

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації

(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук

(повна назва кафедри (предметної, циклової комісії))

## МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Інформаційна технологія гейміфікації

додатку для вивчення основ

математики»

Виконав: студент 2-го курсу, групи

ЗКН-22м спеціальності 122

«Комп'ютерні науки»

(шифр і назва напрямку підготовки, спеціальності)

Чверкун-Капуш М.О.

(прізвище та ініціали)

Керівник: к.т.н., ст. в. кафедри КН

Петришин С.І.

(прізвище та ініціали)

« 04 » 12 2023 р.

Опонент: к.т.н., доц.кафедри АІТ

Барабан М.В.

(прізвище та ініціали)

« 07 » 12 2023 р.

Допущено до захисту

Завідувач кафедри КН

д.т.н., проф. Яровий А.А.

(прізвище та ініціали)

« 08 » 12 2023 р.

Вінниця ВНТУ - 2023 рік

Вінницький національний технічний університет  
Факультет інтелектуальних інформаційних технологій  
та автоматизації

Кафедра комп'ютерних наук

Рівень вищої освіти II-й (магістерський)

Галузь знань – 12 «Інформаційні технології»

Спеціальність – 122 «Комп'ютерні науки»

Освітньо-професійна програма – «Системи штучного інтелекту»

**ЗАТВЕРДЖУЮ**

**Завідувач кафедри КН**

**Д.т.н., проф. Яровий А.А.**

29.08 2023 року

### **ЗАВДАННЯ**

#### **НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Чверкун Капуш Михайло Олексійович

(прізвище, ім'я, по батькові)

1. Тема роботи «Інформаційна технологія гейміфікації додатку для вивчення основ математики»

керівник роботи к.т.н., ст. в. кафедри КН Петришин С.І.

затвердені наказом вищого навчального закладу від 18.08.2023 року № 247

2. Строк подання студентом роботи 13.11 2023 року

3. Вихідні дані до роботи:

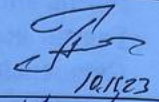
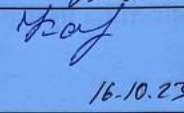
1) наявність не менше ніж 4 видів математичних задач; 2) можливість підвантаження бази до 100 питань 3) об'єктно-орієнтована мова програмування, ОС Windows

4. Зміст текстової частини:

Вступ, аналіз предметної області гейміфікації, проектування роботи програмного модуля гейміфікації, реалізація програмного забезпечення програмного модуля гейміфікації, економічна частина, висновки.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)  
UML-діаграма ігрової математичної програми, діаграма прецедентів ігрової математичної програми, алгоритм функціонування системи, алгоритм функціонування основних модулів системи, структурна схема роботи додатку, робочі вікна програми.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-3	Петришин С.І., к.т.н., ст. викл. каф. КН	 29.08.23	 10.10.23
4	Кавецький В.В., к.е.н., доц. каф. ЕПВМ	 16.10.23	 20.10.23

7. Дата видачі завдання 29.08 2023 року

КАЛЕНДАРНИЙ ПЛАН


№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області гейміфікації	01.09.23 - 07.09.23	
2	Проектування інформаційної технології гейміфікації додатку для вивчення основ математики	08.09.23 - 24.09.23	
3	Реалізація інформаційної технології гейміфікації додатку для вивчення основ математики	25.09.23 - 15.10.23	
4	Підготовка економічної частини	16.10.23 - 20.10.23	
5	Апробація та/або впровадження результатів дослідження	21.10.23 - 5.11.23	
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	06.11.23 - 10.11.23	

Студент

  
(підпис)

Чверкун-Капуш М.О.

Керівник роботи

  
(підпис)

Петришин С.І.

## АНОТАЦІЯ

УДК 004.9:51.

Чверкун-Капуш М.О. Інформаційна технологія гейміфікації додатку для вивчення основ математики. Магістерська кваліфікаційна робота зі спеціальності 122 – “Комп’ютерні науки”, освітня програма – “Системи штучного інтелекту”. Вінниця:ВНТУ, 2023. 105 с.

Укр. мовою. Бібліогр.: 20 назв; рис.: 16; табл. 14.

В ході даного дослідження було зосереджено увагу на важливості гейміфікації при розробці освітніх систем. Проведений аналіз передумов для впровадження гейміфікації та основних принципів цього підходу. Також досліджено принципи розробки наявного програмного забезпечення для гейміфікації і зроблено аналіз його переваг і недоліків, на основі якого була поставлена задача дослідження.

В рамках роботи було розроблено алгоритм функціонування модуля гейміфікації для вивчення основ математики, і на його основі було створено відповідний програмний модуль. При розробці програмного забезпечення була обрана об'єктно-орієнтована мова програмування C# та використані бібліотеки, такі як LINQ to XML, LINQ to Objects, LINQ to Entities і інші. Розроблений модуль має кілька переваг, зокрема, відсутність обов'язкової авторизації, наявність різних режимів тестування, дружній інтерфейс та лаконічне представлення вікон і кнопок. Програмний модуль пройшов успішне тестування у різних режимах, підтвердивши його коректну роботу. Крім того, тестування підтвердило ефективність розробленої системи гейміфікації в порівнянні з аналогами.

Ключові слова: гейміфікація, програмний модуль, математика, формула, десктопний додаток.

