Aim slightly into the basket

    # Basket movement parameters
    basket_center_x = basket.x + basket.width / 2
    basket_speed = basket.move_speed * basket.direction

```

Рис. 2. Алгоритм прогнозування траєкторії польоту

Джерело: опрацювання власне

Бот розраховує вектор сили на основі відстані до рухомої цілі (рис. 3).

Користувач керує м'ячем за допомогою миші. Модуль `aiming_arrow.py` динамічно малює стрілку, колір якої змінюється від зеленого до червоного залежно від сили натягу.



Рис. 3. Візуальний інтерфейс влучення

Джерело: опрацювання власне

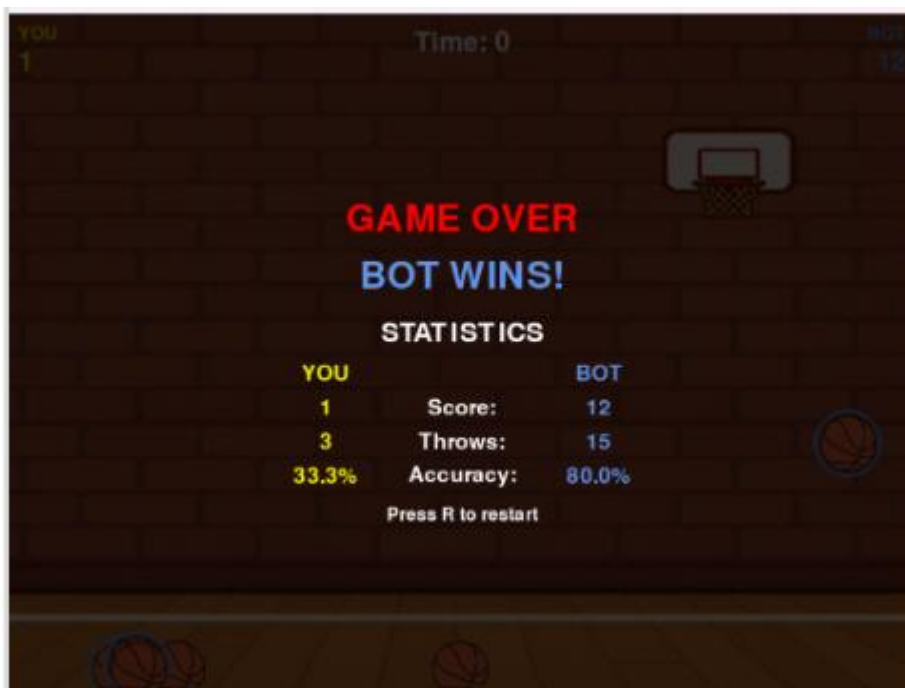


Рис. 4. Вікно підсумкової статистики
Джерело: опрацювання власне

Колірна індикація дозволяє гравцеві інтуїтивно оцінювати силу кидка [5]. В процесі тестування було перевірено механіку нарахування очок (рис. 4). У `game_manager.py` реалізовано логіку, де влучання зараховується лише при падінні м'яча зверху вниз через кільце.

Система вираховує точність (Accuracy) гравця та бота у відсотках. Ключові сценарії перевірки: влучання в кошик: перевірка перетину м'ячем `score_zone`. Умова влучання: м'яч повинен рухатися зверху вниз та перебувати в межах кільця. Завершення сесії: після закінчення 60 секунд гри (`GAME_DURATION`) програма повинна заблокувати керування та вивести фінальну статистику.

У рамках дослідження озроблено програмний продукт «Basketball Arcade Game», який демонструє принципи взаємодії людини та штучного інтелекту в ігровому середовищі. Застосовано об'єктно-орієнтований підхід (ООП), що дозволило створити гнучку структуру класів (`Ball`, `Basket`, `Bot`, `GameManager`). Це забезпечує легку модифікацію та розширюваність коду. Фізичне моделювання [6]: реалізовано реалістичну модель руху м'яча, що

враховує гравітацію, вектори швидкості та коефіцієнт пружності при зіткненні з поверхнями.

Розроблено інтелектуального агента (бота), який на основі фізичних розрахунків прогнозує траєкторію кидка та забезпечує змагальний елемент гри. Інтерфейс користувача: Створено систему динамічного моделювання із візуальною індикацією сили кидка, що робить ігровий процес інтуїтивно зрозумілим [7]. Проведене тестування підтвердило стабільність роботи програми та коректність виконання всіх ігрових алгоритмів. Розроблену систему можна використовувати як базі для створення складніших спортивних симуляторів або як навчальний приклад застосування фізики.

Анотація. У рамках дослідження здійснено проектування і практичну реалізацію тривимірної ігрової системи типу «Баскетбол». На початковому етапі було визначено вимоги до апаратного та програмного забезпечення, було детально розглянуто логіку функціонування ігрової системи. Зокрема, описано головний ігровий цикл, алгоритм руху та відбиття м'яча, а також механізм мережевої синхронізації. У рамках реалізації програмної частини було розроблено основні класи системи, зокрема клас керування гравцем та клас м'яча. Додатково було сформовано архітектуру програмної системи з розподілом на функціональні рівні, що забезпечує модульність і гнучкість розробки. Побудовано діаграми, які відображають структуру та взаємодію компонентів, що дозволило формалізувати внутрішню організацію програмного продукту. У рамках дослідження реалізовано користувацький інтерфейс, що забезпечує зручну взаємодію гравця з ігровим середовищем. Окрему увагу приділено тестуванню програмної системи, у результаті чого підтверджено коректність роботи основних функціональних компонентів, стабільність продуктивності та відповідність нефункціональним вимогам. Отримані результати свідчать про надійність і ефективність реалізованого програмного продукту.

Ключові слова: баскетбол, дослідницьке навчання, варіативні моделі, математичне моделювання, об'єктно орієнтованого програмування, комп'ютерне моделювання, варіативні моделі, ігрова система, AI, КОМСДН, Python.

Література

1. Hrybiuk O. Experience in Implementing Computer-Oriented Methodological Systems of Natural Science and Mathematics Research Learning in Ukrainian Educational Institutions. In: Machado J., Soares F. (eds) *Innovations in Mechatronics Engineering. Lecture Notes in Mechanical Engineering*, 2022. P. 55-68. Springer, Cham Online.
2. Hrybiuk O. Problems of expert evaluation in terms of the use of variative models of a computer-oriented learning environment of mathematical and natural science disciplines in schools,

[w:] *Zeszyty Naukowe Politechniki Poznańskiej. Seria: Organizacja i Zarządzanie*, Zeszyt Nr 79, Poznań: Wydawnictwo Politechniki Poznańskiej (WPP), 2019. P. 101-119.

3. Гриб'юк О.О. Імерсивні технології в освіті: особливості когнітивного розвитку дитини у віртуальному середовищі в процесі дослідницького навчання. *Сучасні інформаційні технології та інноваційні методики навчання в підготовці фахівців: методологія, теорія, досвід, проблеми: збірник наукових праць*. Київ-Вінниця: ТОВ Фірма «Планер», 2020.

4. Гриб'юк О.О. Комп'ютерне моделювання та робототехніка в навчально-виховному процесі сучасного навчального закладу. // *Сьома міжнародна науково-практична конференція FOSS Lviv-2017: Збірник наукових праць*, 27-30 квітня 2017 року, м. Львів. С. 38-43. ISBN 978-966-2598-86-5.

5. Hrybiuk Olena. Engineering in Educational Institutions: Standards for Arduino Robots as an Opportunity to Occupy an Important Niche in Educational Robotics in the Context of Manufacturing 4.0, in: *Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*. Volume 27-32, 2020. P. 770-785.

6. Hrybiuk O., Vedishcheva O. Experimental Teaching of Robotics in the Context of Manufacturing 4.0: Effective Use of Modules of the Model Program of Environmental Research Teaching in the Working Process of the Centers "Clever". In: , et al. *Innovations in Mechatronics Engineering II. icieng 2022. Lecture Notes in Mechanical Engineering*. Springer, Cham. 2022.

7. Hrybiuk O. Improvement of the Educational Process by the Creation of Centers for Intellectual Development and Scientific and Technical Creativity. In: Hamrol A., Kujawińska A., Barraza M. (eds). *Advances in Manufacturing II. MANUFACTURING 2019. Lecture Notes in Mechanical Engineering*, 2019. P. 370-382. Springer, Cham Online.

УДК 004.942:[327

Онофрійчук В.О. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О
Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-технічний університет імені академіка Юрія Бугая» (Київ)

МАТЕМАТИЧНЕ ПІДГРУНТЯ РОЗРОБЛЕННЯ ІГРОВОЇ СИСТЕМИ НА ПЛАТФОРМІ UNITY З ВИКОРИСТАННЯМ ГЕЙМДИЗАЙН ОРІЄНТОВАНИХ ПІДХОДІВ

Актуальність дослідження. Сучасне розроблення ігрових застосунків вимагає створення гнучких та масштабованих програмних архітектур, здатних забезпечувати синхронізацію дій гравця та автономних агентів (NPC) у реальному часі. Особливої уваги потребує реалізація механік у замкнених просторах з обмеженими ресурсами, де виникають конфлікти навігації та доступу до об'єктів. Використання стандартних жорстких скриптових послідовностей призводить до ускладнення коду та зниження продуктивності. Тому актуальним є впровадження патернів об'єктно-орієнтованого

програмування та подієво-орієнтованої архітектури для створення модульних ігрових систем на платформі *Unity* [1]. Таким чином, розробка інформаційної системи парсингу даних є актуальною задачею, яка дозволяє автоматизувати процеси збору, обробки та надання структурованої інформації користувачам у зручному форматі.

Мета дослідження полягає у проектуванні та розробленні ігрової системи на платформі *Unity* з використанням геймдизайн орієнтованих підходів. Для досягнення поставленої мети було розроблено архітектуру, що базується на слабкій зв'язності компонентів (*decoupling*). Ключовим елементом системи взаємодії став інтерфейс *IInteractable* [2]. Він необхідний для уніфікації логіки роботи з різнорідними об'єктами (столи, плити, контейнери) та реалізації поліморфної поведінки: контролер гравця або NPC не залежить від конкретної реалізації об'єкта, а звертається до нього через абстракцію. Такий підхід допоміг легко розширювати систему новими типами інтерактивних об'єктів без зміни коду контролерів персонажів. Для візуалізації архітектурних зв'язків між менеджерами, контролерами та об'єктами даних було побудовано діаграму класів (рис. 1). Важливою складовою стала реалізація системи переміщення предметів між об'єктами. Інтерфейс *IKitchenObjectParent* дозволяє динамічно змінювати ієрархію об'єктів на сцені (*Transform Parenting*) та керувати їх фізичним станом (кінематика/гравітація) при передачі від гравця до столу і навпаки.

Реалізація поведінки NPC. Для моделювання поведінки неігрових персонажів (*NPC*) було розроблено архітектуру на основі патерну *State* (Стан) [3]. Центральний контролер агента (*NPCController*) не містить розгалуженої логіки прийняття рішень, а делегує виконання дій окремим класам-станам, що наслідуються від абстрактного класу *BaseState*. Управління переходами між станами здійснює компонент *StateMachine*.

Логіка поведінки агентів формалізована у вигляді орієнтованого графу станів, що містить: 1. *IdleState* - аналіз навколишнього середовища та пошук

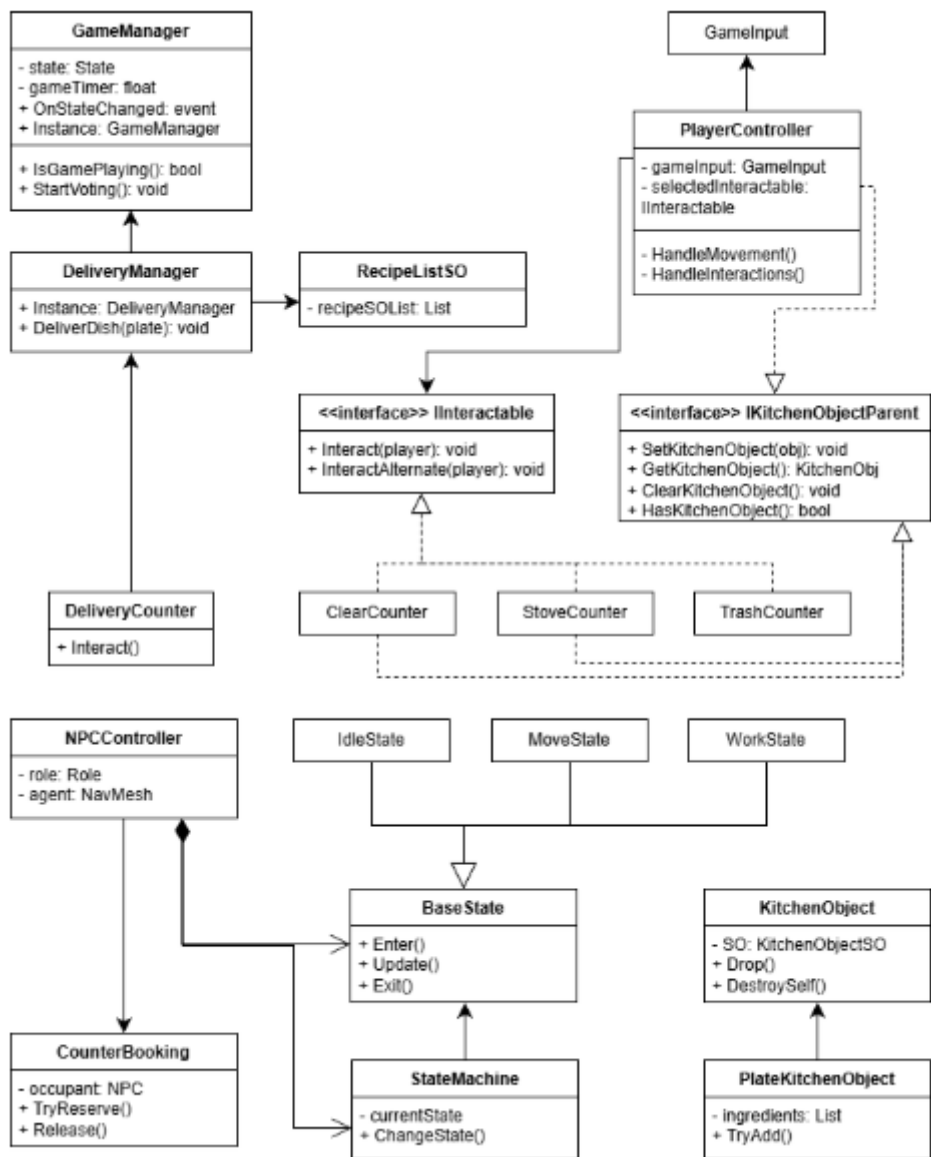


Рис. 1. Фрагмент ескізного проєкту ігрової системи
Джерело: опрацювання власне

доступних завдань. 2. *MoveToState* - переміщення до цілі з використанням навігаційної сітки (NavMesh). 3. *WorkState* - виконання контекстно-залежної дії (обробка продукту або саботаж) [4].

Вирішення конфлікту між NPC. Критичною проблемою мультиагентних систем у замкненому просторі є конкуренція за ресурси (робочі поверхні). Стандартні засоби навігації Unity вирішують проблему фізичних зіткнень, але не запобігають логічним конфліктам, коли декілька агентів одночасно обирають одну ціль. Для вирішення цієї проблеми розроблено та імплементовано алгоритм «Попереднього бронювання».

Кожен інтерактивний об'єкт містить компонент CounterBooking, який зберігає посилання на поточного користувача. У стані IdleState агент виконує перебір доступних цілей та намагається зарезервувати вільний стіл перед початком руху [5].

Програмна реалізація рольової моделі. Система підтримує динамічну рольову модель. Використовуючи поліморфізм, стан *WorkState* змінює свою поведінку залежно від ролі агента [6]. У методі *Enter()* відбувається ідентифікація ролі, налаштування тривалості роботи та перевірка типу об'єкта для можливого саботажу. Метод *Update()* відповідає за таймер та коректне звільнення ресурсу через *CounterBooking*. Візуалізація роботи механіки саботажу в симуляційному середовищі наведена на рис. 2.

Організація роботи з даними. Для керування конфігураційними даними (параметри рецептів, час обробки, властивості предметів) використано патерн Data Transfer Object (DTO), реалізований засобами Unity ScriptableObjects [7]. Це дозволяє відокремити статичні дані від програмної логіки. Завдяки атрибуту *CreateAssetMenu*, геймдизайнери можуть створювати нові ігрові об'єкти безпосередньо в редакторі, не втручаючись у програмний код.

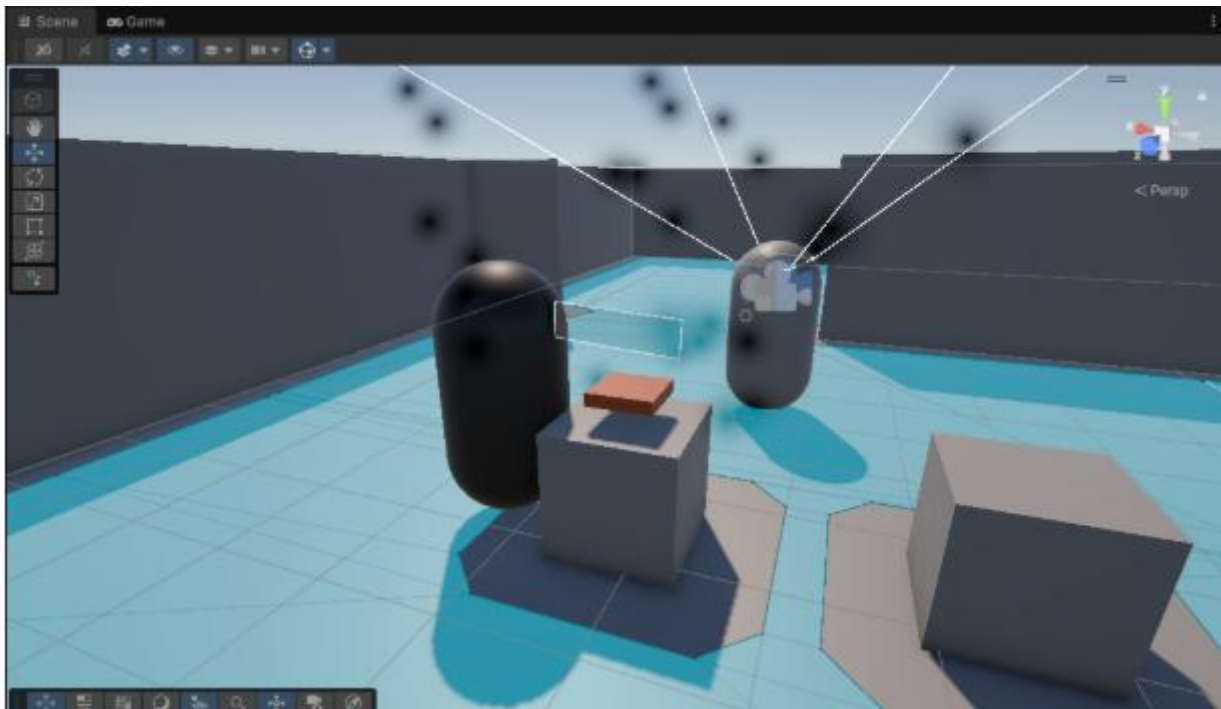


Рис. 2. Візуалізація роботи агентів і механіки саботажу в середовищі Unity

Джерело: опрацювання власне

Анотація. У рамках дослідження здійснено проєктування та розроблення базової архітектури ігрової системи, що включає підсистему взаємодії, штучний інтелект на основі скінченних автоматів (FSM) та механізм координації агентів для уникнення колізій при доступі до ресурсів. У результаті роботи спроектовано та програмно реалізовано архітектуру ігрової системи на платформі Unity з використанням геймдизайн орієнтованих підходів. Продемонстровано застосування FSM для створення модульної й розширюваної логіки ігрових персонажів. Розроблено алгоритм попереднього бронювання ресурсів, який повністю усунув проблему логічних колізій у середовищі. Інформаційна система підтримує динамічну зміну ролей у реальному часі та готова до подальшого масштабування.

Ключові слова: CleverCOMSRL, дослідницьке навчання, варіативні моделі, математика, комп'ютерне моделювання, КОМСДН, ігрова система, геймдизайн орієнтований підхід, AI, Unity.

Література

1. Hrybiuk, O. Improvement of the educational process by the creation of centers for intellectual development and scientific and technical creativity. In: Hamrol, A., Kujawińska, A., Barraza, M.F.S. (eds.) *MANUFACTURING 2019*. LNME, 2019. P. 370–382. Springer, Cham. https://doi.org/10.1007/978-3-030-18789-7_31.
2. Hrybiuk, O., Kant, G.S. CleverCOMSRL: Implementation of an AI Computer-Aided Design System in the Context of the Cognitive Science Paradigm for the Research Training Process. In: Machado, J., et al. *Innovations in Mechatronics Engineering III. icieng 2024. Lecture Notes in Mechanical Engineering*. Springer, Cham. 2024. P. 351-362. https://doi.org/10.1007/978-3-031-61575-7_32.
3. Hrybiuk, O.O. Research learning of the natural science and mathematics cycle using computer-oriented methodological systems. Monograph. Drahomanov NPU, Kyiv, 2019. P. 277–325.
4. Khang A., Abdullayev V., Hrybiuk O., & Shukla A.K. (Eds.). *Computer Vision and AI-Integrated IoT Technologies in the Medical Ecosystem* (1st ed.). CRC Press. 2024. <https://doi.org/10.1201/9781003429609>.
5. 1. Refactoring.Guru. State Design Pattern. URL: <https://refactoring.guru/design-patterns/state> (дата звернення: 17.11.2025).
6. 2. Unity Technologies. NavMesh Agent Scripting Reference. Unity Documentation. URL: <https://docs.unity3d.com/6000.2/Documentation/ScriptReference/AI.NavMeshAgent.html> (дата звернення: 24.11.2025).
7. 3. Unity Technologies. ScriptableObject Architecture. Unity Documentation. URL: <https://docs.unity3d.com/Manual/class-ScriptableObject.html> (дата звернення: 01.12.2025).

Пилипенко Д. Д. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О
*Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-
технічний університет імені академіка Юрія Бугая» (Київ)*

МАТЕМАТИЧНІ ПРИНЦИПИ ПРОЕКТУВАННЯ КАЛЬКУЛЯТОРА ТРИГОНОМЕТРИЧНИХ ФУНКЦІЙ «SYNTAXMIND»

У дослідженні реалізована Clean Architecture [1] з розподілом на шари, де бізнес-логіка ізольована від реалізації Infrastructure, Web, що дозволяє додавати зміни у будь-який шару без впливу на ядро системи. Core – це шар бізнес-логіки, який складається з інтерфейсів бізнес-логіки. IExpressionEvaluator – основні методи визначення контракту: ToRpn – перетворення інфіксного виразу у постфіксну [2]; CalculateRpn – обчислення виразу у формі RPN. Цей рівень містить всю логіку домену без залежностей від БД, .NET Framework або UI. Шар Infrastructure (рис. 1) містить реалізацію класу ExpressionEvaluator - ядро системи, що реалізує інтерфейс IExpressionEvaluator. Реалізовано алгоритми перетворення та обчислення RPN [3].

Web – рівень представлення ASP.NET Core MVC [4]. Архітектура: контролери: HomeController; Index.cshtml – головна сторінка калькулятора, Graph.cshtml – візуалізація графіка, Manual.cshtml – довідка користувача. CalculatorViewModel передає дані між контролером і представленнями. Працює через абстракцію IExpressionEvaluator – не знає про конкретний ExpressionEvaluator. Unit test (рис. 2) для бізнес-логіки через інтерфейс IExpressionEvaluator [29], [30], який дозволяє ізольоване тестування без залежності від Web або Infrastructure.

```

namespace KR.Infrastructure
{
    Ссылка: 3
    public class ExpressionEvaluator : IExpressionEvaluator
    {
        private readonly string[] operators =
            string[] {
                "+", "-", "*", "/", "^", "sqrt",
                "sin", "cos", "tan", "ctg", "sec", "csc",
                "arcsin", "arccos", "arctan", "arcctg", "arcsec", "arccsc",
                "deg", "abs"
            };

        Ссылка: 39
        public double CalculateRpn(string rpn, int digits = 4)
        {
            if (string.IsNullOrEmpty(rpn))
                throw new ArgumentException(message: "Вхідний рядок не може бути порожнім");

            string[] tokens = rpn.Split(new[] { ' ' }, StringSplitOptions.RemoveEmptyEntries);
            Stack<double> stack = new Stack<double>();

```

Рис. 1. Sharp Infrastructure
Джерело: опрацювання власне

Web – рівень представлення ASP.NET Core MVC [4]. Архітектура: контролери: HomeController; Index.cshtml – головна сторінка калькулятора, Graph.cshtml – візуалізація графіка, Manual.cshtml – довідка користувача. CalculatorViewModel передає дані між контролером і представленнями. Працює через абстракцію IExpressionEvaluator – не знає про конкретний ExpressionEvaluator. Unit test (рис. 2) для бізнес-логіки через інтерфейс IExpressionEvaluator [29], [30], який дозволяє ізольоване тестування без залежності від Web або Infrastructure.

```

> using ...
namespace KRTest
{
    Ссылка: 0
    public class ExpressionEvaluatorOperatorsTests
    {
        private readonly KR.Core.IExpressionEvaluator _evaluator = new ExpressionEvaluator();

        [Fact]
        ● | Ссылка: 0
        public void Add_Works()
        {
            Assert.Equal(expected: 5, _evaluator.CalculateRpn(rpn: "2 3 +"));
        }

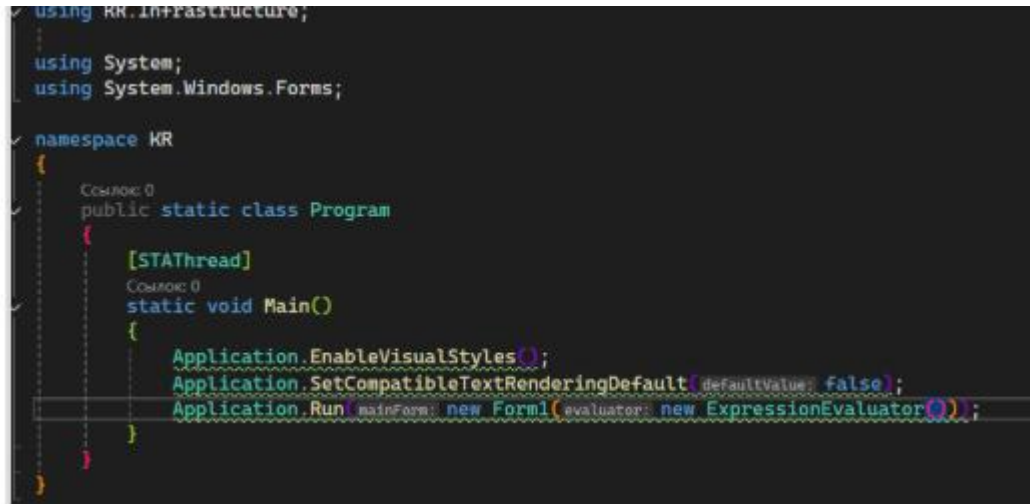
        [Fact]
        ● | Ссылка: 0
        public void Subtract_Works()
        {
            Assert.Equal(expected: 2, _evaluator.CalculateRpn(rpn: "5 3 -"));
        }

        [Fact]
        ● | Ссылка: 0
        public void Multiply_Works()
        {
            Assert.Equal(expected: 6, _evaluator.CalculateRpn(rpn: "2 3 *"));
        }
    }
}

```

Рис. 2. Unit test
Джерело: опрацювання власне

WinForms (рис. 3) [5] використовує IExpressionEvaluator; у поточній реалізації Form1 отримує певний ExpressionEvaluator під час створення (новий Form1(new ExpressionEvaluator())).



```
using KR.Infrastructure;
using System;
using System.Windows.Forms;

namespace KR
{
    Ссылка: 0
    public static class Program
    {
        Ссылка: 0
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(mainForm: new Form1(evaluator: new ExpressionEvaluator()));
        }
    }
}
```

Рис. 3. WinForms
Джерело: опрацювання власне

Було сформульовані функціональні та нефункціональні вимоги (див нижче для «Syntaxmind» [6]. Вимоги розроблено на основі аналізу потреб користувача та специфіки роботи з тригонометричними обчисленнями. Для функціонального моделювання процесу обробки для «SyntaxMind» застосовано методологію IDEF0. Головний процес A0 – обробка тригонометричного виразу реалізує ЖЦ від введення математичного виразу користувачем до отримання числового результату. Процес реалізований через взаємодію інтерфейсу Form1 та модуля обчислень ExpressionEvaluator. Головний процес A0 декомпонується на 5 взаємопов'язаних підпроцесів. Детальний опис входів, виходів, управління та механізмів.

Токенізація через Regex (рис. 4) для 15 унарних (sin, cos, tan, ctg, sec, csc, arcsin, arccos, arctan, arcctg, arcsec, arccsc, sqrt, deg, abs) та 5 бінарних (+, -, *, /, ^) операторів, який розбиває вираз на окремі токени. new Regex використовує стрілочну функцію => та патерн-матчинг is ... or

```

Слайд 3
public class ExpressionEvaluator : IExpressionEvaluator
{
    private readonly string[] operators =
    {
        "+", "-", "*", "/", "^", "sqrt",
        "sin", "cos", "tan", "ctg", "sec", "csc",
        "arcsin", "arccos", "arctan", "arcctg", "arcsec", "arccsc",
        "deg", "abs"
    };

    Слайд 39
    public double CalculateRpn(string rpn, int digits = 4)
    {
        if (string.IsNullOrEmpty(rpn))
            throw new ArgumentException("Вхідний рядок не може бути порожнім");

        string[] tokens = rpn.Split(new[] { ' ' }, StringSplitOptions.RemoveEmptyEntries);
        Stack<double> stack = new Stack<double>();

        foreach (var string token in tokens)
        {
            if (double.TryParse(token, System.Globalization.NumberStyles.Any, System.Globalization.Cu
            {
                stack.Push(number);
                continue;
            }

            if (IsUnaryOperator(token))

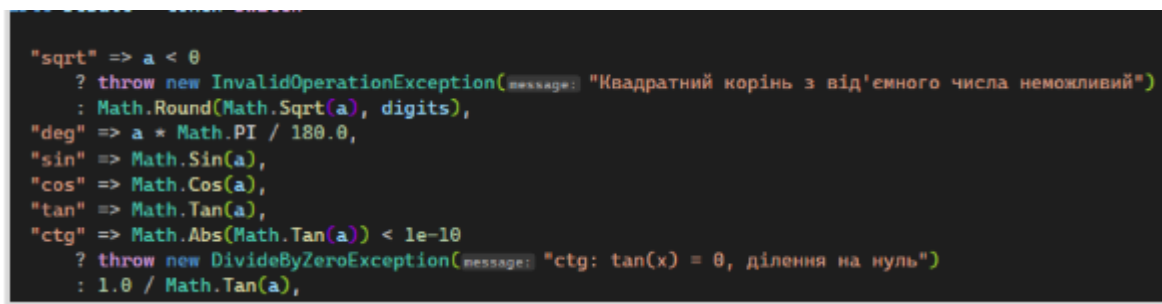
```

Рис. 4. Парсинг даних виразів
Джерело: опрацювання власне

Регулярний вираз $([.,]?\d+)?$ підтримує як крапку $.$, так і кому $,$], обробка від'ємних значень $-\d+$. Підхід: дворівнева обробка - лексичний аналіз + розбивка синтаксису з перетворенням інфіксного у постфіксний запис [7]. Проект, реалізований в ExpressionEvaluator.Rpn і обчислювати RPN в ExpressionEvaluator.CalculateRpn. Підтримувані токени: числа (з $,$ та $.$), унарні функції (sin, cos, deg, sqrt, abs, зворотні функції arcsin), бінарні оператори (+, -, *, /,), дужки. Обробка помилок: винятки (ArgumentException, InvalidOperationException, DivideByZeroException, OverflowException). Перевірка функцій: перевірка аргументів (наприклад, arcsin/arccos - $|x| \leq 1$), захист від ділення на 0 (ctg/sec/csc), обробка double.MinValue з abs. Місце і формат введення: коми введення замінюються точками Replace(",", "."), парсинг чисел за допомогою CultureInfo.InvariantCulture.

Метод ToRpn() розбиває рядок на токени за допомогою регулярного виразу [5], розставляє пріоритети операторів (тригонометричні функції мають найвищий пріоритет 4, степінь – 3, множення/ділення – 2, додавання/віднімання – 1), обробляє дужки і видає RPN вираз, яка передається в CalculateRpn(). RPN метод використовує стек [3]: числа додають до списку, а при зустрічі унарного оператора (sin, cos, tan, ctg, sec, csc, arcsin, arccos, arctan, arcctg, arcsec, arccsc,

sqrt, deg, abs) з вершини стеку знімається один операнд, застосовується відповідна функція з класу Math і результат повертається назад у стек; для бінарних операторів знімаються два операнди. Тригонометричні функції обчислюються напряму через Math.Sin, Math.Cos, Math.Tan, але є оператор «deg» (1), тому можна писати sin(30 deg) і отримувати правильний результат без ручного обчислення (рис. 5).



```
"sqrt" => a < 0
    ? throw new InvalidOperationException(message: "Квадратний корінь з від'ємного числа неможливий")
    : Math.Round(Math.Sqrt(a), digits),
"deg" => a * Math.PI / 180.0,
"sin" => Math.Sin(a),
"cos" => Math.Cos(a),
"tan" => Math.Tan(a),
"ctg" => Math.Abs(Math.Tan(a)) < 1e-10
    ? throw new DivideByZeroException(message: "ctg: tan(x) = 0, ділення на нуль")
    : 1.0 / Math.Tan(a),
```

Рис. 5. Тригонометричні функції
Джерело: опрацювання власне

Для перевірки коректності та стійкості застосунку виконано модульне та інтеграційне тестування ключових сценаріїв калькулятора [3]. Модульне тестування (KRTTest) охоплює: базові арифметичні операції (+, -, *, /, ^); унарні функції (sqrt, sin, cos, tan, ctg, sec, csc, abs); обернені тригонометричні функції (arcsin, arccos, arctan, arcctg, arcsec, arccsc) [6]; перетворення градусів у радіани (deg); перевірку виняткових станів: ділення на нуль, некоректні діапазони аргументів, NaN, Infinity, переповнення. Перевірка надійності web-рівня (HomeController) включає: валідацію вхідних даних (порожні вирази, некоректні параметри); безпечну обробку помилок із поверненням зрозумілих повідомлень користувачу; захист POST-запитів через антифальсифікаційний токен; контроль обсягу обчислень для графіків (обмеження кількості точок, пропуск проблемних значень).

За результатами тестування виконано рефакторинг коду. У методі ToRpn() виокремлено приватні методи для розпізнавання токенів і визначення пріоритетів операторів [7]. У методі CalculateRpn() усунуто дублювання логіки обробки унарних функцій: замість switch застосовано словник Dictionary<string, Func<double, double>>, який порівнює назви функцій із відповідними

делегатами Math.*. Повідомлення для винятків: `ArgumentException` і `InvalidOperationException` містить конкретну причину помилки замість узагальненого повідомлення. На підставі аналізу результатів повторного тестування підтверджено коректність змін і підвищення надійності застосунку.

Анотація. У рамках дослідження ґрунтовно розроблено архітектуру та здійснено практичну реалізацію калькулятора тригонометричних функцій «SyntaxMind» відповідно до функціональних і нефункціональних вимог. Реалізовано повний ЖЦ обробки математичного виразу: введення в інфіксній формі, токенизацію, перетворення у формат RPN та обчислення результату. Реалізовано арифметичні, тригонометричні, обернені тригонометричні та математичні функції; є обробка десяткових чисел, округлення та дужки. Додатково реалізовано побудову графіків із перевіркою параметрів. Архітектуру побудовано за принципами Clean Architecture, що розподіляє шар Core, Infrastructure, Web. Ключовим елементом є абстракція `IExpressionEvaluator`, яка ізолює обчислювальну логіку, спрощує тестування та розширення функціональних можливостей. Для підтвердження коректності та стійкості застосунку виконано модульне й інтеграційне тестування, з перевіркою виняткових ситуацій. Реалізовано типізовану обробку помилок, зрозумілі повідомлення для користувача, захист POST-запитів анти-CSRF токеном, підтримку HTTPS/HSTS і надійну роботу з некоректними даними сесії. Результати тестування білого та чорного ящика підтвердили правильність обчислень і стабільність роботи системи.

Ключові слова: об'єктно орієнтоване програмування, дослідницьке навчання, варіативні моделі, математичне моделювання, інформаційна система, калькулятор тригонометричних функцій, комп'ютерне моделювання, КОМСДН, Clean Architecture, AI.

Література

1. Hrybiuk Olena. Engineering in Educational Institutions: Standards for Arduino Robots as an Opportunity to Occupy an Important Niche in Educational Robotics in the Context of Manufacturing 4.0, in: *Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*. Volume 27-32, 2020. P. 770-785.
2. Hrybiuk O. Problems of expert evaluation in terms of the use of variative models of a computer-oriented learning environment of mathematical and natural science disciplines in schools, [w:] *Zeszyty Naukowe Politechniki Poznańskiej. Seria: Organizacja i Zarządzanie*, Zeszyt Nr 79, Poznań: Wydawnictwo Politechniki Poznaskiej (WPP), 2019. P. 101-119.
3. Гриб'юк О.О. Дослідницьке навчання учнів предметів природничо-математичного циклу з використанням комп'ютерно орієнтованих методичних систем / О. О. Гриб'юк. Монографія. Київ: НПУ імені М. П. Драгоманова, 2019.
4. Hrybiuk O., Vedishcheva O. Experimental Teaching of Robotics in the Context of Manufacturing 4.0: Effective Use of Modules of the Model Program of Environmental Research Teaching in the Working Process of the Centers “Clever”. In: , et al. *Innovations in Mechatronics Engineering II. icieng 2022. Lecture Notes in Mechanical Engineering*. Springer, Cham. 2022.
5. Hrybiuk O. Experience in Implementing Computer-Oriented Methodological Systems of Natural Science and Mathematics Research Learning in Ukrainian Educational

Institutions. In: Machado J., Soares F. (eds) *Innovations in Mechatronics Engineering. Lecture Notes in Mechanical Engineering*, 2022. P. 55-68. Springer, Cham Online.

6. Гриб'юк О.О. Комп'ютерне моделювання та робототехніка в навчально-виховному процесі сучасного навчального закладу. / Гриб'юк О.О. // *Сьома міжнародна науково-практична конференція FOSS Lviv-2017: Збірник наукових праць*, 27-30 квітня 2017 року, м. Львів. С. 38-43. ISBN 978-966-2598-86-5.

7. Hrybiuk O. Improvement of the Educational Process by the Creation of Centers for Intellectual Development and Scientific and Technical Creativity. In: Hamrol A., Kujawińska A., Barraza M. (eds) *Advances in Manufacturing II. MANUFACTURING 2019. Lecture Notes in Mechanical Engineering*, 2019. P. 370-382. Springer, Cham Online.

УДК 004.942:[327

Посмаренко О. С. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О
Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-технічний університет імені академіка Юрія Бугая» (Київ)

РОЗРОБЛЕННЯ ІГРОВОЇ СИСТЕМИ ЖАНРУ “АРКАНОЇД” З ВИКОРИСТАННЯМ МАТЕМАТИЧНИХ ОСНОВ ТЕХНОЛОГІЇ ОБ’ЄКТНО ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ

Проблематика розроблення тривимірної версії ігрової системи жанру «Арканойд» полягає у необхідності вирішення ряду спеціалізованих технічних задач: забезпечення коректної фізичної моделі взаємодії об’єктів у 3D-просторі, побудова оптимізованої графічної підсистеми, а також розроблення надійного механізму мультиплеєрної синхронізації з мінімізацією латентності [1]. Додатковим аспектом є кросплатформеність, що вимагає створення уніфікованої архітектури, сумісної з різними мобільними й десктопними середовищами (*Windows, Android, iOS, macOS*).

Актуальність роботи зумовлюється необхідністю поєднання класичної ігрової концепції із сучасними технологічними вимогами, що забезпечує конкурентоспроможність продукту та розширює спектр його застосування у розважальній індустрії. Розроблення проєкту на базі рушія *Unity* є доцільним рішенням з огляду на можливості використання *PhysX* для моделювання колізій, архітектури *Scriptable* для параметризації об’єктів і відповідних мережевих фреймворків у режимі реального часу [2].

Мета дослідження полягає у розробленні ігрового застосунку жанру «Арканойд» із підтримкою мультиплеєру та адаптивною системою управління у контексті виваженого використання технології об'єктно орієнтованого програмування. Визначення системних вимог є критично важливим етапом проектування ігрової системи, оскільки це безпосередньо впливає на вибір технологічного стеку, методів оптимізації графіки та архітектуру мережевої взаємодії [3]. Оскільки розробка ведеться на базі ігрового рушія Unity, вимоги поділяються на дві категорії: вимоги до середовища розробки Editor та вимоги до цільової платформи користувача Build. Для кінцевого продукту (3D Arkanoid) вимоги є значно нижчими, ніж для середовища розробки, проте вони мають враховувати специфіку тривимірної графіки та потребу в стабільному мережевому з'єднанні для мультиплеєра [4]. Вимоги до ігрового застосунку прийнято розділяти на мінімальні (забезпечують запуск гри при 30 FPS на низьких налаштуваннях) та рекомендовані (забезпечують стабільні 60+ FPS на високих налаштуваннях графіки).

Діаграма архітектури програмної системи відображає загальну структуру розробленого ігрового застосунку «Arkanoid» та демонструє взаємодію між його основними компонентами (рис. 1): *Gameplay Layer*, *Presentation Layer*, *Cross-Cutting* і *External/Platform*. *Gameplay Layer* є центральним елементом системи та відповідає за реалізацію всієї ігрової логіки. До нього входять такі компоненти: *Player Controller* - обробляє введення користувача (клавіші A/D) та ініціює початок гри; *Ball System* - керує рухом м'яча, обробляє зіткнення та передає події іншим підсистемам; *Block System* - відповідає за логіку руйнування блоків та обробку отриманого пошкодження; *Bonus System* - створює бонуси після знищення блоків та застосовує їх ефекти до гравця. *Presentation Layer* відповідає за взаємодію з користувачем та відображення інформації на екрані. До нього входять: *Main Menu UI* - головне меню гри; *In-Game UI* - інтерфейс під час гри (рахунок, повідомлення); *MenuManager* - керує переходами між сценами, обробляє UI-події та відповідає за логіку меню.

До Cross-Cutting входять компоненти, які використовуються в різних частинах системи: Scene Lifecycle (DontDestroyOnLoad) - забезпечує збереження об'єктів між сценами; Visual Environment (LightLampAnimation) - відповідає за візуальні ефекти, зокрема зміну освітлення. External / Platform включає зовнішні інструменти та механізми рушія Unity: Unity Engine - забезпечує фізику, обробку введення, управління сценами та створення/знищення об'єктів; Scene Assets - містить ресурси сцени (префаби, теги, об'єкти); LeanTween - бібліотека для реалізації анімацій (UI, тексту, переходів).

Рис. 1. Діаграма архітектури ігрової системи
Джерело: опрацювання власне

Логіка функціонування тривимірної ігрової системи «Арканойд» базується на подійно-орієнтованій моделі (Event-Driven Design) та життєвому циклі об'єктів Unity. Основна складність алгоритмізації полягає у поєднанні детермінованої фізики (рух м'яча) та недетермінованого введення користувача, особливо в умовах мережевої синхронізації [5]. *Алгоритм головного ігрового циклу Game Loop.* Головний цикл програми відповідає за послідовне оновлення станів ігрових об'єктів. В Unity цей процес розділений на три основні фази: обробка введення (Update), розрахунок фізики (FixedUpdate) та рендеринг (LateUpdate). *Ініціалізація:* Завантаження сцени, створення об'єктів (Pool Object для блоків), налаштування параметрів гравця; *Опитування введення:* Зчитування положення курсора або натискання клавіш (A/D) для визначення

вектора руху платформи [6]; *Оновлення позицій м'яча* та перевірка колізій (зіткнень); *Оновлення стану*: Перевірка умов перемоги (всі блоки знищено) або поразки (м'яч за межами поля); *Візуалізація*: Відмальовування кадру на основі розрахованих координат.

Алгоритм руху та відбиття м'яча. Рух м'яча в 3D-просторі не є випадковим. Для забезпечення ігрового балансу використовується алгоритм збереження кінетичної енергії, де модуль швидкості залишається сталим, а змінюється лише вектор напрямку. Коли м'яч зіштовхується з об'єктом, алгоритм виконує наступні кроки: Визначається точка зіткнення та нормаль n до поверхні у цій точці. У разі контакту обчислюється вектор відбиття згідно з формулою (1); якщо зіткнення відбулося з платформою гравця, кут відбиття додатково коригується залежно від відстані між точкою дотику та центром платформи. Це надає можливість керувати напрямком руху м'яча [7]. *Алгоритм мережевої синхронізації Multiplayer Sync.* У мультиплеєрному режимі використовується модель «Клієнт - Авторитетний сервер». Основне завдання - синхронізувати положення м'яча, який рухається з великою швидкістю, щоб уникнути розсинхронізації.

Реалізація класу керування гравцем. Клас `Player` відповідає за реалізацію керування ігровою платформою, взаємодію з фізичною системою `Unity`, запуск ігрового процесу та обробку бонусів. Він є одним із ключових компонентів ігрової логіки, оскільки забезпечує безпосередню взаємодію користувача з ігровим середовищем. Метод `Update()` відповідає за обробку натискань клавіш: при натисканні клавіші `A` платформа рухається вліво; при натисканні клавіші `D` - вправо; за відсутності введення рух припиняється. Рух реалізовано шляхом безпосереднього встановлення швидкості `Rigidbody` через властивість `linearVelocity` (рис. 2).

```
private void Update()
{
    if (Input.GetKey(KeyCode.A)) _rb.linearVelocity = new Vector3(-movement_speed, 0, 0);
    else if (Input.GetKey(KeyCode.D)) _rb.linearVelocity = new Vector3(movement_speed, 0, 0);
    else _rb.linearVelocity = Vector3.zero;
}
```

Рис. 2. Метод `Update()` у класі `Player`

Джерело: опрацювання власне

Метод *OnTriggerEnter(Collider other)* реалізує обробку зіткнень із бонусами. При попаданні платформи в тригер з тегом Bonus (рис. 3). Це дозволяє реалізувати розширювану систему бонусів, де кожен бонус інкапсулює власну логіку впливу на гравця (наприклад, збільшення платформи, зміна швидкості, додаткові життя тощо).

```
private void OnTriggerEnter(Collider other)
{
    if (other.gameObject.CompareTag("Bonus"))
    {
        Bonus bonus = other.gameObject.GetComponent<Bonus>();
        bonus.TakeEffect(this);
    }
}
```

Рис. 3. Метод *OnTriggerEnter(Collider other)* у класі *Player*
Джерело: опрацювання власне

Реалізація класу м'яча Ball. Клас *Ball* реалізує один із центральних елементів ігрової системи - поведінку м'яча, його фізичний рух, відбиття від поверхонь, підтримання сталої швидкості, а також візуальні ефекти при зіткненнях. У методі *Awake()* виконується початкове налаштування компонента *Rigidbody*, що використовується для фізичної симуляції руху м'яча. Основна логіка руху реалізована у методі *FixedUpdate()*, який викликається синхронно з фізичним циклом Unity. У цьому методі виконується постійний контроль параметрів руху м'яча (рис. 4).

```
void FixedUpdate()
{
    if (!_gameStarted) return;

    lastVel = rb.linearVelocity;
    if (lastVel.sqrMagnitude < 1e-6f) lastVel = new Vector3(0.7f, 0.7f, 0f) * speed;
    Vector3 v = new Vector3(lastVel.x, lastVel.y, 0f);
    if (Mathf.Abs(v.x) < 0.12f) v.x = Mathf.Sign(v.x == 0 ? Random.value * 2f - 1f : v.x) * 0.12f;
    rb.linearVelocity = v.normalized * speed;
    rb.position = new Vector3(rb.position.x, rb.position.y, 2.93f);

    float spin = v.magnitude * rotationSpeed;

    rb.angularVelocity = new Vector3(0f, 0f, spin);
}
```

Рис. 4. Метод *FixedUpdate()* у класі *Ball*
Джерело: опрацювання власне

Метод *OnCollisionEnter(Collision c)* відповідає за реакцію м'яча на зіткнення з іншими об'єктами сцени (рис. 5). Якщо зіткнення відбулося з об'єктом типу *Bonus*, воно ігнорується, оскільки бонуси не повинні змінювати траєкторію польоту. Для всіх інших об'єктів спочатку отримується нормаль поверхні у точці контакту. На її основі обчислюється вектор відбиття за допомогою функції *Vector3.Reflect()*. Це є стандартною реалізацією фізичної моделі дзеркального відбиття: кут падіння дорівнює куту відбиття.

```
void OnCollisionEnter(Collision c)
{
    if (c.gameObject.CompareTag("Bonus")) return;

    Vector3 n = c.GetContact(0).normal;
    Vector3 inV = new Vector3(lastVel.x, lastVel.y, 0f);
    Vector3 r = Vector3.Reflect(inV, new Vector3(n.x, n.y, 0f));
    if (Mathf.Abs(r.x) < 0.12f) r.x = Mathf.Sign(r.x == 0 ? Random.value * 2f - 1f : r.x) * 0.12f;
    rb.linearVelocity = r.normalized * speed;
    rb.position += n * 0.005f;
    rb.angularVelocity = Vector3.zero;

    if (c.gameObject.CompareTag("Block"))
    {
        Block block = c.gameObject.GetComponent<Block>();
        if (block) block.Damage(1);

        Vector3 spawnPos = c.contacts[0].point + Vector3.up * 0.3f;
        GameObject go = Instantiate(prefabFloatingDamage, spawnPos, Quaternion.identity, GameObject.Find("WorldUI").GetComponent<Transform>());
        go.transform.LookAt(go.transform.position + Camera.main.transform.forward);

        GameObject blockDamageParticle = Instantiate(prefabBlockDamageParticle, spawnPos, Quaternion.identity);
    }
    else if (c.gameObject.CompareTag("Player"))
    {
        Vector3 spawnPos = c.contacts[0].point + Vector3.up * 0.3f;
        GameObject playerDamageParticle = Instantiate(prefabPlayerDamageParticle, spawnPos, Quaternion.identity);
    }
}
```

Рис. 5. Метод *OnCollisionEnter(Collision c)* у класі *Ball*
Джерело: опрацювання власне

У процесі розроблення тривимірної гри «*Arkanoid*» на базі *Unity* було проведено комплексне тестування основних компонентів програмної системи відповідно до визначених категорій вимог. Протестовано класи, що відповідають за поведінку м'яча, платформи та блоків. Було підтверджено правильність обробки колізій через систему *Collider* та *Rigidbody*, адекватність відбиття м'яча відповідно до фізичної моделі, а також коректне оновлення стану ігрових об'єктів при взаємодії [5]. Перевірено логіку взаємодії гравця з ігровим середовищем. Зокрема, реалізовано та протестовано механізм зменшення кількості життів у разі втрати м'яча, перехід до наступного рівня після знищення всіх блоків, а також можливість призупинення гри через обробку натискання клавіші *ESC* із використанням *Time.timeScale*. Ігровий застосунок стабільно функціонує на операційних системах *Windows 10–11* при різних роздільностях екрана. Частота кадрів (FPS) залишається на рівні не

нижче 60 під час активного ігрового процесу. Аудіосистема Unity (AudioSource, AudioListener) забезпечує коректне відтворення звукових ефектів зіткнень, бонусів та інших ігрових подій.

Для обробки помилок було протестовано сценарії, пов'язані з відсутністю необхідних ресурсів, зокрема сцен, префабів або матеріалів. У всіх випадках система не завершує роботу аварійно, а обробляє помилки шляхом логування в консолі Unity та виведення відповідних повідомлень. Виконання вимог до супроводу забезпечується завдяки використанню компонентно-орієнтованої архітектури Unity. Логіка розподілена між окремими MonoBehaviour-скриптами, що відповідають за конкретний функціонал. Такий підхід спрощує модифікацію системи, додавання нових рівнів, бонусів або ігрових механік без значного втручання в існуючий код.

У межах вимог до збереження даних було перевірено механізми збереження ігрового прогресу. Поточний рівень, кількість життів і набрані очки зберігаються між ігровими сесіями за допомогою стандартних засобів Unity (наприклад, PlayerPrefs або локальних файлів). Відповідність вимогам доступності досягнута завдяки реалізації налаштувань гучності звуку, а також адаптивного користувацького інтерфейсу, який коректно масштабується під різні роздільності екрана. Застосунок підтримує керування як з клавіатури, так і з ігрового контролера, використовуючи стандартну систему Input System у Unity, що забезпечує універсальність та зручність використання. Окрім функціонального та структурного тестування, у процесі розроблення було застосовано модульне тестування. Було перевірено окремі компоненти програмної системи, зокрема логіку руху м'яча, обробку колізій та зміну станів ігрових об'єктів. Тестування здійснювалося із використанням вбудованих засобів Unity Test Framework, що дозволило перевірити коректність роботи окремих методів незалежно від інших частин системи.

***Анотація.** У рамках дослідження здійснено проєктування і практичну реалізацію тривимірної ігрової системи типу «Аркоїд». На початковому етапі було визначено вимоги до апаратного та програмного забезпечення, було детально розглянуто логіку функціонування ігрової системи. Зокрема, описано*

головний ігровий цикл, алгоритм руху та відбиття м'яча, а також механізм мережевої синхронізації. У рамках реалізації програмної частини було розроблено основні класи системи, зокрема клас керування гравцем та клас м'яча. Додатково було сформовано архітектуру програмної системи з розподілом на функціональні рівні, що забезпечує модульність і гнучкість розробки. Побудовано діаграми, які відображають структуру та взаємодію компонентів, що дозволило формалізувати внутрішню організацію програмного продукту. Також реалізовано користувацький інтерфейс, що забезпечує зручну взаємодію гравця з ігровим середовищем. Окрему увагу приділено тестуванню програмної системи, у результаті чого підтверджено коректність роботи основних функціональних компонентів, стабільність продуктивності та відповідність нефункціональним вимогам. Отримані результати свідчать про надійність і ефективність реалізованого програмного продукту.

Ключові слова: арканойд, дослідницьке навчання, варіативні моделі, об'єктно орієнтованого програмування, комп'ютерне моделювання, варіативні моделі, ігрова система, AI, КОМСДН, OpenGL, SDL2.

Література

1. Hrybiuk O. Experience in Implementing Computer-Oriented Methodological Systems of Natural Science and Mathematics Research Learning in Ukrainian Educational Institutions. In: Machado J., Soares F. (eds) *Innovations in Mechatronics Engineering. Lecture Notes in Mechanical Engineering*, 2022. P. 55-68. Springer, Cham Online.
2. Hrybiuk O. Problems of expert evaluation in terms of the use of variative models of a computer-oriented learning environment of mathematical and natural science disciplines in schools, [w:] *Zeszyty Naukowe Politechniki Poznańskiej. Seria: Organizacja i Zarządzanie*, Zeszyt Nr 79, Poznań: Wydawnictwo Politechniki Poznańskiej (WPP), 2019. P. 101-119.
3. Гриб'юк О.О. Імерсивні технології в освіті: особливості когнітивного розвитку дитини у віртуальному середовищі в процесі дослідницького навчання. *Сучасні інформаційні технології та інноваційні методики навчання в підготовці фахівців: методологія, теорія, досвід, проблеми: збірник наукових праць*. Київ-Вінниця: ТОВ Фірма «Планер», 2020.
4. Гриб'юк О.О. Комп'ютерне моделювання та робототехніка в навчально-виховному процесі сучасного навчального закладу. // *Сьома міжнародна науково-практична конференція FOSS Lviv-2017: Збірник наукових праць*, 27-30 квітня 2017 року, м. Львів. С. 38-43. ISBN 978-966-2598-86-5.
5. Hrybiuk Olena. Engineering in Educational Institutions: Standards for Arduino Robots as an Opportunity to Occupy an Important Niche in Educational Robotics in the Context of Manufacturing 4.0, in: *Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*. Volume 27-32, 2020. P. 770-785.
6. Hrybiuk O., Vedishcheva O. Experimental Teaching of Robotics in the Context of Manufacturing 4.0: Effective Use of Modules of the Model Program of Environmental Research Teaching in the Working Process of the Centers "Clever". In: , et al. *Innovations in Mechatronics Engineering II. icieng 2022. Lecture Notes in Mechanical Engineering*. Springer, Cham. 2022.
7. Hrybiuk O. Improvement of the Educational Process by the Creation of Centers for Intellectual Development and Scientific and Technical Creativity. In: Hamrol A., Kujawińska A., Barraza M. (eds). *Advances in Manufacturing II. MANUFACTURING 2019. Lecture Notes in Mechanical Engineering*, 2019. P. 370-382. Springer, Cham Online.

Прилуцький М. І. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О
*Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-
технічний університет імені академіка Юрія Бугая» (Київ)*

РОЗРОБЛЕННЯ МАТЕМАТИЧНО ОБГРУНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПАРСИНГУ ДАНИХ

Особливо актуальною є проблема автоматизації накопичення та опрацювання даних у контексті інформації про відключення електроенергії та ринку оголошень. В умовах воєнного стану в Україні інформація про графіки відключення світла має критичне значення для населення, оскільки дозволяє планувати діяльність та підготуватися до періодів відсутності електропостачання. Однак така інформація часто публікується у різних форматах і на різних платформах, що ускладнює її збір та структурування.

Традиційні методи збору даних, що передбачають ручне копіювання інформації, мають низку недоліків: висока трудомісткість та часову затратність, можливість людських помилок, неможливість обробки великих обсягів даних у реальному часі, складність підтримки актуальності інформації та обмежена масштабованість процесу.

Автоматизовані системи парсингу даних дозволяють усунути ці недоліки, забезпечуючи безперервний моніторинг джерел інформації, автоматичне витягування та структурування даних, високу швидкість обробки великих обсягів інформації, мінімізацію помилок завдяки алгоритмічній обробці та можливість масштабування для обробки додаткових джерел [1]. Таким чином, розробка інформаційної системи парсингу даних є актуальною задачею, яка дозволяє автоматизувати процеси збору, обробки та надання структурованої інформації користувачам у зручному форматі.

Мета дослідження полягає у створенні комплексної інформаційної системи парсингу даних, яка автоматизує процеси збору інформації про графіки відключення електроенергії з різних джерел (текстові повідомлення та

зображення), парсингу оголошень з веб-сайтів за заданими критеріями, зберігання та структурування зібраних даних у базі даних та надання користувачам доступу до інформації через Telegram-бота з можливістю налаштування персональних оповіщень.

У ході дослідження було вивчено актуальне питання вибору архітектури, інструментів та алгоритмів для побудови інформаційної системи парсингу даних. На ринку програмного забезпечення існує значна кількість рішень для парсингу та збору даних з веб-ресурсів. Всі вони можуть бути класифіковані за різними критеріями: типом ліцензії, способом використання, технологічним стеком та функціональними можливостями. Серед комерційних рішень найбільш відомі нижче наведені: *Scrapy* (open-source фреймворк для веб-скрапінгу на Python), *Beautiful Soup* (бібліотека для парсингу HTML/XML), *Selenium* (інструмент для автоматизації браузерів), *Puppeteer* (бібліотека для керування headless Chrome) та *Apify* (платформа для веб-скрапінгу та автоматизації). Комерційні сервіси, такі як *ParseHub*, *Octoparse*, *Import.io*, пропонують графічні інтерфейси для створення парсерів без написання коду, однак вони мають обмеження щодо складності задач, вартості та залежності від зовнішніх сервісів [2].

Інформаційна система складається з двох основних модулів: модуль парсингу оголошень *OLX* та модуль парсингу графіків планових відключень електроенергії з Telegram-каналу Черкаської ОДА. Кожен модуль реалізує автономні алгоритми обробки даних та взаємодіє з базою даних для збереження отриманої інформації. Модуль парсингу *OLX* забезпечує автоматичний збір нових оголошень на основі заданих параметрів. Вхідними даними є категорія товару, ключові слова для фільтрації, а також мінімальна та максимальна ціна.

На основі цих параметрів формується URL для запиту до платформи *OLX*. Система отримує *HTML-контент* сторінки та за допомогою бібліотеки *BeautifulSoup* визначає блоки з оголошеннями. Кожне оголошення перевіряється на відповідність умовам: дата публікації не менше часу запуску парсера, відповідність ключовим словам [3] та ціна. Для оголошень, що

відповідають умовам, витягується детальна інформація у форматі JSON, що включає назву, опис, локацію, ціну, зображення та *URL* оголошення. Отримані дані зберігаються у базі даних, а ідентифікатори оброблених оголошень зберігаються у локальному файлі для запобігання дублюванню.

Модуль парсингу відключень електроенергії працює з *Telegram-каналом* Черкаської ОДА. Система отримує *HTML-контент* сторінки каналу та визначає останній пост за допомогою регулярних виразів (рис. 1).

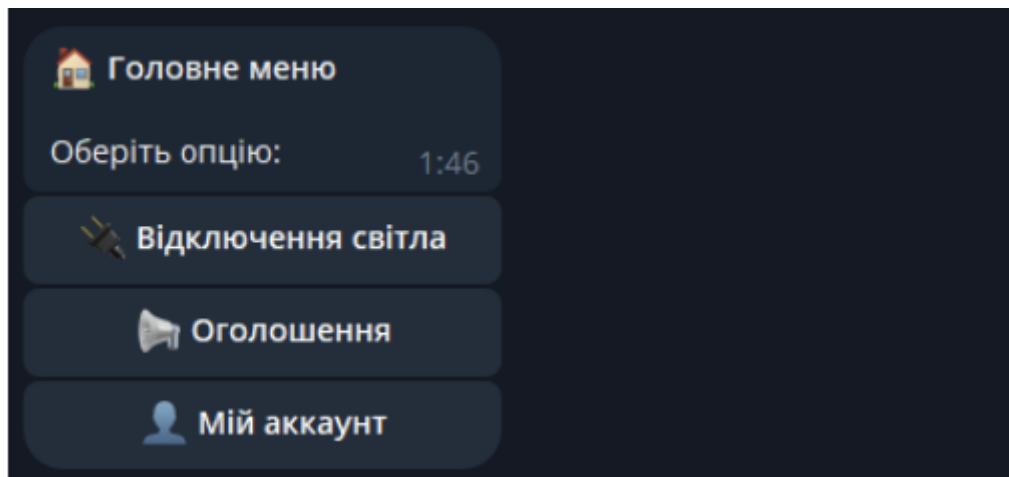


Рис. 1. Інтерфейс головної сторінки ТБПД
Джерело: опрацювання власне

Текст кожного поста аналізується для виділення дати та часу відключень. Регулярні вирази дозволяють визначити номери груп та інтервали часу відключень, які формуються у структуровані масиви. Модуль забезпечує перевірку графіків на актуальність щодо поточної дати, а також розрізняє графіки на сьогодні та на завтра. Отримані дані передаються у базу даних для використання користувачами системи або іншими підсистемами. Архітектура системи модульна та асинхронна.

Кожен модуль працює автономно і взаємодіє з іншими компонентами через базу даних. Використання асинхронних запитів *aiohhttp* дозволяє зменшити час очікування при отриманні даних, а бібліотека *BeautifulSoup* забезпечує ефективну обробку HTML. Для обробки текстових даних застосовуються регулярні вирази, що дозволяє швидко адаптувати систему під зміну формату даних у джерелах [4]. Структура даних передбачає використання

JSON-формату для зберігання оголошень OLX та словників для графіків відключень електроенергії (рис. 2).

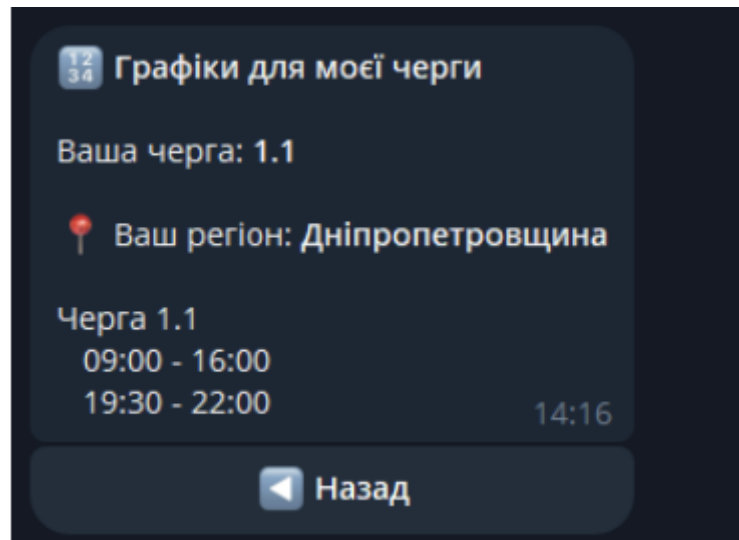


Рис. 2. Моніторинг відключень по області з урахуванням черги
Джерело: опрацювання власне

Для кожного оголошення зберігаються: ідентифікатор, URL, назва, опис, ціна, локація та зображення. Для графіків відключень електроенергії зберігаються дати, номери груп та часові інтервали, що забезпечує швидку обробку та аналіз інформації. Для забезпечення стабільності роботи системи передбачено логування подій, обробку помилок при отриманні даних та відновлення після збоїв. Асинхронна обробка дозволяє працювати з великою кількістю запитів без блокування програми.

Технічні обмеження включають захист від ботів та автоматизації, оскільки багато веб-сайтів використовують різні механізми для виявлення та блокування автоматизованих запитів: *CAPTCHA* (*Completely Automated Public Turing test to tell Computers and Humans Apart*) - тести для перевірки, що запит надсилає людина, перевірка *User-Agent* та інших *HTTP-заголовків*, аналіз поведінки (швидкість запитів, патерни навігації), використання *JavaScript* для перевірки наявності браузера та *fingerprinting* браузера для виявлення автоматизації. Динамічний контент є проблемою, оскільки сучасні веб-додатки все частіше використовують *JavaScript* для динамічного завантаження контенту. Простий *HTTP-запит* не дозволяє отримати такий контент, оскільки він генерується після виконання *JavaScript-коду* в браузері.

Анотація. На основі проведених досліджень здійснено проектування і розроблено комплексну інформаційну систему парсингу даних, яка автоматизує процеси збору інформації про графіки відключення електроенергії з різних джерел, парсингу оголошень з веб-сайтів за заданими критеріями, зберігання та структурування зібраних даних у базі даних та надання користувачам доступу до інформації через Telegram-бота з можливістю налаштування персональних оповіщень. ІС також призначена для операторів аналітичних і моніторингових служб, що включає модулі реєстрації та авторизації користувачів і дозволяє автоматизувати процес збору даних, знизити трудові витрати, підвищити оперативність реагування на нові дані та забезпечує можливість масштабування для інтеграції нових джерел інформації. Процес парсингу веб-даних стикається з численними обмеженнями та перешкодами, які розробники намагаються подолати для забезпечення ефективного збору інформації. Розуміння цих обмежень та методів їх подолання є критично важливим для створення надійних інформаційних систем парсингу даних.

Ключові слова: дослідницьке навчання, варіативні моделі, математика, комп'ютерне моделювання, КОМСДН, CleverCOMSRL, інформаційна система, парсингування, інтелектуальний аналіз даних, AI.

Література

1. Hrybiuk, O. Improvement of the educational process by the creation of centers for intellectual development and scientific and technical creativity. In: Hamrol, A., Kujawińska, A., Barraza, M.F.S. (eds.) *MANUFACTURING 2019*. LNME, 2019. P. 370–382. Springer, Cham. https://doi.org/10.1007/978-3-030-18789-7_31.
2. Hrybiuk, O., Kant, G.S. CleverCOMSRL: Implementation of an AI Computer-Aided Design System in the Context of the Cognitive Science Paradigm for the Research Training Process. In: Machado, J., et al. *Innovations in Mechatronics Engineering III. icieng 2024. Lecture Notes in Mechanical Engineering*. Springer, Cham. 2024. P. 351-362. https://doi.org/10.1007/978-3-031-61575-7_32.
3. Hrybiuk, O.O. Research learning of the natural science and mathematics cycle using computer-oriented methodological systems. Monograph. Drahomanov NPU, Kyiv, 2019. P. 277–325.
4. Khang A., Abdullayev V., Hrybiuk O., & Shukla A.K. (Eds.). *Computer Vision and AI-Integrated IoT Technologies in the Medical Ecosystem* (1st ed.). CRC Press. 2024. <https://doi.org/10.1201/9781003429609>.

Руденко О. В. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О
*Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-
технічний університет імені академіка Юрія Бугая» (Київ)*

РОЗРОБЛЕННЯ ІГРОВОЇ СИСТЕМИ «ТРИ В РЯД» ІЗ ВИКОРИСТАННЯМ МАТЕМАТИЧНИХ ОСНОВ ТЕХНОЛОГІЇ ОБ'ЄКТНО ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ

Актуальність дослідження полягає у вивченні принципів створення ігрових застосунків та реалізації власної гри з використанням сучасних підходів до програмування. Ігри жанру «Три в ряд» належать до логічних ігор, які поєднують прості правила, швидке освоєння та динамічний ігровий процес. Основна ідея таких ігор полягає у тому, що гравець взаємодіє з елементами на ігровому полі та формує комбінації з трьох і більше однакових об'єктів. Завдяки цьому жанр залишається популярним серед широкого кола користувачів, оскільки не потребує складного керування та спеціальної підготовки [1].

Передумовою для розроблення власної ігрової системи є необхідність створення простої, зрозумілої та стабільної гри, яка реалізує основні механіки жанру «Три в ряд». Розроблена ігрова система має зручний для користувачів інтерфейс, забезпечує швидку реакцію на дії користувача та підтримує базові ігрові процеси: створення ігрового поля, переміщення елементів, перевірку комбінацій, нарахування балів і завершення рівня.

Мета дослідження полягає у розробленні ігрового застосунку жанру «Три в ряд» із підтримкою мультиплеєру та адаптивною системою управління у контексті виваженого використання технології об'єктно орієнтованого програмування. Визначення системних вимог є критично важливим етапом проектування ігрової системи, оскільки це безпосередньо впливає на вибір технологічного стеку, методів оптимізації графіки та архітектуру мережевої взаємодії [2]. У розробленій ігровій системі передбачається використання

класичної механіки жанру: гравець переміщує сусідні елементи, після чого система перевіряє наявність комбінацій. Якщо комбінацію знайдено, відповідні елементи видаляються, на їх місце з'являються нові, а гравцю нараховуються бали [3]. Важливою передумовою розроблення є також використання об'єктно-орієнтованого підходу. Ігрова система складається з окремих взаємопов'язаних компонентів: ігрового поля, елементів, системи керування, підрахунку балів та інтерфейсу користувача. Поділ програми на окремі класи дозволяє зробити структуру гри більш зрозумілою та зручною для подальшого вдосконалення. Розроблення гри жанру «Три в ряд» дозволяє краще зрозуміти, як працює ігрова логіка, а також застосувати на практиці основні принципи об'єктно-орієнтованого програмування [4].

Основні концепції ігрової системи «StreakMatch» визначають принципи роботи гри, її функції та вимоги до взаємодії користувача з програмою. Розроблювана система базується на *класичній механіці жанру*, де гравець переміщує сусідні елементи на ігровому полі для утворення комбінацій із трьох і більше однакових об'єктів [5].

Ігрове поле формується у вигляді сітки розміром 8×8. На початку гри система автоматично заповнює поле випадковими елементами, але без готових комбінацій. Це дозволяє забезпечити коректний старт ігрової сесії. Після виконання ходу система перевіряє його допустимість і визначає, чи утворилася комбінація. Якщо комбінацію знайдено, відповідні елементи видаляються з поля, гравцеві нараховуються бали, а порожні клітинки заповнюються новими елементами [6]. У разі послідовного утворення нових комбінацій виникає каскадний ефект.

Важливою частиною гри є система бонусів. При утворенні комбінацій із чотирьох або п'яти елементів можуть створюватися спеціальні елементи, які дозволяють очищати рядок, стовпець або групу елементів, ще робить ігровий процес більш різноманітним і підвищує інтерес користувача [7].

Анотація. У рамках дослідження здійснено проектування і практичну реалізацію тривимірної ігрової системи типу «Три в ряд». На початковому

етапі було визначено вимоги до апаратного та програмного забезпечення, було детально розглянуто логіку функціонування ігрової системи. Зокрема, описано головний ігровий цикл, алгоритм руху та відбиття м'яча, а також механізм мережевої синхронізації. У рамках реалізації програмної частини було розроблено основні класи системи, зокрема клас керування гравцем та клас м'яча. Додатково було сформовано архітектуру програмної системи з розподілом на функціональні рівні, що забезпечує модульність і гнучкість розробки. Побудовано діаграми, які відображають структуру та взаємодію компонентів, що дозволило формалізувати внутрішню організацію програмного продукту. Також реалізовано користувацький інтерфейс, що забезпечує зручну взаємодію гравця з ігровим середовищем. Окрему увагу приділено тестуванню програмної системи, у результаті чого підтверджено коректність роботи основних функціональних компонентів, стабільність продуктивності та відповідність нефункціональним вимогам. Отримані результати свідчать про надійність і ефективність реалізованого програмного продукту.

Ключові слова: головоломки, дослідницьке навчання, варіативні моделі, об'єктно орієнтованого програмування, три в ряд, комп'ютерне моделювання, варіативні моделі, ігрова система, математичне моделювання, AI, КОМСДН.

Література

1. Hrybiuk O. Experience in Implementing Computer-Oriented Methodological Systems of Natural Science and Mathematics Research Learning in Ukrainian Educational Institutions. In: Machado J., Soares F. (eds) *Innovations in Mechatronics Engineering. Lecture Notes in Mechanical Engineering*, 2022. P. 55-68. Springer, Cham Online.
2. Hrybiuk O. Problems of expert evaluation in terms of the use of variative models of a computer-oriented learning environment of mathematical and natural science disciplines in schools, [w:] *Zeszyty Naukowe Politechniki Poznańskiej. Seria: Organizacja i Zarządzanie*, Zeszyt Nr 79, Poznań: Wydawnictwo Politechniki Poznańskiej (WPP), 2019. P. 101-119.
3. Гриб'юк О.О. Імерсивні технології в освіті: особливості когнітивного розвитку дитини у віртуальному середовищі в процесі дослідницького навчання. *Сучасні інформаційні технології та інноваційні методики навчання в підготовці фахівців: методологія, теорія, досвід, проблеми: збірник наукових праць*. Київ-Вінниця: ТОВ Фірма «Планер», 2020.
4. Гриб'юк О.О. Комп'ютерне моделювання та робототехніка в навчально-виховному процесі сучасного навчального закладу. // *Сьома міжнародна науково-практична конференція FOSS Lviv-2017: Збірник наукових праць*, 27-30 квітня 2017 року, м. Львів. С. 38-43. ISBN 978-966-2598-86-5.
5. Hrybiuk Olena. Engineering in Educational Institutions: Standards for Arduino Robots as an Opportunity to Occupy an Important Niche in Educational Robotics in the Context of Manufacturing 4.0, in: *Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*. Volume 27-32, 2020. P. 770-785.
6. Hrybiuk O., Vedishcheva O. Experimental Teaching of Robotics in the Context of Manufacturing 4.0: Effective Use of Modules of the Model Program of Environmental Research Teaching in the Working Process of the Centers "Clever". In: , et al. *Innovations in Mechatronics Engineering II. icieng 2022. Lecture Notes in Mechanical Engineering*. Springer, Cham. 2022.
7. Hrybiuk O. Improvement of the Educational Process by the Creation of Centers for Intellectual Development and Scientific and Technical Creativity. In: Hamrol A., Kujawińska A.,

УДК 004.942:[3275

Свидан А. А. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О
Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-технічний університет імені академіка Юрія Бугая» (Київ)

РОЗРОБЛЕННЯ ГРИ «ФУТБОЛ» ІЗ ВИКОРИСТАННЯМ МАТЕМАТИЧНИХ ОСНОВ ТЕХНОЛОГІЇ ОБ'ЄКТНО ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ

Актуальність дослідження. Комп'ютерна гра «Футбол» належить до жанру спортивних симуляторів, які відтворюють динаміку та атмосферу реальних футбольних матчів. Такі ігри розвивають стратегічне мислення, тренують швидкість реакції й допомагають краще зрозуміти основні принципи командної взаємодії. Футбол є однією з найпопулярніших спортивних дисциплін у світі, тому створення його комп'ютерної версії викликає інтерес серед користувачів різного віку [1].

Мета дослідження полягає у проєктуванні та розробленні гри «Футбол» із використанням технології об'єктно орієнтованого програмування. *Завдання:*
1. Здійснити аналіз вимог до програмної системи і дослідити існуючі застосунки, сформувавши вимоги до ігрової системи «Футбол». 2. Здійснити проєктування ігрової системи «Футбол», розробивши архітектуру та відповідні бази даних. 3. Здійснити реалізацію програмного забезпечення, розробивши функціонал ігрової системи «Футбол». 4. Здійснити тестування та відповідну перевірку ігрової системи на предмет функціональності, безпеки та відповідності вимогам.

Основні аспекти створення об'єктів гри: ігрове поле, гравці, м'яч, ворота, табло з рахунком. *Реалізація взаємодії між об'єктами:* керування гравцем, передача м'яча, удари, зіткнення м'яча з полем і воротами, відскоки та зміна напрямку руху. *Фізика гри:* траєкторія польоту м'яча, сила удару,

швидкість руху гравців, зіткнення з іншими об'єктами, розрахунок відскоків і зміна положення м'яча залежно від сили взаємодії. *Компоненти гри*: ігрове поле, гравці двох команд, м'яч, ворота, індикатор рахунку, таймер матчу, меню керування [2]. У процесі розроблення виконано аналіз основних вимог до функціонування футбольного симулятора та визначено структуру системи, необхідну для реалізації гри. Процес розробки ігрового застосунку «V-GOAL» вимагає чіткого визначення комплексу технічних, архітектурних та геймдизайнерських вимог. Вони є основою для вибору програмних інструментів та побудови стабільної архітектури, здатної забезпечити високу продуктивність та комфортний ігровий досвід [3].

Усі вимоги до проєктування системи можна розділити на кілька ключових категорій: *вимоги до апаратно-програмного забезпечення*: *кросплатформність*: Гра повинна запускатися на будь-якій операційній системі (Windows, macOS, Linux), що підтримує інтерпретатор Python. *Мова програмування та бібліотеки*: Основною мовою розробки обрано Python 3.x. Для обробки графіки, вводу з клавіатури та відтворення звуку необхідна мультимедійна бібліотека pygame. Для обчислення фізики (кутів, векторів) використовується стандартний модуль math, а для генерації ефектів погоди - модуль random. Системні шляхи до ресурсів (спрайтів, звуків) обробляються через модуль pathlib для уникнення помилок маршрутизації на різних ОС [4]. *Апаратні вимоги*: Завдяки оптимізованій 2D-графіці (Pixel Art), система не вимагає наявності дискретної відеокарти і повинна стабільно працювати на базових офісних комп'ютерах зі стандартними процесорами.

Архітектурні вимоги до рушія. Ігровий цикл (Game Loop): Архітектура повинна базуватися на класичному неперервному циклі, який чітко розділяє три фази: обробку подій (ввід користувача), оновлення логіки (розрахунок фізики, ІІІ, витривалості) та рендеринг (відмальовка об'єктів на екран). Синхронізація часу: Оновлення ігрової логіки має залежати від частоти кадрів (clock.tick(FPS)). Продуктивність повинна бути зафіксована на рівні 60 FPS для

забезпечення максимальної плавності руху та уникнення фізичних «проскоків» м'яча крізь текстури [5].

Вимоги до ігрової механіки та штучного інтелекту. Векторна фізика: Рух м'яча та гравців має розраховуватися у числах з плаваючою комою (float) для забезпечення ідеально плавних траєкторій. Система враховує інерцію, максимальну швидкість об'єктів та тертя об газон (BALL_FRICTION). Комп'ютерні суперники адекватно реагують на ігрову ситуацію. Воратар рухається виключно вздовж лінії воріт (з урахуванням 2.5D перспективи), а польові гравці застосовують тактику зонного захисту, повертаючись на стартові позиції, якщо м'яч знаходиться далеко [6]. У дослідженні використовується механіка витривалості (Stamina) для спринту, щоб мотивувати гравців приймати тактичні рішення, а не просто постійно використовувати максимальну швидкість.

Вимоги до інтерфейсу користувача (UI) та візуалізації: Графічний інтерфейс: наявність головного меню з можливістю запуску гри, виклику екрану налаштувань (для перемикання режимів PvE/PvP та звуку) і коректного виходу. Інтерфейс адаптивний до подій (наприклад, підсвічування кнопок при наведенні миші). Система зворотного зв'язку [7]: гравець отримує чіткий візуальний та аудіальний зворотний зв'язок: відображення шкали витривалості над активним гравцем, зміна рахунку на табло, звуки удару, гола та свистка судді. Багатошаровий рендеринг: Для створення глибини кадру відмальовка відбувається у строгому порядку: фон, тіні, гравці, м'яч, візуальні ефекти (сніг) та текстовий інтерфейс (UI).

Гра представляє собою 2D-симуляцію зимового футболу, розроблену за допомогою мови програмування Python та бібліотеки Pygame. Проєкт реалізує динамічний ігровий процес з акцентом на фізичну взаємодію об'єктів та атмосферну візуалізацію. У межах проєкту реалізовано наступні механіки: *керування гравцями:* система переміщення польових гравців та специфічна логіка руху воратарів. *Фізика м'яча:* реалістичні відскоки, тертя об поверхню та взаємодія з гравцями. Забезпечується автоматизована поведінка ботів,

включаючи зонний захист та тактичні паси. *Система налаштувань*: вибір режиму гри (одиначний або локальний мультиплеєр) та керування звуковими ефектами. *Візуальні ефекти*: анімація падаючого снігу та динамічні тіні для створення глибини сцени. *Основні механіки гри*. *Керування гравцем*. Користувач керує нападником за допомогою клавіш WASD. Реалізовано механіку *спринту* (клавіша Shift), що дозволяє прискорюватися, витрачаючи обмежений запас витривалості (Stamina). Для PvP-режиму додано підтримку керування стрілками для другого гравця. Воратарі мають унікальну логіку руху вздовж ліній воріт з урахуванням перспективи ігрового поля (2.5D).

Використано алгоритм лінійної інтерполяції для коректного відображення руху по діагоналі. М'яч має векторну модель швидкості. Реалізовано анти-кут логіку для ботів: при затисканні м'яча біля стінки ШІ автоматично виконує пас у центр поля для уникнення застрягання об'єктів. Гол зараховується при повному перетині м'ячем лінії воріт у межах штанг. Після взяття воріт спрацьовує звуковий ефект, а позиції гравців скидаються до початкових. *Графічні та візуальні елементи*. Для гри обрано піксельний стиль (Pixel Art). Використовується статичне тло із зимовим стадіоном, поверх якого накладаються динамічні шари (рис. 1).



Рис. 1. Візуальна реалізація ігрового процесу на зимовому полі
Джерело: опрацювання власне

Шар снігу: клас Snow генерує масив об'єктів-сніжинок, що рухаються з випадковою швидкістю та вектором вітру. *Шар тіней*: під кожним спрайтом

малюється напівпрозорий еліпс, що візуально «приземляє» персонажів на поле. У рамках дослідження розроблено ескізний проєкт та прототипи елементів гри. Вони допомагають візуалізувати зовнішній вигляд гравців, м'яча, воріт, ігрового поля, інтерфейсу гри, меню, налаштувань та панелі перемоги. Усі прототипи створені самостійно відповідно до концепції гри. Основним фоном гри обрано зимовий стадіон у стилі Pixel Art, що дозволяє поєднати спортивну динаміку з атмосферним візуальним рядом.

Однією з ключових частин є клас Player, який відповідає за безпосереднє керування рухом гравця по полю, а також за логіку штучного інтелекту (рис. 2). Фізичне переміщення об'єкта реалізовано з використанням координат з плаваючою комою (float) для забезпечення максимальної плавності, а зчитування вводу здійснюється через системний метод `pygame.key.get_pressed()`. Окрему увагу приділено механіці спринту: гравець може прискорюватися за допомогою клавіші Shift, але це обмежується показником витривалості (stamina). Наведений нижче лістинг демонструє, як у методі `update()` зчитується рух гравця та застосовується логіка витрати сил.

```
if self.control_type == "wasd":
    dx = keys[pygame.K_d] - keys[pygame.K_a]
    dy = keys[pygame.K_s] - keys[pygame.K_w]
    kick_pressed = keys[pygame.K_SPACE]
    sprint_pressed = keys[pygame.K_LSHIFT]

    current_speed = self.speed
    is_moving = (dx != 0 or dy != 0)

    if sprint_pressed and is_moving and self.stamina > 0:
        current_speed = self.speed * 1.6
        self.stamina = max(0.0, self.stamina - 1.0)
    else:
        self.stamina = min(self.max_stamina, self.stamina + 0.5)
```

Рис. 2. Рух гравця та механіка спринту (клас Player)

Джерело: опрацювання власне

Клас Ball реалізує усю логіку, пов'язану з м'ячем. Він розраховує тертя об газон, інерцію та обробляє відскоки від віртуальних меж поля, не

дозволяючи м'ячу вилетіти за межі екрана (окрім зони воріт). Механіка удару базується на тригонометричних розрахунках (рис. 3): при натисканні відповідної клавіші обчислюється кут між гравцем і м'ячем за допомогою функції `atan2`, після чого м'ячу надається векторна швидкість. Центральну логіку гри, таку як підрахунок голів та контроль таймера, зібрано у головному ігровому циклі функції `main_game()`. Вона виконує роль своєрідного менеджера стану. Функція `check_goal()` перевіряє, чи перетнув м'яч лінію воріт. Якщо так, програма відтворює відповідний звуковий ефект, оновлює рахунок на екрані, активує коротку паузу та повертає всі об'єкти на стартові позиції за допомогою методу `reset_position()`.

```
goal = check_goal(ball, width, height)
if goal:
    play_sound("goal")

    if goal == 'a': score_a += 1
    else: score_b += 1

    goal_pause = 1.0
    ball.x, ball.y, ball.vx, ball.vy = float(width//2), float(height//2), 0, 0
    for p in players: p.reset_position()
    round_start_time = pygame.time.get_ticks()
```

Рис. 3. Фіксація гола та скидання позицій
Джерело: опрацювання власне

Для ботів розроблено систему зонного захисту та спеціалізовану логіку для воротарів. Оскільки поле намальоване у перспективі 2.5D, воротар не може просто рухатися по прямій вертикальній лінії. Було імплементовано алгоритм лінійної інтерполяції, який автоматично розраховує координату X залежно від поточної позиції Y, змушуючи воротаря рухатися вздовж похилої лінії воріт.

Додатково в межах програми було створено систему користувацького інтерфейсу, що включає головне меню та екран налаштувань. Функція `settings_screen()` дозволяє перемикати ігровий режим (PvE проти ботів або PvP з іншим гравцем), а також керувати звуком. Стан цих налаштувань зберігається у

глобальному словнику GAME_SETTINGS, що забезпечує застосування обраних параметрів у наступних ігрових сесіях.

Окрему увагу приділено тестуванню програмної системи, у результаті чого підтверджено коректність роботи основних функціональних компонентів, стабільність продуктивності та відповідність нефункціональним вимогам. Отримані результати свідчать про надійність і ефективність реалізованого програмного продукту.

Анотація. У рамках дослідження здійснено проектування і практичну реалізацію тривимірної ігрової системи типу «Футбол». На початковому етапі було визначено вимоги до апаратного та програмного забезпечення, було детально розглянуто логіку функціонування ігрової системи. Зокрема, описано головний ігровий цикл, алгоритм руху та відбиття м'яча, а також механізм мережевої синхронізації. У рамках реалізації програмної частини було розроблено основні класи системи, зокрема клас керування гравцем та клас м'яча. Додатково було сформовано архітектуру програмної системи з розподілом на функціональні рівні, що забезпечує модульність і гнучкість розробки. Побудовано діаграми, які відображають структуру та взаємодію компонентів, що дозволило формалізувати внутрішню організацію програмного продукту. Також реалізовано користувацький інтерфейс, що забезпечує зручну взаємодію гравця з ігровим середовищем.

Ключові слова: футбол, дослідницьке навчання, варіативні моделі, математичне моделювання, об'єктно орієнтованого програмування, комп'ютерне моделювання, варіативні моделі, ігрова система, симулятор, AI, КОМСДН.

Література

1. Hrybiuk O. Experience in Implementing Computer-Oriented Methodological Systems of Natural Science and Mathematics Research Learning in Ukrainian Educational Institutions. In: Machado J., Soares F. (eds) *Innovations in Mechatronics Engineering. Lecture Notes in Mechanical Engineering*, 2022. P. 55-68. Springer, Cham Online.
2. Hrybiuk O. Problems of expert evaluation in terms of the use of variative models of a computer-oriented learning environment of mathematical and natural science disciplines in schools, [w:] *Zeszyty Naukowe Politechniki Poznańskiej. Seria: Organizacja i Zarządzanie*, Zeszyt Nr 79, Poznań: Wydawnictwo Politechniki Poznaskiej (WPP), 2019. P. 101-119.
3. Гриб'юк О.О. Імерсивні технології в освіті: особливості когнітивного розвитку дитини у віртуальному середовищі в процесі дослідницького навчання. *Сучасні інформаційні технології та інноваційні методики навчання в підготовці фахівців: методологія, теорія, досвід, проблеми: збірник наукових праць*. Київ-Вінниця: ТОВ Фірма «Планер», 2020.
4. Гриб'юк О.О. Комп'ютерне моделювання та робототехніка в навчально-виховному процесі сучасного навчального закладу. // *Сьома міжнародна науково-практична конференція FOSS Lviv-2017: Збірник наукових праць*, 27-30 квітня 2017 року, м. Львів. С. 38-43. ISBN 978-966-2598-86-5.
5. Hrybiuk Olena. Engineering in Educational Institutions: Standards for Arduino Robots as an Opportunity to Occupy an Important Niche in Educational Robotics in the Context of

Manufacturing 4.0, in: *Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*. Volume 27-32, 2020. P. 770-785.

6. Hrybiuk O., Vedishcheva O. Experimental Teaching of Robotics in the Context of Manufacturing 4.0: Effective Use of Modules of the Model Program of Environmental Research Teaching in the Working Process of the Centers “Clever”. In: , et al. *Innovations in Mechatronics Engineering II. icieng 2022. Lecture Notes in Mechanical Engineering*. Springer, Cham. 2022.

7. Hrybiuk O. Improvement of the Educational Process by the Creation of Centers for Intellectual Development and Scientific and Technical Creativity. In: Hamrol A., Kujawińska A., Barraza M. (eds). *Advances in Manufacturing II. MANUFACTURING 2019. Lecture Notes in Mechanical Engineering*, 2019. P. 370-382. Springer, Cham Online.

УДК 004.942:[327

Сидоров Н. Г (здобувач бакалаврату, 4 курс),

Науковий керівник – доц. Гриб'юк О. О

Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-технічний університет імені академіка Юрія Бугая» (Київ)

МАТЕМАТИЧНІ ПРИНЦИПИ СТВОРЕННЯ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ ДЛЯ ДОСЛІДЖЕННЯ РИНКУ НЕРУХОМОСТІ

Проблема багатьох сучасних компаній, що займаються дослідженням ринку нерухомості, полягає в тому, що вони досі використовують неефективні методи роботи. У сучасному цифровому світі величезну кількість даних генерують різноманітні джерела, починаючи від соціальних мереж і закінчуючи великими корпоративними базами даних. Цей потік інформації, що постійно зростає, має великий потенціал для виявлення цінної інформації та підтримки прийняття рішень. Збільшення обсягів даних ставить перед сучасними інформаційними системами нові виклики у забезпеченні їх ефективності та масштабованості [1].

Дослідження в галузі штучного інтелекту проводяться в таких напрямках: моделювання на ЕОМ окремих функцій творчих процесів, автоматичне доведення теорем, автоматичний синтез програм, аналіз і синтез. Технології Big Data стали відповіддю на ці виклики, пропонуючи нові підходи до збору, зберігання, обробки та аналізу великих обсягів даних [2]. Розвиток цих технологій відкриває широкі можливості для розроблення програмних

систем, спрямованих на комплексний аналіз даних та отримання інсайтів, які можуть бути використані для підтримки прийняття рішень у різних сферах, від бізнесу до науки та громадської діяльності.

Мета дослідження полягає у розробленні та програмній реалізації інформаційно-аналітичної системи для дослідження ринку нерухомості. Зростання обсягів даних, їх різноманітня та швидкість зміни створюють виклики для існуючих програмних систем, які не завжди можуть ефективно впоратися з такими обсягами інформації. Проблеми, які виникають у зв'язку з аналізом Big Data, включають: масштабованість, швидкість опрацювання даних, різноманітність даних, якість даних на ринку нерухомості. Розроблення архітектури програмної системи комплексного аналізу даних з використанням технологій Big Data є актуальною задачею, що вирішує наведені вище проблеми та відповідає потребам сучасного інформаційного середовища [3].

У рамках дослідження розроблено архітектуру інформаційно-аналітичної системи для дослідження ринку нерухомості і здійснення комплексного аналізу даних із використанням технологій Big Data, що відповідає основним параметрам: *ефективність*: здатність обробляти великі обсяги даних з високою швидкістю. *Масштабованість*: здатність розширюватися для роботи з ще більшими обсягами даних. *Гнучкість*: здатність адаптуватися до нових типів даних та завдань аналізу. *Простота у використанні*: доступність для користувачів з різним рівнем знань та досвіду. *Безпечність*: здатність захищати дані від несанкціонованого доступу.

У рамках дослідження здійснено ґрунтовне проектування архітектури інформаційно-аналітичної системи для дослідження ринку нерухомості з використанням комплексного аналізу даних з використанням технологій Big Data. Це включає в себе розробку концепції системи, вибір технологій, проектування архітектурних рішень, реалізацію програмного забезпечення та його тестування. Предмет дослідження також охоплює аналіз ефективності розробленої інформаційно-аналітичної системи для дослідження ринку нерухомості та порівняння з існуючими аналогами на ринку.

Big Data описує набори даних, що характеризуються 5V: *Volume*: об'єм вказує на величезну кількість даних, яку потрібно обробити та аналізувати [4]. Це можуть бути тексти, числа, зображення, відео, аудіо тощо. Потужність сучасних систем зберігання даних, таких як сервери та хмарні обчислення, дозволяє зберігати терабайти, петабайти, а навіть ексабайти інформації.

Velocity: швидкість описує темп, з яким дані генеруються та опрацьовуються. Великі потоки даних можуть надходити в реальному часі з сенсорів Інтернету речей (IoT) [5], соціальних мереж, медичного обладнання тощо. Здатність обробляти ці дані миттєво чи навіть передбачати їхній потік є ключовою для ефективного аналізу.

Variety: різноманіття означає, що дані можуть бути представлені у різних форматах та структурах. Вони можуть бути структурованими (наприклад, таблиці баз даних), неструктурованими (текстові файли, відео) або напів структурованими (XML-файли, JSON-об'єкти). Управління цією різноманітністю вимагає розуміння та використання різних методів обробки даних.

Veracity: вірогідність відноситься до якості даних та їхньої достовірності. При обробці великих обсягів даних можуть виникати проблеми з несправжніми, неточними або неповними даними. Важливо мати механізми для виявлення та коригування цих проблем для забезпечення точності аналізу.

Value: цінність вказує на потенційну користь, яку можна отримати від аналізу цих даних. Це може бути виявлення нових інсайтів, покращення прийняття рішень у бізнесі, розробка нових продуктів та послуг тощо. Здатність витягти цінну інформацію з великого обсягу даних визначає успішність використання *Big Data* в різних сферах діяльності.

Розуміння та врахування цих п'яти основних характеристик *Big Data* допомагає підготуватися до ефективного аналізу та використання великих обсягів даних у контексті дослідження ринку нерухомості [6].

У рамках дослідження виконано завдання:

1. Здійснено ґрунтовний аналіз можливостей та особливостей використання різних технологій Big Data, обрано оптимальні для розроблення ефективної інформаційно-аналітичної системи для дослідження ринку нерухомості.
2. Визначено функціональні та нефункціональні вимоги до програмної системи.
3. Проведено опитування користувачів для визначення їх потреб та очікувань.
4. Розроблено загальну схему інформаційно-аналітичної системи для дослідження ринку нерухомості.
5. Визначено компоненти програмної системи та їх взаємозв'язки.
6. Обрано відповідні технології для реалізації кожного модуля інформаційно-аналітичної системи для дослідження ринку нерухомості.
7. Розроблено прототип інформаційно-аналітичної системи для дослідження ринку нерухомості, що демонструє її основні функціональні можливості.
8. Проведено тестування програмної системи для перевірки її працездатності, ефективності та безпеки.
9. Здійснено оцінювання програмної системи в контексті відповідності її вимогам.

У рамках дослідження вдалося розробити та вдосконалити програмне забезпечення для дослідження ринку нерухомості з комплексним аналізом великих обсягів даних [7], що дозволило забезпечити ефективну роботу з цими даними та отримати цінні інсайти для прийняття управлінських та стратегічних рішень.

Анотація. *Досягнення в цифрових технологіях призвели до експоненціального зростання обсягів даних, які потребують обробки та аналізу. В дослідницькій роботі здійснено ґрунтовний аналіз існуючих систем комплексного аналізу даних з використання технологій штучного інтелекту. Розроблення програмної системи для комплексного аналізу даних ринку нерухомості з використанням технологій Big Data відповідає актуальним потребам сучасного світу та є важливим етапом у розвитку інформаційного*

суспільства. У рамках дослідження здійснено ґрунтовне проєктування програмного застосунку та розроблено модулі інформаційно-аналітичної системи підтримки продажу автомобілів. Актуальність теми розроблення архітектури програмної системи комплексного аналізу даних ринку нерухомості з використанням технологій Big Data очевидна в сучасному інформаційному суспільстві.

Ключові слова: інтелектуальна система, аналіз даних, математичне моделювання, ринок нерухомості, дослідницьке навчання, варіативні моделі, комп'ютерне моделювання, КОМСДН, Big Data, AI, CleverCOMSRL.

Література

1. Hrybiuk, O., Kant, G.S. CleverCOMSRL: Implementation of an AI Computer-Aided Design System in the Context of the Cognitive Science Paradigm for the Research Training Process. In: Machado, J., et al. *Innovations in Mechatronics Engineering III. icieng 2024. Lecture Notes in Mechanical Engineering*. Springer, Cham. 2024. P. 351-362. https://doi.org/10.1007/978-3-031-61575-7_32.
2. Hrybiuk Olena. Engineering in Educational Institutions: Standards for Arduino Robots as an Opportunity to Occupy an Important Niche in Educational Robotics in the Context of Manufacturing 4.0, in: *Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*. Volume 27-32, 2020. P. 770-785.
3. Hrybiuk, O. Improvement of the educational process by the creation of centers for intellectual development and scientific and technical creativity. In: Hamrol, A., Kujawińska, A., Barraza, M.F.S. (eds.) *MANUFACTURING 2019. LNME*, 2019. P. 370–382. Springer, Cham. https://doi.org/10.1007/978-3-030-18789-7_31.
4. Hrybiuk O. Problems of expert evaluation in terms of the use of variative models of a computer-oriented learning environment of mathematical and natural science disciplines in schools, [w:] *Zeszyty Naukowe Politechniki Poznańskiej. Seria: Organizacja i Zarządzanie*, Zeszyt Nr 79, Poznań: Wydawnictwo Politechniki Poznańskiej (WPP), 2019. P. 101-119.
5. Гриб'юк О.О. Комп'ютерне моделювання та робототехніка в навчально-виховному процесі сучасного навчального закладу. // *Сьома міжнародна науково-практична конференція FOSS Lviv-2017: Збірник наукових праць*, 27-30 квітня 2017 року, м. Львів. С. 38-43. ISBN 978-966-2598-86-5.
6. Hrybiuk, O.O. Research learning of the natural science and mathematics cycle using computer-oriented methodological systems. Monograph. Drahomanov NPU, Kyiv, 2019. P. 277–325.
7. Khang A., Abdullayev V., Hrybiuk O., & Shukla A.K. (Eds.). *Computer Vision and AI-Integrated IoT Technologies in the Medical Ecosystem* (1st ed.). CRC Press. 2024. <https://doi.org/10.1201/9781003429609>.

Сіленко Д. О. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О
*Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-
технічний університет імені академіка Юрія Бугая» (Київ)*

КОНЦЕПЦІЯ МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ МОНІТОРИНГУ СТАНУ ЗДОРОВ'Я ЛЮДИНИ

Актуальність дослідження. Сучасні тенденції розвитку інформаційних технологій демонструють стрімке зростання інтересу до систем, що забезпечують моніторинг та аналіз стану здоров'я людини. У ХХІ столітті цифровізація стала глобальним процесом, який охопив практично всі сфери життєдіяльності — від освіти та економіки до медицини. Медична галузь нині переживає етап активної трансформації, коли традиційні методи діагностики та спостереження поступово доповнюються або навіть замінюються інноваційними технологічними рішеннями [1].

Дотепер затребувані мобільні застосунки, портативні сенсори та носимі пристрої, що дозволяють відстежувати фізичну активність, частоту серцевих скорочень, артеріальний тиск, якість сну та інші параметри.

Подібні технології вже стали невід'ємною частиною щоденного життя мільйонів користувачів у всьому світі. Вони перетворюються з допоміжних інструментів на ефективні системи, здатні підтримувати здоровий спосіб життя, формувати у людей корисні звички та підвищувати рівень їхньої обізнаності про власний організм. Разом із тим, багато сучасних рішень залишаються орієнтованими лише на збір і поверхнєве представлення даних. Вони не завжди пропонують зручні механізми для глибшого аналізу чи прогнозування стану здоров'я у динаміці [2].

У цьому контексті виникає потреба у створенні повноцінної інформаційної системи моніторингу, яка б поєднувала функції збору, обробки, зберігання, аналізу та візуалізації даних у єдиному програмному середовищі.

Мета дослідження полягає у розробленні інформаційної системи для здійснення комплексного моніторингу стану здоров'я людини. У ході дослідження було вивчено актуальне питання вибору архітектури, інструментів та алгоритмів для побудови інформаційної системи для здійснення комплексного моніторингу стану здоров'я людини. Інформаційна система моніторингу здоров'я людини повинна функціонувати як комплексне середовище, що охоплює усі етапи роботи з даними. Йдеться про інтеграцію різних джерел інформації - від носимих пристроїв і мобільних сенсорів до ручного введення користувачем параметрів. Такий підхід забезпечує повноту даних та дозволяє враховувати індивідуальні особливості організму.

Важливою складовою є можливість аналізу отриманих результатів. Система повинна не лише зберігати історію показників, але й надавати засоби для порівняння значень у різних часових інтервалах. Користувач має змогу відслідковувати тенденції: наприклад, покращення фізичної активності, стабілізацію артеріального тиску чи погіршення якості сну. Це дозволяє своєчасно реагувати на відхилення та ухвалювати більш обґрунтовані рішення щодо підтримки здоров'я.

Концептуальні засади: збір даних: читання сну/активності/харчування/води з системних API (Android Health Connect, iOS HealthKit) та через ручне логування; BLE-сенсори. Нормалізація: уніфікація форматів у внутрішні об'єкти (сон, активність, прийоми їжі, вода, метрики тіла) з ідемпотентним злиттям записів з кількох джерел. Зберігання: реляційна БД для сутностей і time-series-шар для ефективною аналітики (денні/тижневі агрегати). Аналітика та інсайти: правила й статистичні детектори. Нагадування: планувальник локальних/пуш-сповіщень за подіями та вікнами часу, спираючись на поведінкові моделі (Fogg, COM-B).

Прозорість і контроль: налаштування дозволів і джерел, експорт/видалення даних, політика приватності. Логічна модель: User - акаунт, цілі; зв'язки з усіма ресурсами. Source - Health Connect/HealthKit/Manual/BLE. SleepSession; ActivitySummary; Meal; HydrationEntry; BodyMetrics; Insight;

Consent; Export Job. Архітектурна концепція (рис. 1). Клієнти: мобільні застосунки (Android/iOS) з інтеграціями Health Connect/HealthKit і ручним логуванням. Бекенд (ядро): PHP (Laravel/Symfony) з REST-CRUD API: /auth/*, /users, /sources, /sleep-sessions, /activities, /meals, /hydration, /body-metrics, /insights (read-only), /exports. Зберігання: PostgreSQL (+ TimescaleDB для часових рядів), Redis для кешу; матеріалізовані/континуальні агрегати для графіків. Безпека: TLS; токен-автентифікація (Sanctum/OAuth2/OIDC); мінімізація персональних даних; аудит доступів

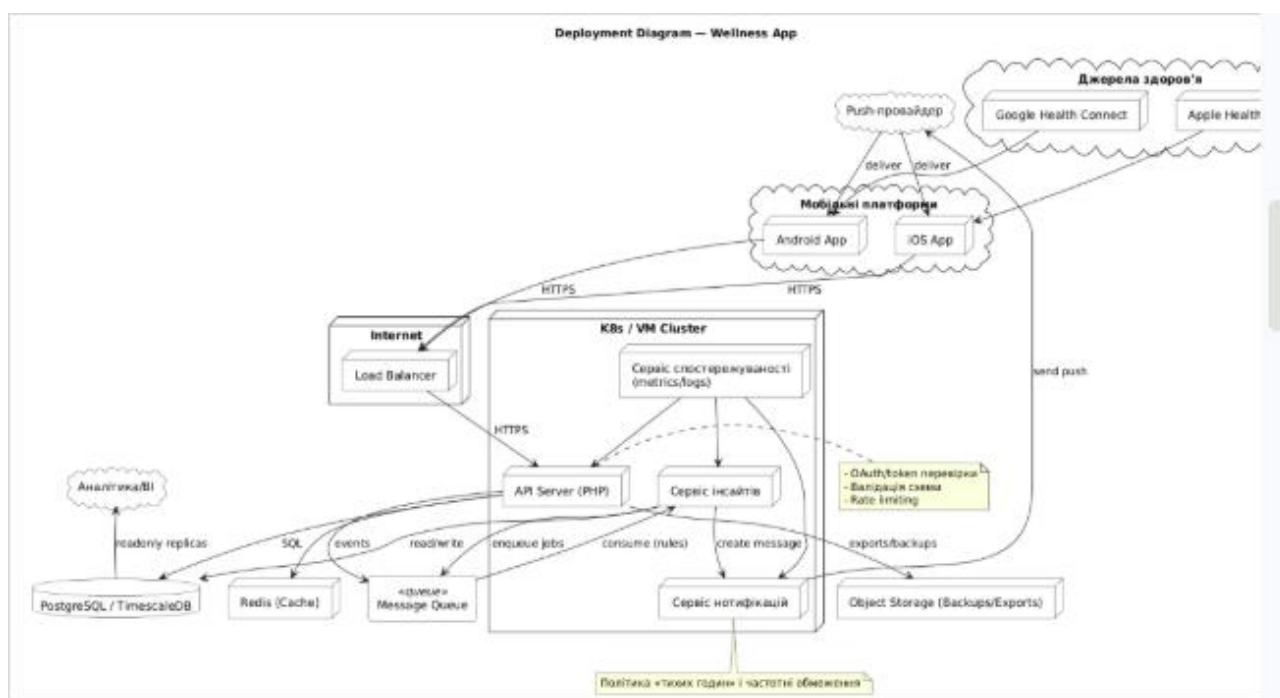


Рис. 1. Фрагмент ескізного проекту інформаційної системи для здійснення комплексного моніторингу стану здоров'я людини
Джерело: опрацювання власне

Ризики та способи їх зняття. Доступність/повнота даних: ідемпотентний імпорт, злиття часткових записів, позначення джерел. Політики магазинів: правильні декларації типів і чіткі екрани згод. Перехід у медичну сферу: виважені формулювання призначення; виключити діагноз/лікування з текстів інтерфейсу. Приватність: мінімізація даних, локальний режим, чітка політика приватності. Програмна система будується за клієнт-серверною архітектурою (рис. 2): клієнт: мобільний застосунок (Android/iOS), який інтегрується з Health Connect/HealthKit та надає інтерфейс для ручного введення даних [3]. Сервер

(бекенд): REST-API на PHP (Laravel або Symfony) з підтримкою CRUD-операцій і токен-автентифікації. Сховище даних: PostgreSQL з розширенням TimescaleDB для часових рядів, Redis для кешування, матеріалізовані агрегати для швидкого доступу до трендів

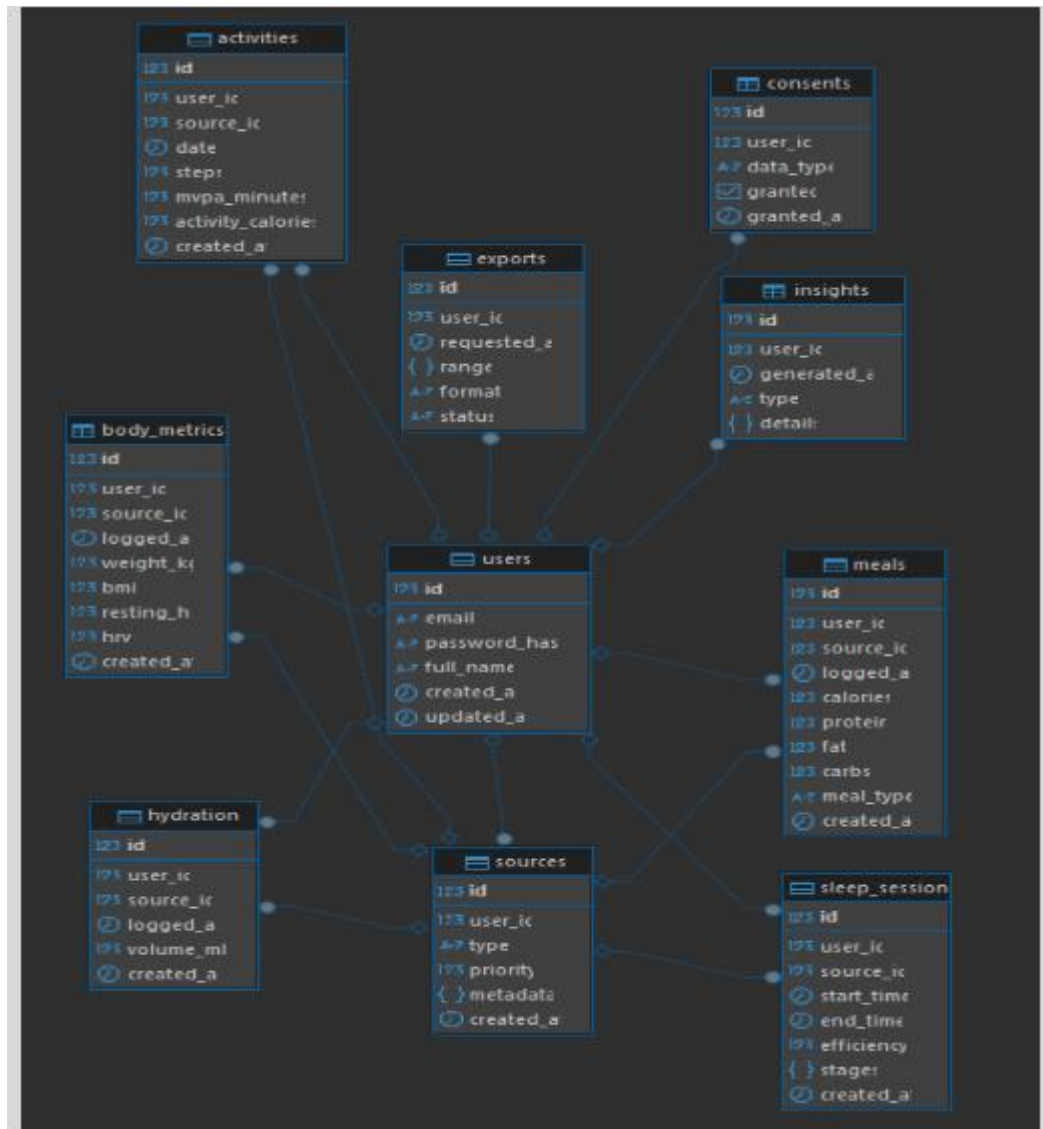


Рис. 2. Фрагмент ескізного проєкту АМ ІС для здійснення комплексного моніторингу стану здоров'я людини
Джерело: опрацювання власне

В основі покладено реляційну БД з варіативними моделями [4]: users; sources (Health Connect, HealthKit, manual, BLE); sleep_session (початок, кінець, ефективність, стадії); activities (кроки, хвилини MVPA, калорії); meals (час, калорії, білки, жири, вуглеводи, тип прийому); hydration (час, об'єм); body_metrics (вага, пульс у спокої, HRV); insights; consents; exports.

Анотація. Розроблення інформаційної системи моніторингу стану здоров'я людини є важливим кроком у напрямі цифрової трансформації медицини та охорони здоров'я. Такі системи дозволяють ефективно збирати й обробляти дані, формувати наочну картину фізичного стану, забезпечувати своєчасне реагування на відхилення та мотивувати людей до дотримання здорового способу життя. ІС призначена для немедичного моніторингу добробуту (wellness) користувача: сну, фізичної активності, харчування та гідратації, з можливістю формування пояснюваних інсайтів і мікро-рекомендацій. Система не виконує діагностики чи лікування і не надає медичних тверджень; її цінність полягає у щоденній підтримці здорових звичок. Їхня практична користь проявляється як на індивідуальному рівні - у щоденному самоконтролі та профілактиці, так і на суспільному - у зменшенні навантаження на медичні заклади та підвищенні якості надання медичної допомоги. Таким чином, створення подібних рішень відповідає сучасним викликам і відкриває перспективи для подальшого розвитку галузі цифрової медицини.

Ключові слова: CleverCOMSRL, дослідницьке навчання, варіативні моделі, математичне моделювання, КОМСДН, інформаційна система, інтелектуальний аналіз даних, моніторинг, AI.

Література

1. Hrybiuk, O., Kant, G.S. CleverCOMSRL: Implementation of an AI Computer-Aided Design System in the Context of the Cognitive Science Paradigm for the Research Training Process. In: Machado, J., et al. *Innovations in Mechatronics Engineering III*. *icieng* 2024. *Lecture Notes in Mechanical Engineering*. Springer, Cham. 2024. P. 351-362. https://doi.org/10.1007/978-3-031-61575-7_32.
2. Hrybiuk, O. Improvement of the educational process by the creation of centers for intellectual development and scientific and technical creativity. In: Hamrol, A., Kujawińska, A., Barraza, M.F.S. (eds.) *MANUFACTURING 2019*. LNME, 2019. P. 370–382. Springer, Cham. https://doi.org/10.1007/978-3-030-18789-7_31.
3. Hrybiuk, O.O. Research learning of the natural science and mathematics cycle using computer-oriented methodological systems. Monograph. Drahomanov NPU, Kyiv, 2019. P. 277–325.
4. Khang A., Abdullayev V., Hrybiuk O., & Shukla A.K. (Eds.). *Computer Vision and AI-Integrated IoT Technologies in the Medical Ecosystem* (1st ed.). CRC Press. 2024. <https://doi.org/10.1201/9781003429609>.

Ст. лаборант Степура І. В.

*Інститут психології ім. Г.С. Костюка НАПН України, лабораторія
когнітивної психології*

МЕТОДИКИ МАТЕМАТИЧНОГО ТА НЕЙРОМЕРЕЖЕВОГО АНАЛІЗУ ПСИХОФІЗІОЛОГІЧНИХ ДАНИХ

Сучасні методи штучного інтелекту (Ш І) вже майже повністю асоціюються з нейромережевими технологіями та перетворились на

меркетинговий хід їхніх виробників. Але не варто забувати і про існування символічного підходу до ШІ (логіка, алгебри, експертні системи, статистика тощо). Разом з тим, в 1960–70-х рр. в Україні широко розвивався квазібіологічний (біонічний) напрямок до проблем кібернетики (Амосов М.М.) з моделюванням систем та функцій організму. Вчені цього напрямку протистоять певною мірою розробкам в межах символічного підходу до ШІ. [1;4;7]. Один з методів квазібіологічного напрямку і є нейромережі. Про їх застосування в біомедичній галузі див. [3]. Нейромережі використовуються для розв'язання складних проблем таких як кластеризація чи розпізнавання об'єктів. Відділ психофізіології Інституту психології на чолі з А.М.Карпукіною в 1970–80 рр. тематично був спрямований на аналіз функціонування оператора авіакосмічної галузі в складних людино-технічних контурах, але переносив такі методи й на водіїв (в відділі був автотренажер з оптомеханічним проєктуваннями дорожньої ситуації та фізичними органами керування легковика), і на оцінювання психофізіологічного стану військових, школярів і студентів.

В 1970-ті рр. В.Й. Цапом [2] під керівництвом А.М. Карпукіної розроблювалася система оцінювання психофізіологічного стану операторів за допомоги самостійно зробленої фізичної (апаратної) мережі на 70–100 вузлів перцептронного типу (цифрового обчислювача не було; а промислові нейропроцесори ще не з'явилися – Mark, Odyssey, Anza, це 1982–). Математичні основи нейромережевої технології можна подивитися в [1, 75–63, 87–110] та в скороченому вигляді в [3]. Оцінювалися (на вході) параметри серцевої діяльності (частота пульсу), шкіряно-гальванічні реакції, темп дихання, температура тіла. Використовувалась властивість перцептрону до кваліфікаційного розбиття простору показників, що дозволяє віднести того чи іншого досліджуваного до групи «успішних» чи «неуспішних» операторів за фізіологічною ціною організації діяльності, стресостійкості тощо, який з'являвся на виході мережі. Фізична рівність будови нейромережі в шарі прийняття рішення (аналізу входів) збігається з гіпотезою Карпукіної про

рівневість та ієрархічність психофізіологічних станів людини. Окремою системою оцінювання був метод Р.М. Баєвського – математичний аналіз серцевого ритму, вперше використаний ще під час польотів на космічних кораблях «Восток» і «Восход», та згодом і на «МКС» [5]. Р. М. Баєвський був всесвітньо знаним фахівцем в галузі космічної кардіології та з методик фізіологічних вимірів в авіації та космонавтиці. Також використовувалися психологічні методики оцінювання станів людини. Разом з тим, від біонічного підходу для прогнозування успішності діяльності людини на початку 1980-х рр. відійшли, віддаючи перевагу регресійному аналізу. Останній використовується для аналізу рядів прикладів входу/виходу досліджуваної системи показників, з метою знайти неперервну функцію, на основі якої можна було б прогнозувати вихід.

Цікаво зауважити, що проблематика оцінювання психофізіологічних параметрів апаратурною нейромережею виявилася дещо подібною до задачі орієнтації безпілотних малих електрокарів (ТАІР, Малюк тощо; Ін-т кібернетики НАНУ) та цехових транспортних роботів в просторі. Там також застосовували апаратурні нейромережі на 100 вузлів, розбиті на тематичні «ансамблі» щодо реалізації функцій орієнтації та руху машини [1, 210–242]

Проблематика діагностики психофізіологічних станів дотична до завдань реалізації процедур біологічного зворотного зв'язку (БЗЗ) [6]. В основі БЗЗ знаходиться теорія функціональних систем П.К.Анохіна, а також методи утворення умовних рефлексів оперантного типу (Н.В.Мілер, Л.В.ДіКара). Дж. Камія показав, що досліджувані можуть змінювати параметри власної енцефалограми (та інших психофізіологічних параметрів – електропровідності шкіри, частоти серцевих скорочень та дихання, температури тощо) якщо їм відомі ці показники від наявної апаратури. БЗЗ використовується для регуляції адаптивно-компенсаторних реакції при фізичних та психоемоційних навантаженнях. БЗЗ застосовується в задачах відеоспостереження за станом оператора транспортних засобів на залізничному та автомобільному транспорті

з метою попередження засинання та втрати свідомості. Також, виходячи з досвіду [2], можна додавати й інші датчики фізіологічних станів операторів.

Висновки. Історична ретроспектива досліджень квазібіологічного напрямку в кібернетиці в Україні дозволяє розкрити неочікувані зв'язки між апаратним оцінюванням психофізіологічних показників людини-оператора та розв'язання завдання орієнтації безпілотного транспортного засобу за допомогою нейромереж; історію методів БЗЗ.

Анотація. В статті прослідковуються зв'язки між історичними дослідженнями психофізіологічних станів методами апаратних нейромереж щодо операторської діяльності в системі «людина – машина» та «робот-машина» і з сучасною проблематикою біологічних зворотних зв'язків в ракурсі формальних нейронідов.

Ключові слова: нейромережа, біоніка, психофізіологічні стани, біологічний зворотний зв'язок, людина-оператор, система людина-машина

Література

1. Нейрокомп'ютери та інтелектуальні роботи / за ред.. Амосова М.М. Київ : Наукова думка, 1991. 272 с.
2. Цап В.Й. Деякі підходи використання спеціалізованих однорідних обчислювальних засобів для обробки психофізіологічної інформації. *Методичні рекомендації з вдосконалення навчально-виховного процесу*. Київ : ІІ Міносвіти УРСР, 1977, с.82–85.
3. Amato, F., López, A., Peña-Méndez, E.M., Vaňhara, P., Hampl, A. and Havel, J. Artificial Neural Networks in Medical Diagnosis. *Journal of Applied Biomedicine*, 2013, 11, 47–58. DOI 10.2478/v10136-012-0031-x URI: <https://jab.zsf.jcu.cz/pdfs/jab/2013/02/01.pdf>
4. Amosov, N. M., Karpukhina, A. M., Zaslavskii, S. Ya., Lishchuk, V. A., Ivanov-Muromskii K.A. Electrical Activity of Respiratory Center Neurons in the Bionic Aspect. *Nekotome Problemy Biokibernetiki Primenenye Elektroniki v Biglogli i Meditsine*. 1967, No 1, 34–55. Certain problems of biocybernetics, the use of electronics in biology and in medicine (collection of articles), FTD-MT-24-291-67
5. Baevsky R.M., Baranov V.M., Funtova I.I., Diedrich A., Pashenko A.V., Chernikova A.G., Drescher J., Jordan J., Tank J. Autonomic cardiovascular and respiratory control during prolonged spaceflights aboard the International Space Station. *J Appl Physiol*. 2007. Jul; 103 (1), p. 156–161. doi: 10.1152/japplphysiol.00137.2007. Epub 2007 Apr 19.
6. Biofeedback: a Practitioner's Guide / ed. by : M. Schwartz, F. Andrasik : New York: The Guilford Press, 2016. 764 p.
7. Karpukhina, A. M., Ponomareva I. D. Two varieties of functional indicators in the study of respiratory neurons. *Kibernetika i vychislitel'naya tekhnika*, no. 4, [Kiyev, Naukova dumka], 1970, p. 147–158. <https://apps.dtic.mil/sti/tr/pdf/AD0750123.pdf>

Таран В.В. (здобувач бакалаврату, 2 курс),
Чірва А.А. (здобувач бакалаврату, 2 курс),
Науковий керівник – доц Гадецька С. В.
Харківський національний автомобільно-дорожній університет

ЗАСТОСУВАННЯ ІНТЕГРАЛЬНОГО ЧИСЛЕННЯ ДЛЯ ОЦІНЮВАННЯ ІНТЕНСИВНОСТІ ТРАНСПОРТНИХ ПОТОКІВ

Транспортні системи відіграють ключову роль у функціонуванні сучасних міст. Зростання рівня автомобілізації населення, розвиток міських агломерацій та підвищення мобільності населення призводять до значного збільшення навантаження на дорожню інфраструктуру. У результаті виникають транспортні затори, знижується швидкість руху, збільшується час пересування та зростає рівень екологічного навантаження. Для ефективного управління транспортними потоками необхідно використовувати математичні методи аналізу та моделювання. Одним із важливих інструментів математичного аналізу є інтегральне числення. Застосування інтегралів дозволяє визначати сумарну кількість транспортних засобів, що проходять через певний переріз дороги протягом заданого періоду часу. Метою даної роботи є дослідження можливостей застосування інтегрального числення для оцінювання інтенсивності транспортних потоків, а також демонстрація практичного використання цього підходу на конкретних числових прикладах.

1. Основні характеристики транспортного потоку. У теорії транспортних потоків розглядають три основні параметри, що характеризують рух транспортних засобів: інтенсивність транспортного потоку; щільність потоку; середню швидкість руху.

Інтенсивність транспортного потоку визначається як кількість транспортних засобів, що проходять через певний переріз дороги за одиницю часу:

$$q = N / T,$$

де N – кількість транспортних засобів, T – час спостереження.

Щільність транспортного потоку показує кількість автомобілів, що знаходяться на одиниці довжини дороги. Середня швидкість характеризує середню швидкість руху транспортних засобів у потоці. Ці параметри пов'язані фундаментальним співвідношенням:

$$q = k \cdot v,$$

де k – щільність потоку, v – середня швидкість руху.

2. Використання інтегрального числення. У реальних умовах інтенсивність транспортного потоку не є сталою величиною. Вона змінюється протягом часу залежно від різних факторів: години доби; погодних умов; роботи світлофорів; дорожніх пригод; особливостей міської інфраструктури. Тому інтенсивність доцільно розглядати як функцію часу $q(t)$.

Загальна кількість транспортних засобів, що проходять через певний переріз дороги за інтервал часу $[t_1, t_2]$, визначається за допомогою визначеного інтеграла:

$$N = \int q(t) dt$$

Ця формула дозволяє обчислювати сумарний транспортний потік навіть у випадку, коли інтенсивність руху змінюється протягом часу.

Приклад 1 (ранковий пік). Нехай інтенсивність транспортного потоку змінюється за законом: $q(t) = 800 + 120t$, де t – час у годинах. Знайдемо кількість автомобілів, що проходять через переріз дороги за 2 години.

$$N = \int (800 + 120t) dt = 1840 \text{ (автомобілів)}.$$

Отже, за дві години через досліджувану ділянку проходить приблизно 1840 транспортних засобів.

Приклад 2 (добовий потік). Розглянемо інший приклад, коли інтенсивність руху описується функцією $q(t) = 400 + 50t$ для інтервалу часу від 0 до 10 годин. Тоді загальна кількість автомобілів визначається інтегралом

$$N = \int (400 + 50t) dt = 6500 \text{ (автомобілів)}.$$

Цей результат дозволяє оцінити завантаженість дорожньої ділянки.

Математичні моделі транспортних потоків широко використовуються у транспортному плануванні. Основні напрями застосування: планування

транспортної інфраструктури; оптимізація роботи світлофорів; прогнозування транспортних заторів; розробка інтелектуальних транспортних систем. Крім того, використання математичних моделей, зокрема моделі LWR, дозволяє описувати складні процеси утворення транспортних заторів та динаміку транспортних потоків.

***Анотація.** У роботі досліджено застосування інтегрального числення для оцінювання інтенсивності транспортних потоків. Показано, що інтегральні методи дозволяють визначати сумарну кількість транспортних засобів при змінній інтенсивності руху, а також використовуються при плануванні транспортних систем та оптимізації дорожнього руху у сучасних містах.*

***Ключові слова:** інтегральне числення, інтенсивність транспортних потоків, транспортні затори.*

Література

1. Абрамова Л. С. Методи визначення складу транспортного потоку на автомобільних дорогах // Вісник НТУ «ХПІ». – Харків, 2017. URL: <https://repository.kpi.kharkov.ua/items/6bfd17c0-95bc-4fa2-9b73-6b35b4908abd>.
2. Transportation Research Board. Traffic Flow Theory : A State-of-the-Art Report. – Washington : TRB, 2016. URL: <https://onlinepubs.trb.org/onlinepubs/sr/sr165/165.pdf>.

УДК 004.942:[327

Турушева М. В. (здобувач бакалаврату, 3 курс),
Науковий керівник – доц. Гриб'юк О. О
Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-технічний університет імені академіка Юрія Бугая» (Київ)

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ІГРОВОГО ЗАСТОСУНКУ ЖАНРУ СПОРТИВНОГО СИМУЛЯТОРА ФУТБОЛУ

Сучасні комп'ютерні ігри невпинно еволюціонують, об'єднуючи реалістичну фізику, графіку та алгоритми штучного інтелекту. Серед них гра «Футбол» залишається одним із найпопулярніших спортивних симуляторів, оскільки вона пропонує можливість відчувати дух змагань, тренувати реакцію, уважність і розвивати тактичне мислення. Основний принцип гри полягає у суперництві двох команд, які намагаються забити м'яч у ворота опонентів.

Мета дослідження полягає у розробленні ігрового застосунку в жанрі спортивного симулятора футболу з підтримкою мультиплеєру та адаптивною системою управління. *Завдання:* 1. Здійснити ґрунтовний аналіз інформаційного та програмного забезпечення, існуючих застосунків, виокремивши переваги та недоліки систем; добір технологій та інструментальних засобів для розроблення ігрової програмної системи в жанрі спортивного симулятора футболу. 2. Розробити концепцію власного програмного продукту з урахуванням сучасних вимог до інтерфейсу та зручності користування. 3. Здійснити проєктування архітектури ігрової програмної системи в жанрі спортивного симулятора футболу. 4. Реалізувати модулі ігрової програмної системи в жанрі спортивного симулятора футболу. 5. Забезпечити перевірку коректності функціонування модулів ігрової програмної системи.

Основним завданням на етапі практичної реалізації кваліфікаційної роботи стала повна реалізація візуалу гри задля уникнення спрощення прототипу. Відмова від статичних зображень на користь ігрової системи двовимірної анімації вимагала розробки детального ескізного проєкту, який включає графічну документацію, прототипи інтерфейсів, UML-діаграми та масиви спрайтів. Ігрові персонажі були поділені на дві команди: команду білих та команду рудих котів.

Для забезпечення візуальної різноманітності було розроблено систему шкінів, щоб користувач мав змогу вибирати зовнішній вигляд команди. Для кожної з двох команд створено базовий вигляд персонажа, а також три додаткові варіанти футболок: жовта, червона та блакитна. Для забезпечення ілюзії динамічного руху, для кожного з восьми унікальних виглядів було створено набори покадрових спрайтів Sprite Sheets (див. Додаток Б). Логіка анімації побудована на чергуванні двох станів: стані спокою *Idle*, що імітує дихання персонажа за допомогою двох кадрів, та стані руху *Run*, що відображає фази бігу також за допомогою двох графічних кадрів. Увесь створений масив графічних ресурсів було імпортовано в середовище розробки Unity із

застосуванням фільтрації текстур Point (no filter), що є обов'язковою вимогою для збереження стилістики Pixel Art та уникнення розмиття країв.

Процес програмної інтеграції цієї графіки здійснювався через вбудовану підсистему Unity Animator, відому як Mecanim [1]. Для кожного скіна було згенеровано власний контролер анімації *Runtime Animator Controller*. У графі станів *State Machine* налаштовано логічні переходи між вузлами Idle та Run (рис. 1), що керуються булевою змінною *isRunning*. Для забезпечення чуйності управління, особливу увагу було приділено налаштуванням переходів *Transitions* між вузлами анімаційного графа.

У параметрах переходу від стану очікування до стану бігу, а також у зворотному напрямку, було примусово вимкнено базову опцію *Has Exit Time*. Це дозволило системі перемикаати анімації миттєво в момент зміни значення булевого параметра, не чекаючи завершення поточного циклу відтворення попередніх кадрів. Крім того, параметр *Transition Duration* було встановлено на нульову позначку. Таке архітектурне рішення є важливим для обраної стилістики *Pixel Art*, оскільки воно цілком запобігає автоматичному плавному змішуванню кадрів внутрішніми алгоритмами рушія, зберігаючи необхідну різкість, динаміку та піксельну точність рухів ігрових персонажів.

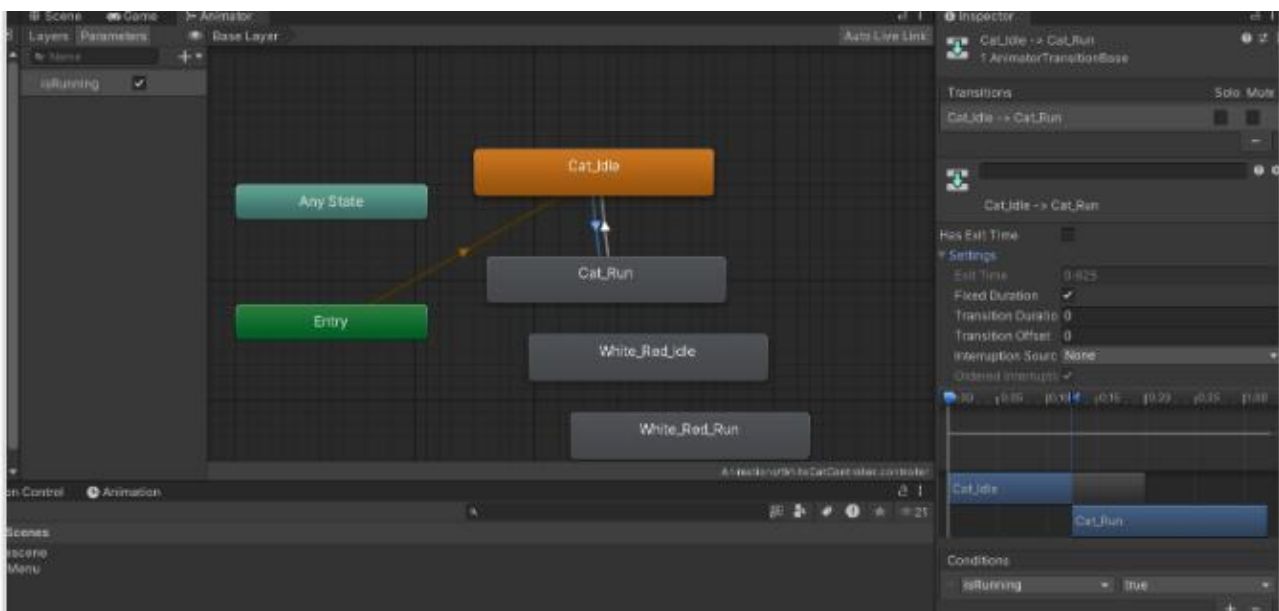


Рис. 1. Контролер анімації Runtime Animator Controller із графом станів
Джерело: опрацювання власне

Враховуючи специфіку двокадрової анімації, критично важливим етапом стало ручне налаштування параметра *Sample Rate* в інспекторі Unity. Зниження частоти вибірки дозволило забезпечити природний ритм зміни кадрів, повністю усунувши ефект миготіння текстур, характерного для неоптимізованих 2D-ігор. Зміна станів синхронізована з фізичним рухом у класі контролера: щойно модуль вектора швидкості об'єкта перевищує мінімальний поріг, параметр *isRunning* отримує значення *true*, і графіка персонажа динамічно адаптується до його фізичного стану [2].

Управління усім масивом графічних даних реалізовано через гнучку архітектуру класів *TeamSkinManager* та *SkinApplier* (рис. 2). Глобальні налаштування кольору футболок задаються користувачем у графічному інтерфейсі та надійно зберігаються у системний кеш гри за допомогою інструменту *PlayerPrefs*. Під час ініціалізації ігрової сцени скрипт *SkinApplier* здійснює сканування всіх ігрових об'єктів на полі, визначає їхню приналежність до певної команди та зчитує обраний користувачем скін. На основі цих параметрів скрипт динамічно підміняє анімаційний контролер безпосередньо в компоненті *Animator* кожного кота. Таке рішення дозволило використовувати на сцені лише один базовий префаб гравця, суттєво зекономивши оперативну пам'ять проєкту та уникнувши дублювання об'єктів в ієрархії.

```
if (player.teamName == "A")
{
    // Об'являємо відповідний контролер з анімаціями
    if (teamASkin == "Red" && a_Red != null) anim.runtimeAnimatorController = a_Red;
    else if (teamASkin == "Blue" && a_Blue != null) anim.runtimeAnimatorController = a_Blue;
    else if (teamASkin == "Yellow" && a_Yellow != null) anim.runtimeAnimatorController = a_Yellow;
    else if (a_Default != null) anim.runtimeAnimatorController = a_Default;
}
else if (player.teamName == "B")
{
    if (teamBSkin == "Red" && b_Red != null) anim.runtimeAnimatorController = b_Red;
    else if (teamBSkin == "Blue" && b_Blue != null) anim.runtimeAnimatorController = b_Blue;
    else if (teamBSkin == "Yellow" && b_Yellow != null) anim.runtimeAnimatorController = b_Yellow;
    else if (b_Default != null) anim.runtimeAnimatorController = b_Default;
}
```

Рис. 2. Фрагмент програмної реалізації ігрової системи
Джерело: опрацювання власне

Для вирішення поширеної проблеми 2D-симуляторів, коли комп'ютерні гравці збиваються в одну купку, впроваджено алгоритм просторового

розділення *Separation* та метод *CalculateSeparation()*, який в реальному часі вимірює векторні відстані між усіма об'єктами. Якщо дистанція стає меншою за заданий параметр *personalSpace* 1.3 юніта (це приблизний радіус «особистого простору» персонажа в Unity), алгоритм генерує нормалізований вектор відштовхування, який математично додається до основного вектора швидкості об'єкта. Це створює ілюзію органічної, продуманої командної взаємодії та запобігає накладанню спрайтів один на одного.

Особлива складність полягала у проєктуванні алгоритму атаки та прийняття рішень *AI Decide Action*. Важливим моментом було, щоб боти не реагували миттєво, як машини, а діяли як живі люди. Тому, коли бот отримує м'яч, було додано невелику затримку реакції *reactionDelay*, під час якої агент продовжує рух по інерції. Якщо бот приймає рішення здійснити удар по воротах, вмикається логіка прицілювання. Замість неточного удару в центр зони, алгоритм сканує сцену на наявність ворожого воротаря та зчитує його поточну координату Y. Шляхом порівняння цієї координати з координатами країв воріт, штучний інтелект визначає найбільш вразливу зону вільного кута. Розрахувавши цільову точку в порожньому куті, алгоритм додає до неї випадкову похибку через *Random.Range*. Це потрібно для того, щоб був певний відсоток промахів. Завдяки цьому бот грає небезпечно, але іноді робить помилки, як і реальний гравець. Фінальним етапом було впровадження звукового супроводу та глобального менеджменту станів. Вся акустична логіка інкапсульована в окремому глобальному менеджері *AudioManager*.

Використання властивості *DontDestroyOnLoad* дозволило менеджеру існувати безперервно під час переходів між сценою головного меню та безпосередньо ігровою сесією, забезпечуючи плавне відтворення фонові музики без обривів [3]. Система чітко розрізняє два типи аудіоданих: зациклену музику *BackGround Music* та одноразові просторові ефекти *Sound Effects*. Логіка функціонування даної підсистеми та переходи між режимами відтворення візуалізовані нижче на діаграмі станів звуку, що дозволяє чітко простежити моменти перехоплення сигналу для відтворення ігрових подій.

Завдяки цій системі у грі з'явилися звуки відбивання м'яча, гучний аудіальний сигнал при фіксації голу та музична тема, що супроводжує появу панелі перемоги. Використання методу *PlayOneShot* для звукових ефектів дозволило системі накладати аудіокліпи один на одного без переривання (наприклад, під час швидкого пасування м'яча), що формує насичену та реалістичну акустичну картину [4]. Ще одним важливим елементом користувацького інтерфейсу (UI) та глобального менеджменту стала розробка підсистеми збору та виведення ігрової статистики матчу [5]. Для її реалізації було розширено архітектуру глобального класу *GameManager*, який також побудований за патерном Singleton *public static GameManager Instance*, що забезпечує доступ до його методів без необхідності ресурсоємного пошуку об'єктів на сцені [6]. У цьому класі ініціалізовано цілочисельні змінні для підрахунку кількості ударів по воротах та успішних пасів для обох команд.

Логіка цих подій була інтегрована у клас *PlayerController*. Для керованого користувачем персонажа тригери виклику методів *AddShot* та *AddPass* спрацьовують у блоці обробки натискань клавіш вводу. Для комп'ютерних суперників ці тригери вбудовані безпосередньо у скінченний автомат у методі *AI DecideAction*: система фіксує події у моменти, коли алгоритм приймає математично обґрунтоване рішення здійснити удар по воротах або передати м'яч вільному союзнику. Зібрані аналітичні дані виводяться на екран у фінальній панелі статистики по завершенню матчу, забезпечуючи гравця інформацією про результати обох команд. Під час перезапуску матчу метод *RestartMatch* усі лічильники обнуляються для гарантування чистоти даних [7]. На завершальному етапі розробки було проведено комплексне функціональне тестування програмного продукту. В основу верифікації покладено поєднання методів «білого» та «чорного ящика», що дозволило оцінити якість коду та відповідність гри початковим вимогам.

Анотація. Актуальність дослідження полягає у розвитку 2D-гри «Футбол» через інтеграцію нових функціональних можливостей, спрямованих на підвищення реалістичності та динамічності ігрового процесу. Одним із ключових напрямів є впровадження ботів із елементами штучного інтелекту,

здатних автономно взаємодіяти з м'ячем, виконувати передачі, здійснювати захист воріт та організовувати атаки. Додатково, значну увагу слід приділити розробці анімації рухів гравців, що сприятиме створенню більш плавної візуальної складової гри. У ході дослідження було здійснено ґрунтовний аналіз інформаційного та програмного забезпечення в жанрі спортивного симулятора футболу, існуючих застосунків, виокремивши переваги та недоліки систем, зокрема примітивну логіку комп'ютерних опонентів та низьку якість візуального зворотного зв'язку також добрано технології та інструментальні засоби. Було розроблено концепцію власного програмного продукту з урахуванням сучасних вимог до інтерфейсу та зручності користування, яка передбачала створення гри з тактичною складовою, орієнтованої на однокористувацький режим проти автономних ботів. Завдяки вибору ігрового рушія Unity та мови програмування C# вдалося здійснити проектування архітектури та розробити модулі ігрової системи «Футбол» на основі принципів об'єктно-орієнтованого програмування, що забезпечило міцну базу для подальшої практичної реалізації.

Ключові слова: об'єктно орієнтоване програмування, дослідницьке навчання, футбол, варіативні моделі, математичне моделювання, комп'ютерне моделювання, КОМСДН, ігрова система, спортивний симулятор, AI, C#, Unity.

Література

1. Hrybiuk Olena. Engineering in Educational Institutions: Standards for Arduino Robots as an Opportunity to Occupy an Important Niche in Educational Robotics in the Context of Manufacturing 4.0, in: *Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*. Volume 27-32, 2020. P. 770-785.
2. Hrybiuk O. Problems of expert evaluation in terms of the use of variative models of a computer-oriented learning environment of mathematical and natural science disciplines in schools, [w:] *Zeszyty Naukowe Politechniki Poznańskiej. Seria: Organizacja i Zarządzanie*, Zeszyt Nr 79, Poznań: Wydawnictwo Politechniki Poznańskiej (WPP), 2019. P. 101-119.
3. Гриб'юк О.О. Дослідницьке навчання учнів предметів природничо-математичного циклу з використанням комп'ютерно орієнтованих методичних систем / О. О. Гриб'юк. Монографія. Київ: НПУ імені М. П. Драгоманова, 2019.
4. Hrybiuk O., Vedishcheva O. Experimental Teaching of Robotics in the Context of Manufacturing 4.0: Effective Use of Modules of the Model Program of Environmental Research Teaching in the Working Process of the Centers "Clever". In: , et al. *Innovations in Mechatronics Engineering II. icieng 2022. Lecture Notes in Mechanical Engineering*. Springer, Cham. 2022.
5. Hrybiuk O. Experience in Implementing Computer-Oriented Methodological Systems of Natural Science and Mathematics Research Learning in Ukrainian Educational Institutions. In: Machado J., Soares F. (eds) *Innovations in Mechatronics Engineering. Lecture Notes in Mechanical Engineering*, 2022. P. 55-68. Springer, Cham Online.
6. Гриб'юк О.О. Комп'ютерне моделювання та робототехніка в навчально-виховному процесі сучасного навчального закладу. / Гриб'юк О.О. // *Сьома міжнародна науково-практична конференція FOSS Lviv-2017: Збірник наукових праць*, 27-30 квітня 2017 року, м. Львів. С. 38-43. ISBN 978-966-2598-86-5.
7. Hrybiuk O. Improvement of the Educational Process by the Creation of Centers for Intellectual Development and Scientific and Technical Creativity. In: Hamrol A., Kujawińska A.,

УДК 004.942:[327

Філіпенков О. В. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гриб'юк О. О
Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-технічний університет імені академіка Юрія Бугая» (Київ)

ДОСЛІДЖЕННЯ МАТЕМАТИЧНО ОБҐРУНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ АНАЛІЗУ ДІЯЛЬНОСТІ СПІВРОБІТНИКІВ КОМПАНІЇ

Актуальність дослідження. На відміну від випадкових або штучно згенерованих хаотичних наборів даних, синтетичні HR-дані мають містити структурні та поведінкові закономірності, властиві реальному кадровому середовищу, зокрема залежності між посадою, посадовим рівнем, датами прийому на роботу, рівнем заробітної плати, штрафними санкціями, бонусними коефіцієнтами, метриками ефективності, дисциплінарними показниками та іншими атрибутами [1]. Існуючі засоби генерації синтетичних даних призначені переважно для загальних сценаріїв і не орієнтовані на бізнес-середовище України. Існує потреба у створенні системи, здатної формувати не лише окремі довільні значення, а зв'язні набори даних, що імітують реальні кадрові структури та підтримують сценарії розрахунку заробітної плати, оцінювання співробітників, моделювання кар'єрного розвитку та аналізу операційних ризиків.

Мета дослідження полягає у проєктуванні та розробленні інформаційної системи для аналізу діяльності співробітників компаній, доступної для використання інженерами, аналітиками, науковцями, викладачами та розробниками інформаційних систем для забезпечення HR-аналітики реалістичними синтетичними українськими даними, що зберігаються у форматах, придатних для інтеграції у ВІ-середовища, машинне навчання та ETL-конвеєри.

У рамках дослідження розроблено інформаційну систему Основний матеріал дослідження присвячено опису підходів до побудови системи, яку реалізовано в проєкті FakeUkrBase. Система ґрунтується на модульній архітектурі, де окремі компоненти відповідають за генерацію даних, їх перевірку, перетворення, збереження та подальшу аналітичну обробку (рис. 1).

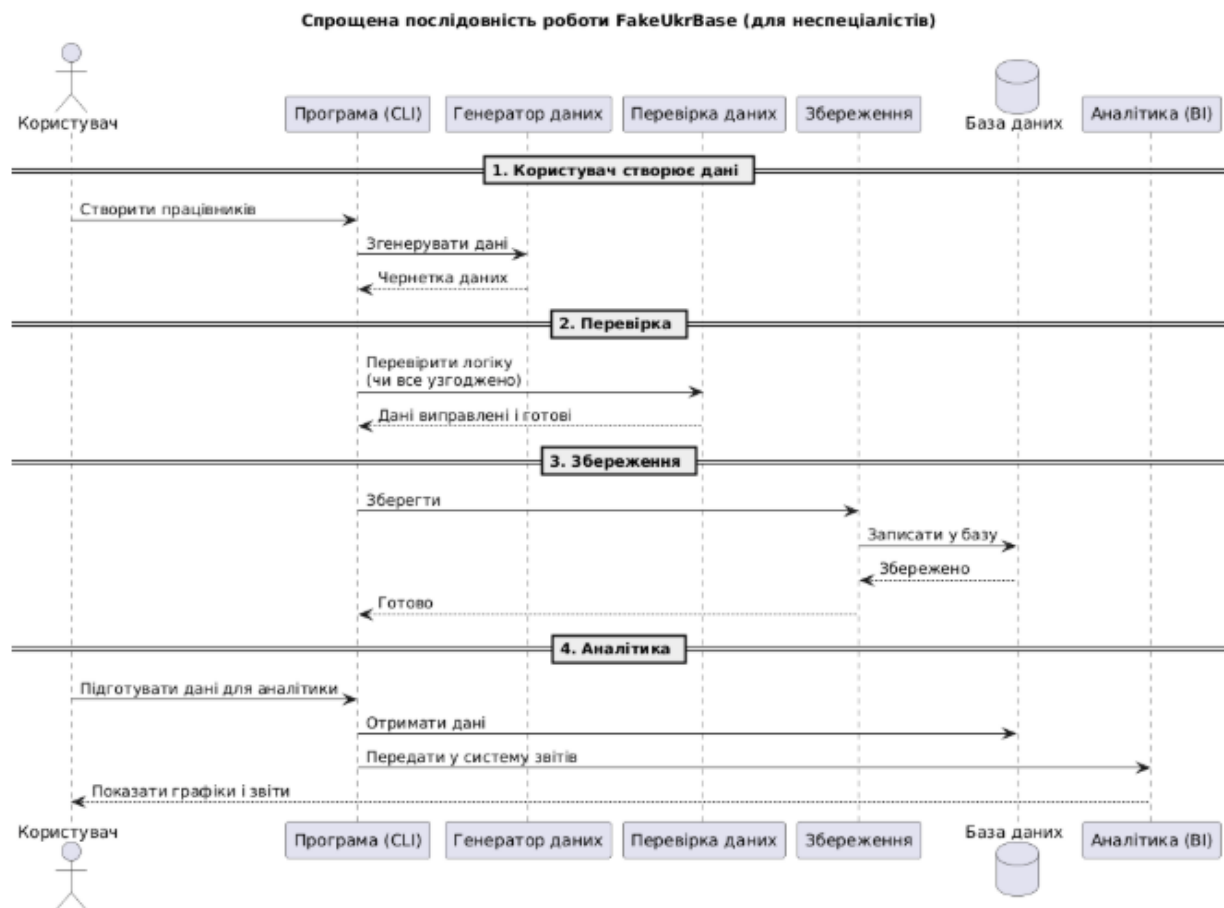


Рис. 1. Фрагмент ескізного проєкту УВМ ІС для аналізу діяльності співробітників компаній
Джерело: опрацювання власне

Важливою частиною реалізації є застосування seed-механізму відтворюваності даних, який забезпечує можливість створювати однакові набори даних за умови використання однакового початкового значення випадкової послідовності. Такий підхід дозволяє проводити експерименти, тестування та А/В сценарії з повною повторюваністю результатів.

Методологія побудови системи базується на принципах модульності, масштабованості, відтворюваності та відокремленості відповідальності. Для побудови архітектури було застосовано концепцію багат шарових систем, де кожен логічний компонент відповідає лише за одну функцію. Серед

застосованих технологій Python відіграє роль основного середовища розробки. PostgreSQL використовується як операційна реляційна база даних, а ClickHouse забезпечує можливості колоночного зберігання даних для OLAP-аналітики. Формат Parquet дає змогу ефективно зберігати дані у файлових сховищах, а Metabase дозволяє здійснювати візуалізацію аналітичних показників.

Архітектура інформаційної системи представлена наступною структурою: шар генерації даних містить модулі, що відповідають за створення окремих атрибутів працівників. Далі дані проходять через валідаційний шар, де перевіряється їх коректність, узгодженість та відповідність бізнес-правилам (рис. 2).

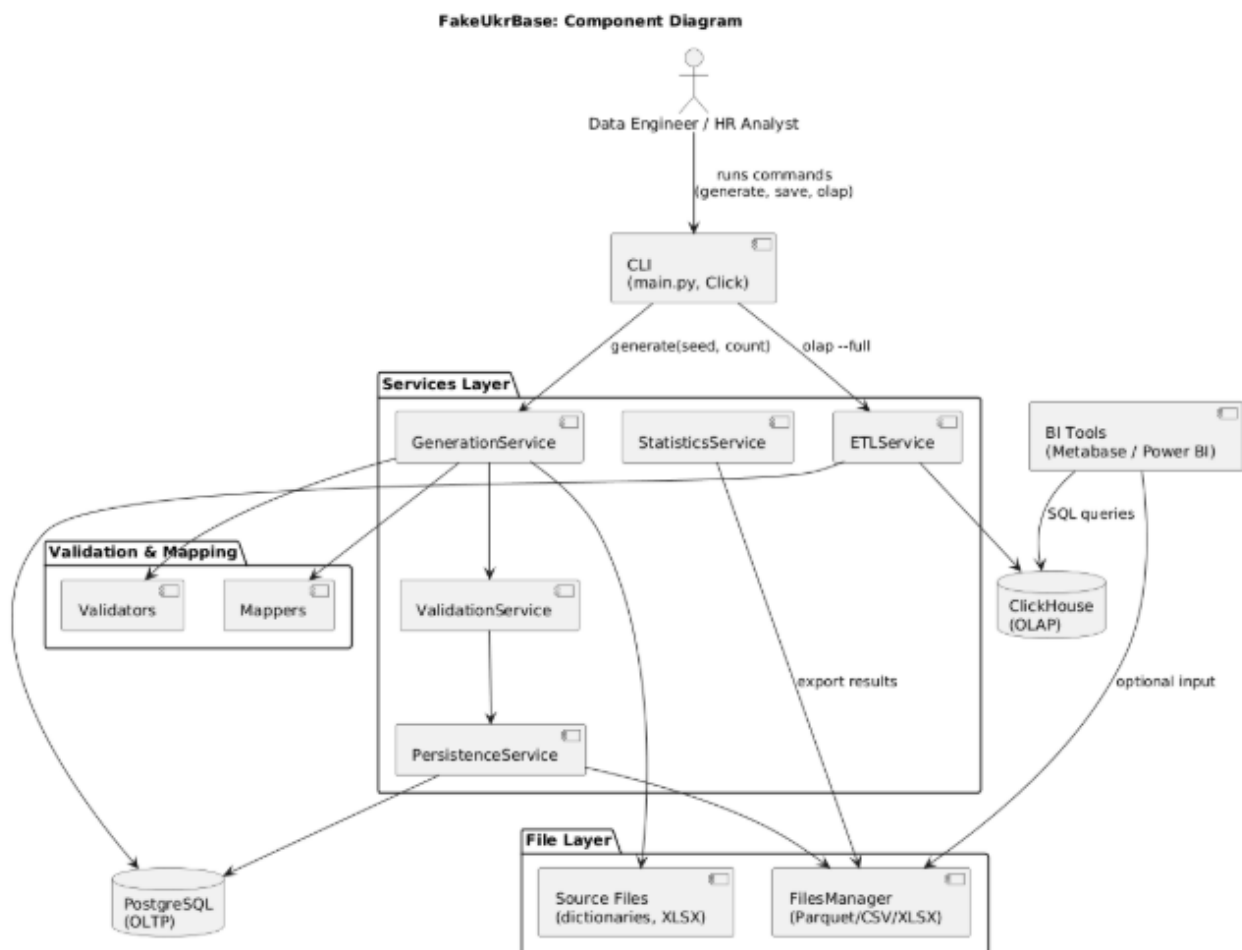


Рис. 2. Фрагмент ескізного проєкту ВМ ІС для аналізу діяльності співробітників компаній

Джерело: опрацювання власне

Наступним етапом є збереження у PostgreSQL, після чого дані перетворюються та експортуються у ClickHouse. На завершальному етапі користувач може аналізувати згенеровані дані через Metabase або інші BI-платформи. Синтетичні дані в контексті кадрової аналітики формують окремий напрям застосування інформаційних технологій, який сьогодні активно інтегрується у корпоративні структури, університетські дослідницькі процеси та розробку програмного забезпечення. З огляду на обмеження, накладені законодавчими нормами щодо персональних даних, а також враховуючи важливість моделювання сценаріїв у бізнес-середовищах, використання штучно сформованих, але структурно достовірних даних стає невід’ємною частиною процесів розроблення HR-систем, CRM-інтеграцій, автоматизованих систем управління персоналом і систем прогнозування.

У процесі тестування було проведено аналіз відтворюваності, коректності логіки залежностей, стабільності генерації великих масивів даних, швидкості експорту та можливості інтеграції з BI-середовищами (рис. 3).

	id	first_name	last_name	email	phone_number	address	city	state_province	country	postal_code	hire_date	job_id	salary	commission_pct
0	0	John	Deere	john.deere@ic.com	(555) 555-5555	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	1	12000	0.1
1	1	Jane	Smith	jane.smith@ic.com	(555) 555-5556	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	2	15000	0.15
2	2	Michael	Brown	michael.brown@ic.com	(555) 555-5557	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	3	18000	0.18
3	3	Emily	Johnson	emily.johnson@ic.com	(555) 555-5558	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	4	20000	0.2
4	4	David	Wilson	david.wilson@ic.com	(555) 555-5559	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	5	22000	0.22
5	5	Sarah	Davis	sarah.davis@ic.com	(555) 555-5560	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	6	24000	0.24
6	6	James	Miller	james.miller@ic.com	(555) 555-5561	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	7	26000	0.26
7	7	Laura	Moore	laura.moore@ic.com	(555) 555-5562	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	8	28000	0.28
8	8	Robert	Taylor	robert.taylor@ic.com	(555) 555-5563	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	9	30000	0.3
9	9	Michelle	Anderson	michelle.anderson@ic.com	(555) 555-5564	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	10	32000	0.32
10	10	Christopher	Thomas	christopher.thomas@ic.com	(555) 555-5565	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	11	34000	0.34
11	11	Amanda	White	amanda.white@ic.com	(555) 555-5566	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	12	36000	0.36
12	12	Kevin	Green	kevin.green@ic.com	(555) 555-5567	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	13	38000	0.38
13	13	Stephanie	Black	stephanie.black@ic.com	(555) 555-5568	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	14	40000	0.4
14	14	Brandon	Gray	brandon.gray@ic.com	(555) 555-5569	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	15	42000	0.42
15	15	Nicole	King	nicole.king@ic.com	(555) 555-5570	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	16	44000	0.44
16	16	Eric	Wright	eric.wright@ic.com	(555) 555-5571	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	17	46000	0.46
17	17	Christina	Scott	christina.scott@ic.com	(555) 555-5572	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	18	48000	0.48
18	18	Matthew	Young	matthew.young@ic.com	(555) 555-5573	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	19	50000	0.5
19	19	Olivia	Reed	olivia.reed@ic.com	(555) 555-5574	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	20	52000	0.52
20	20	Benjamin	Cook	benjamin.cook@ic.com	(555) 555-5575	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	21	54000	0.54
21	21	Sophia	Baker	sophia.baker@ic.com	(555) 555-5576	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	22	56000	0.56
22	22	Lucas	Nelson	lucas.nelson@ic.com	(555) 555-5577	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	23	58000	0.58
23	23	Isabella	Hill	isabella.hill@ic.com	(555) 555-5578	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	24	60000	0.6
24	24	Alexander	Ward	alexander.ward@ic.com	(555) 555-5579	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	25	62000	0.62
25	25	Evelyn	Allen	evelyn.allen@ic.com	(555) 555-5580	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	26	64000	0.64
26	26	William	Young	william.young@ic.com	(555) 555-5581	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	27	66000	0.66
27	27	Grace	King	grace.king@ic.com	(555) 555-5582	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	28	68000	0.68
28	28	Henry	Wright	henry.wright@ic.com	(555) 555-5583	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	29	70000	0.7
29	29	Victoria	Scott	victoria.scott@ic.com	(555) 555-5584	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	30	72000	0.72
30	30	Thomas	Young	thomas.young@ic.com	(555) 555-5585	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	31	74000	0.74
31	31	Mia	Reed	mia.reed@ic.com	(555) 555-5586	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	32	76000	0.76
32	32	Joseph	Cook	joseph.cook@ic.com	(555) 555-5587	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	33	78000	0.78
33	33	Charlotte	Baker	charlotte.baker@ic.com	(555) 555-5588	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	34	80000	0.8
34	34	Samuel	Nelson	samuel.nelson@ic.com	(555) 555-5589	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	35	82000	0.82
35	35	Aria	Hill	aria.hill@ic.com	(555) 555-5590	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	36	84000	0.84
36	36	Matthew	Ward	matthew.ward@ic.com	(555) 555-5591	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	37	86000	0.86
37	37	Madison	Allen	madison.allen@ic.com	(555) 555-5592	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	38	88000	0.88
38	38	David	Young	david.young@ic.com	(555) 555-5593	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	39	90000	0.9
39	39	Olivia	King	olivia.king@ic.com	(555) 555-5594	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	40	92000	0.92
40	40	Benjamin	Wright	benjamin.wright@ic.com	(555) 555-5595	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	41	94000	0.94
41	41	Sophia	Scott	sophia.scott@ic.com	(555) 555-5596	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	42	96000	0.96
42	42	Lucas	Young	lucas.young@ic.com	(555) 555-5597	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	43	98000	0.98
43	43	Isabella	Reed	isabella.reed@ic.com	(555) 555-5598	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	44	100000	1.0
44	44	Alexander	Cook	alexander.cook@ic.com	(555) 555-5599	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	45	102000	1.02
45	45	Evelyn	Baker	evelyn.baker@ic.com	(555) 555-5600	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	46	104000	1.04
46	46	William	Nelson	william.nelson@ic.com	(555) 555-5601	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	47	106000	1.06
47	47	Grace	Hill	grace.hill@ic.com	(555) 555-5602	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	48	108000	1.08
48	48	Henry	Ward	henry.ward@ic.com	(555) 555-5603	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	49	110000	1.1
49	49	Victoria	Allen	victoria.allen@ic.com	(555) 555-5604	1400 14th Ave. S.W.	Atlanta	GA	USA	30334	2017-01-01	50	112000	1.12

Рис. 3. Візуалізація працівників компанії IC для аналізу діяльності співробітників компаній
Джерело: опрацювання власне

За результатами експериментів встановлено, що система коректно масштабується і може генерувати десятки тисяч записів без втрати узгодженості моделі. Дотепер корпоративні структури збирають дані про

співробітників у значних масштабах. Вони включають інформацію про освіту, досвід роботи, навички, поведінкові метрики, рівень відповідальності, результативність виконаних завдань, психологічні характеристики, часові показники присутності і виконані операції [2]. Проте безпекові та юридичні бар'єри унеможливають передавання таких даних стороннім розробникам, університетам або компаніям, що займаються аудитом або тестуванням систем. У таких умовах синтетичні HR-дані починають виконувати роль універсального ресурсу, що дозволяє формувати експериментальні або тренувальні набори з мінімальними ризиками.

Розроблена система FakeUkrBase розглядається не лише як утиліта для генерації окремих значень, а як інструмент моделювання, який формує цілісні структурні дані, здатні відтворювати поведінкові залежності, організаційну ієрархію та економічно обґрунтовані параметри оплати праці. Під час конструювання системи враховувалися такі аспекти, як відповідність структури відділів реальному корпоративному середовищу, прив'язка посад до рівнів відповідальності, залежність між нарахуванням заробітної плати та рівнем кваліфікації, логіка бонусів і штрафів, а також вплив досвіду працівника на його оцінювання.

Особливу увагу під час проектування системи приділено відтворюваності даних. У реальних умовах тестування та аналітики важливо не лише сформувати множину записів, але й забезпечити можливість її повторної реконструкції в ідентичному вигляді. Це дозволяє дослідникам проводити порівняльні аналітичні експерименти, де зміни параметрів виступають єдиною варіаційною величиною. Використання механізму seed-значень гарантує стабільність і контрольованість експериментальних умов.

Технічна реалізація системи передбачає використання декількох рівнів обробки даних. На першому рівні створюються окремі атрибути, такі як ім'я, прізвище, по батькові, дата народження, стать, посада, департамент, дата працевлаштування. На другому рівні формується логіка залежностей: рівень заробітної плати розраховується відповідно до позиції та департаменту,

визначається ймовірність зміни посади протягом певного періоду, встановлюються значення коефіцієнтів бонусів залежно від продуктивності.

На наступних рівнях реалізується зв'язність між відділами, внутрішні управлінські відносини, кар'єрні траєкторії та часові перехідні події, наприклад підвищення зарплати або зміна ролі. Система також враховує сучасні тенденції HR-аналітики, зокрема використання показників ефективності, моделювання ризиків звільнення, аналіз поведінкових профілів, прогнозування плинності кадрів і формування показників workforce health. Ці показники дозволяють застосовувати згенеровані дані в моделюванні алгоритмів машинного навчання [3], таких як класифікація кандидатів, кластеризація співробітників за їхніми професійними профілями чи прогнозування кар'єрного розвитку.

На рівні даних система FakeUkrBase працює не як випадковий генератор, а як доменно-орієнтована модель. Такий підхід відрізняє її від Faker чи Mimesis, де параметри не формують логічної структури бізнес-процесів. Наприклад, у Faker посада може не відповідати зарплаті або галузі. У FakeUkrBase кожен параметр є похідною логічною функцією від інших параметрів, що дозволяє створювати реалістичну інформаційну модель підприємства. Дані можуть бути масштабовані до десятків або сотень тисяч записів без втрати внутрішньої узгодженості.

Важливою особливістю системи є готовність до інтеграції у BI-середовища. Завдяки експорту в Parquet або ClickHouse користувачі можуть швидко завантажувати дані в Metabase, Grafana, Power BI, Superset або Tableau. Це дозволяє створювати звіти, візуалізації, дашборди і проводити глибинний аналіз структури персоналу як у реальних компаніях. Під час практичної перевірки система демонструвала стабільну продуктивність, що дозволило оцінити не лише коректність генерації даних, а й швидкість виконання операцій при різних обсягах. Було проведено серію експериментів, що включали генерацію від 100 до 100000 рядків даних. Час генерації збільшувався лінійно, а перевірка логічної узгодженості забезпечувала стабільність результатів.

Анотація. У рамках дослідження здійснено проектування і розроблено інформаційну систему *FakeUkrBase* демонструє можливість побудови повноцінної інфраструктури синтетичних HR-даних у контексті українського бізнес-середовища. На відміну від існуючих рішень [4], система забезпечує логічну узгодженість моделей, адаптацію до використання в ВІ-процесах та підтримку реалістичних кадрових сценаріїв. Вона може бути використана у тестуванні корпоративних HR-систем, дослідженнях, машинному навчанні та цифровій освіті, де важливим є доступ до якісних, але не конфіденційних даних. У межах експериментального застосування було створено симуляцію середнього українського ІТ-підприємства з розподілом ролей між інженерами, менеджерами, аналітиками, адміністрацією та допоміжним персоналом. Модель дозволила створити реалістичний розподіл зарплат, посад, вікових категорій, рівнів стажу та продуктивності. Перспективи розвитку системи включають інтеграцію із сервісами машинного навчання, моделюванням поведінкових подій, створенням моделі змін у часі, симуляцією продуктивності команд, прогнозуванням кризових сценаріїв, емоційного вигорання, ризику звільнень та оцінювання ефективності управлінських рішень.

Ключові слова: дослідницьке навчання, варіативні моделі, математика, комп'ютерне моделювання, КОМСДН, інформаційна система, інтелектуальний аналіз даних, *CleverCOMSRL*, *AI*.

Література

1. Hrybiuk O., Kant G.S. *CleverCOMSRL: Implementation of an AI Computer-Aided Design System in the Context of the Cognitive Science Paradigm for the Research Training Process*. In: Machado, J., et al. *Innovations in Mechatronics Engineering III*. *icieng* 2024. *Lecture Notes in Mechanical Engineering*. Springer, Cham. 2024. P. 351-362. https://doi.org/10.1007/978-3-031-61575-7_32.
2. Hrybiuk O. Improvement of the educational process by the creation of centers for intellectual development and scientific and technical creativity. In: Hamrol, A., Kujawińska, A., Barraza, M.F.S. (eds.) *MANUFACTURING 2019*. LNME, 2019. P. 370–382. Springer, Cham. https://doi.org/10.1007/978-3-030-18789-7_31.
3. Hrybiuk O.O. Research learning of the natural science and mathematics cycle using computer-oriented methodological systems. Monograph. Drahomanov NPU, Kyiv, 2019. P. 127–135.
4. Hrybiuk O. Experience in Implementing Computer-Oriented Methodological Systems of Natural Science and Mathematics Research Learning in Ukrainian Educational Institutions. In: Machado J., Soares F. (eds) *Innovations in Mechatronics Engineering. Lecture Notes in Mechanical Engineering*, 2022. P. 55-68. Springer, Cham Online.

Шитова М. Д. (здобувач бакалаврату, 4 курс),
Науковий керівник – доц. Гріб'юк О. О
*Інститут цифровізації освіти НАПН України, ЗВО «Міжнародний науково-
технічний університет імені академіка Юрія Бугая» (Київ)*

РОЗРОБЛЕННЯ МАТЕМАТИЧНО ОБҐРУНТОВАНОЇ МОДЕЛІ ІГРОВОЇ СИСТЕМИ В ЖАНРІ ПЛАТФОРМЕРА З ВИКОРИСТАННЯМ РУШІЯ UNITY

Ігрова індустрія є однією з найбільш технологічних та прибуткових галузей сучасності, а жанр «Платформер» незмінно зберігає свою актуальність, еволюціонуючи від простих аркад до складних технічних проєктів.. Пропоновані моделі демонструють, що геймплей тепер будується навколо більш глибоких, взаємопов'язаних механік - чи то систем дослідження, фізичної взаємодії, чи складного контролю над персонажем [1].

Мета дослідження полягає у розробленні ігрової системи в жанрі «Платформер» із використанням рушія *Unity*. *Завдання*: 1. Здійснення ґрунтового аналізу щодо використання існуючих застосунків і добір технологій із урахуванням особливостей жанру «Платформер» та специфіки рушія *Unity*. 2. Розроблення ескізного проєкту архітектури ігрової системи, визначення основних модулів та їхніх взаємозв'язків. 3. Розроблення та програмна реалізація комплексу взаємопов'язаних ігрових механік для жанру «Платформер». 4. Здійснення комплексного тестування взаємодії компонентів ігрової системи для підтвердження їх коректної спільної роботи та позитивного впливу на загальний ігровий процес.

У рамках дослідження було успішно здійснено комплексний теоретичний аналіз, проєктування та програмну реалізацію ігрової системи «Підземелля Клауса» у жанрі платформер. На основі дослідження сучасних аналогів було сформовано концептуальні засади проєкту [2], а практичне розроблення і результати тестування підтверджують ефективність і доцільність використання середовища *Unity*, що разом із мовою *C#* є потужним інструментом для побудови ігрових систем [3]. Активне застосування принципів об'єктно-

орієнтованого програмування дозволило ефективно структурувати кодову базу, забезпечивши гнучкість, модульність та легкість у масштабуванні проєкту [4]. У процесі реалізації було впроваджено нові механіки, зокрема оновлено фізичну модель руху головного героя та додано можливість прискореного переміщення. Важливим технічним етапом стала розробка розумних ворогів, здатних динамічно змінювати поведінку від патрулювання території до переслідування гравця [5]. Для покращення зворотного зв'язку стандартні показники здоров'я було замінено на адаптивний графічний інтерфейс, а також інтегровано лічильник смертей, що додає ігровому процесу додатковий елемент виклику та дозволяє відстежувати статистику [6].

Для керування спрайтами головного героя та ворогів використано компонент *Animator* [7]. Логіка переходів між анімаціями побудована на системі параметрів *Speed (float)*, що відповідає за перехід від стану спокою до бігу (див. Додаток Г, Додаток Д). Значення передається з модуля *Mathf.Abs(horizontalInput)*. Та *IsGrounded (bool)*, який визначає, чи знаходиться персонаж на землі, активуючи анімацію стрибка або падіння. Особливістю візуалізації руху є використання компонента *SpriteRenderer*. Замість створення окремих анімацій для руху вліво та вправо, у методі *PlayerMovement()* реалізовано програмне дзеркальне відображення спрайту (рис. 1).

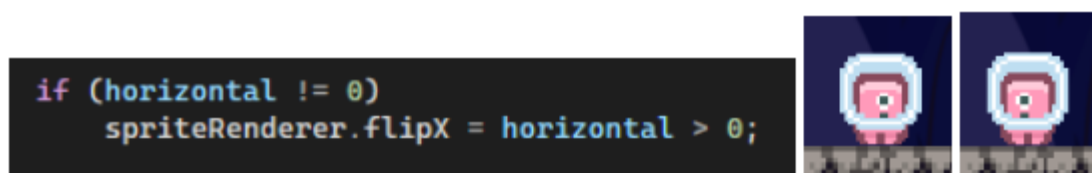


Рис. 1. Дзеркальне відображення спрайту гравця
Джерело: опрацювання власне

Для створення ілюзії польоту ворога була використана комбінація системи *Animation* рушія *Unity* та програмної логіки для контролю напрямку. Було створено анімаційний кліп за допомогою вбудованого вікна *Animation* в *Unity* [6]. Анімація базується на трьох ключових спрайтах, що зображують різні фази руху крил: «крила вгорі», «крила посередині» та «крила вниз». Ці

спрайти розміщені на часовій шкалі анімації для створення безперервного циклічного руху махання крилами, що надає ворогу «життя».

Візуалізація стану здоров'я винесена в окремий шар інтерфейсу (*Canvas*). Замість текстових лічильників використано графічний підхід у вигляді сердечок, що відповідає жанровим стандартам платформерів [2]. Технічна реалізація базується на компоненті *Image* з типом заповнення *Filled (Vertical)*. У методі *Start()* гравцю надається повний запас здоров'я ($health = healthMax$), а UI-шкала оновлюється ($healthImage.fillAmount = health / healthMax$). Коли викликається метод нанесення шкоди *Damage()*, здоров'я гравця *health* зменшується на *damgeCount* (значення, отримане від ворога). UI-шкала *healthImage* оновлюється, візуально показуючи втрату здоров'я (рис. 2).



Рис. 2. UI-шкала *healthImage*
Джерело: опрацювання власне

GameManager.cs виконує роль центрального контролера статистики. Змінна лічильника (*deathCount*) - приватна цілочисельна змінна, яка зберігає поточну кількість невдач. Публічний метод *AddDeath()* збільшує на 1 значення лічильника та викликає оновлення графічного інтерфейсу. Для виведення інформації про кількість смертей гравця використовується компонент *TextMeshPro*. А метод *UpdateUI()* конвертує числове значення у рядковий формат та оновлює текстове поле на екрані [3].

Виклик методу підрахунку інтегровано у систему обробки пошкоджень персонажа. Інкремент лічильника відбувається у двох випадках: критичне пошкодження та втрата здоров'я у бою. Логіка руху персонажа знаходиться в методі *PlayerMovement()*, який викликається з *FixedUpdate()*. *FixedUpdate()* використовується для всіх операцій, пов'язаних із рухом, оскільки він

викликається через фіксовані проміжки часу, що забезпечує плавність і стабільність руху незалежно від частоти кадрів (FPS) (рис. 3) [4].

```
private void PlayerMovement()
{
    float horizontal = Input.GetAxisRaw("Horizontal");
    transform.position += Vector3.right * horizontal * currentMoveSpeed * Time.fixedDeltaTime;

    if (horizontal != 0)
        spriteRenderer.flipX = horizontal > 0;
}
```

Рис. 3. Метод *PlayerMovement()*
Джерело: опрацювання власне

Замість статичного значення *Speed* тепер використовується динамічна змінна *currentMoveSpeed*. Рух реалізовано шляхом прямої зміни *transform.position*. Додано множення на *Time.fixedDeltaTime*, для нормалізації швидкості руху. Він гарантує, що персонаж буде рухатися з однаковою швидкістю незалежно від того, скільки разів на секунду виконується *FixedUpdate()* [5].

Логіка стану патрулювання. Якщо змінна *target == null*, активується стан патрулювання. Цільова позиція (*moveTargetPosition*) встановлюється на координати поточної точки маршруту *points[currentPoint]*. Перевірка досягнення точки реалізована через обчислення дистанції: *Vector2.Distance(...) < 0.1f*. Використання порогового значення (0.1f) замість точного порівняння забезпечує стабільність роботи фізики. Після досягнення точки індекс *currentPoint* збільшується. Досягнувши останньої точки, лічильник скидається на 0, створюючи нескінченний цикл (рис. 4) [6].

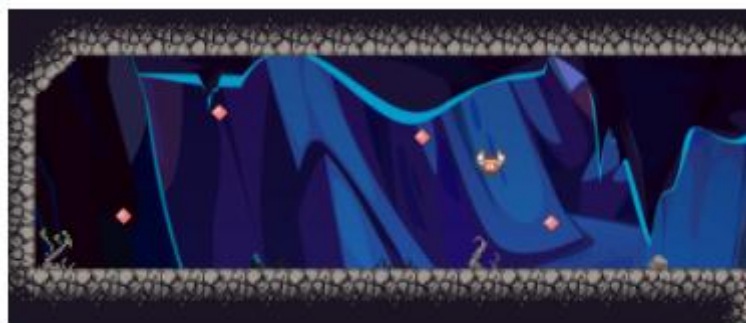


Рис. 4. *Points* нескінченного циклу патрулювання ворога на карті
Джерело: опрацювання власне

Тестування ігрової системи здійснювалося безперервно на всіх стадіях розробки, що дозволило не лише вчасно усунути помилки, а й грамотно налаштувати баланс складності для покращення користувацького досвіду [7]. Проєкт дав можливість закріпити теоретичні знання щодо управління об'єктами у 2D-просторі, обробки подій та створення UI-елементів. Окрім технічних аспектів, робота сприяла розвитку алгоритмічного мислення та вмінню підтримувати чистоту коду. Отже, розробка ІПС «Підземелля Клауса» стала успішною реалізацією поставлених завдань і важливим етапом професійного становлення, створивши міцне підґрунтя для розробки складних ігрових проєктів у майбутньому.

Анотація. Використання ігрових рушіїв, серед яких провідне місце займає Unity, дозволяє ефективно проєктувати та реалізовувати складні інформаційні системи, забезпечуючи гнучкість розробки та високу продуктивність. Розроблення таких систем з нуля вимагає глибокого розуміння принципів об'єктно-орієнтованого програмування та архітектури безпосередньо рушія. Мета дослідження полягає у розробленні ігрової системи в жанрі «Платформер» із використанням рушія Unity. Перспективи подальших розвідок полягають у розширенні функціональних можливостей створеної системи. Передбачається реалізація масштабованої карти світу з новими зонами для дослідження, розширення асортименту предметів, інтеграція NPC із квестами та діалогами, а також додавання системи досягнень за зразком сучасних ігор платформи Steam. Окрім того, подальший розвиток може охоплювати впровадження різних типів ворогів із відмінною поведінкою та складністю, що сприятиме створенню варіативного геймплею.

Ключові слова: геймдизайн орієнтований підхід, дослідницьке навчання, варіативні моделі, комп'ютерне моделювання, КОМСДН, ігрова система, 2D платформер, AI, ігровий інтерфейс, Unity.

Література

1. Hrybiuk Olena. Engineering in Educational Institutions: Standards for Arduino Robots as an Opportunity to Occupy an Important Niche in Educational Robotics in the Context of Manufacturing 4.0, in: *Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*. Volume 27-32, 2020. P. 770-785.
2. Hrybiuk O. Problems of expert evaluation in terms of the use of variative models of a computer-oriented learning environment of mathematical and natural science disciplines in schools, [w:] *Zeszyty Naukowe Politechniki Poznańskiej. Seria: Organizacja i Zarządzanie*, Zeszyt Nr 79, Poznań: Wydawnictwo Politechniki Poznańskiej (WPP), 2019. P. 101-119.

3. Гриб'юк О.О. *Дослідницьке навчання учнів предметів природничо-математичного циклу з використанням комп'ютерно орієнтованих методичних систем* / О. О. Гриб'юк. Монографія. Київ: НПУ імені М. П. Драгоманова, 2019.
4. Hrybiuk O., Vedishcheva O. Experimental Teaching of Robotics in the Context of Manufacturing 4.0: Effective Use of Modules of the Model Program of Environmental Research Teaching in the Working Process of the Centers "Clever". In: , et al. *Innovations in Mechatronics Engineering II. icieng 2022. Lecture Notes in Mechanical Engineering*. Springer, Cham. 2022.
5. Hrybiuk O. Experience in Implementing Computer-Oriented Methodological Systems of Natural Science and Mathematics Research Learning in Ukrainian Educational Institutions. In: Machado J., Soares F. (eds) *Innovations in Mechatronics Engineering. Lecture Notes in Mechanical Engineering*, 2022. P. 55-68. Springer, Cham Online.
6. Гриб'юк О.О. Комп'ютерне моделювання та робототехніка в навчально-виховному процесі сучасного навчального закладу. / Гриб'юк О.О. // *Сьома міжнародна науково-практична конференція FOSS Lviv-2017: Збірник наукових праць*, 27-30 квітня 2017 року, м. Львів. С. 38-43. ISBN 978-966-2598-86-5.
7. Hrybiuk O. Improvement of the Educational Process by the Creation of Centers for Intellectual Development and Scientific and Technical Creativity. In: Hamrol A., Kujawińska A., Barraza M. (eds) *Advances in Manufacturing II. MANUFACTURING 2019. Lecture Notes in Mechanical Engineering*, 2019. P. 370-382. Springer, Cham Online.

Наукове видання

**КЛАСИЧНА ТА ПРИКЛАДНА МАТЕМАТИЧНА ОСНОВА
ГАЛУЗЕВИХ ДОСЛІДЖЕНЬ У НАУКОВОМУ ПРОСТОРІ
ЗДОБУВАЧІВ ВИЩОЇ ОСВІТИ І МОЛОДИХ ВЧЕНИХ**

**Матеріали Всеукраїнської науково-практичної
конференції здобувачів вищої освіти і молодих вчених**

09 квітня 2026 року

Відповідальний за випуск *Ємельянова Т.В.*

В авторській редакції

Комп'ютерна верстка *Купіної Н.А.*

ВИДАВНИЦТВО

Харківського національного автомобільно-дорожнього університету

Видавництво ХНАДУ, 61200, Харків-МСП, вул. Ярослава Мудрого, 25.

Тел. /факс: (057)700-38-72; 707-37-03, e-mail: rio@khadi.kharkov.ua

*Свідоцтво Державного комітету інформаційної політики, телебачення
та радіомовлення України про внесення суб'єкта видавничої справи
до Державного реєстру видавців, виготівників і розповсюджувачів
видавничої продукції, серія № ДК №897 від 17.04 2002 р.*