## ABSTRACT

Chverkun-Kapush M.O. Information technology gamification of the application for learning the basics of mathematics. Master's qualification work on specialty 122 - "Computer science", educational program - "Artificial intelligence systems". Vinnytsia: VNTU, 2023. 105 p.

In Ukrainian speech Bibliography: 20 names; Fig.: 16; table 14.

In the course of this study, attention was focused on the importance of gamification in the development of educational systems. An analysis of the prerequisites for the introduction of gamification and the main principles of this approach was carried out. The principles of development of existing software for gamification were also investigated and an analysis of its advantages and disadvantages was made, on the basis of which the research task was set.

As part of the work, an algorithm for the functioning of the gamification module for the study of the basics of mathematics was developed, and a corresponding software module was created on its basis. When developing the software, the object-oriented programming language C# was chosen and libraries such as LINQ to XML, LINQ to Objects, LINQ to Entities and others were used. The developed module has several advantages, in particular, the absence of mandatory authorization, the availability of various test modes, a friendly interface and a concise presentation of windows and buttons. The software module was successfully tested in various modes, confirming its correct operation. In addition, testing confirmed the effectiveness of the developed gamification system in comparison with analogues.

Keywords: gamification, software module, mathematics, formula, desktop application.

## ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ГЕЙМИФІКАЦІЇ.....	7
1.1 Аналіз предметної області розробок гейміфікації.....	7
1.2 Аналітичний огляд відомих реалізацій систем гейміфікації .....	14
1.3 Методи оцінювання ефективності застосування гейміфікації у навчанні.....	17
1.4 Постановка задачі дослідження .....	29
1.5 Висновок до розділу 1.....	33
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ГЕЙМИФІКАЦІЇ ДОДАТКУ ДЛЯ ВИВЧЕННЯ ОСНОВ МАТЕМАТИКИ .....	34
2.1 Проектування UML діаграм .....	34
2.2 Розробка алгоритмів функціонування інформаційної технології гейміфікації вивчення основ математики.....	45
2.3 Висновок до розділу 2.....	47
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ГЕЙМИФІКАЦІЇ ДЛЯ ВИВЧЕННЯ ОСНОВ МАТЕМАТИКИ .....	48
3.1 Обґрунтування середовища і мови програмування.....	48
3.2 Розробка програмного модулю гейміфікації додатку для вивчення основ математики.....	52
3.3 Тестування розробленого програмного модуля гейміфікації додатку для вивчення основ математики.....	56
3.4 Порівняльна характеристика ефективності розробленої інформаційної технології гейміфікації додатку для вивчення основ математики .....	59
3.5 Висновок до розділу 3 .....	61
4 ЕКОНОМІЧНА ЧАСТИНА .....	62
4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки .....	62
4.2 Розрахунок узагальненого коефіцієнта якості розробки .....	66

4.3 Розрахунок витрат на проведення науково-дослідної роботи .....	67
4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором.....	78
4.5 Висновок до розділу 4 .....	82
ВИСНОВКИ.....	84
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	85
ДОДАТКИ.....	87
Додаток А (обов'язковий) Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень.....	88
Додаток Б (обов'язковий) Лістинг програми .....	89
Додаток В (обов'язковий) Ілюстративна частина.....	99
Додаток Г (довідниковий) Інструкція користувача .....	105

## ВСТУП

**Актуальність.** Сучасні навчальні заклади все більше використовують гейміфікацію як засіб заохочення учнів та поліпшення їх успішності. Гейміфікація, або використання ігрових практик та механізмів у неігрових ситуаціях, є поширеним явищем, що досліджувалось у різних областях, таких як взаємодія з клієнтами, фізичні тренування, фінансові вкладення, якість даних, пунктуальність та навчання.

В сучасній освіті у дітей постійно зростає інтерес до нових методів, матеріалів та підходів до навчання. Одним з цих підходів є гейміфікація – використання елементів гри в освітньому процесі. Гейміфікація в шкільному навчанні – це не просто новий тренд, а важливий інструмент, який сприяє залученню учнів до навчання, покращенню мотивації та розвитку їхніх навичок, залучення зацікавленості та захоплення від чогось нового, нетрадиційного та цікавого.

Актуальність гейміфікації полягає у необхідності зацікавити учнів у процесі навчання. Зростання рівня взаємодії учнів та студентів з інтерактивними програмами та гаджетами призвело до збільшення вимог до ефективності навчального процесу, зокрема до здатності утримувати увагу та зацікавленість учнів, що має прямий вплив на засвоєння інформації. Особливо це стає важливим у контексті зростаючої кількості дистанційного навчання, яке стало необхідним через пандемію та військовий стан, і вимагає мотивації та зацікавленості учнів у надзвичайно складних умовах.

**Зв'язок роботи з науковими програмами, планами, темами.** Магістерська кваліфікаційна робота виконана відповідно до напряму наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, технології та пристрої інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій» та плану наукової та навчально-методичної роботи кафедри.



**Мета та завдання досліджень.** Мета магістерської кваліфікаційної роботи – підвищення ефективності вивчення основ математики за рахунок застосування гейміфікації.

Для досягнення мети роботи потрібно виконати такі задачі:

- Провести аналіз предметної області гейміфікації вивчення основ математики;
- Розробити структуру інформаційної технології гейміфікації вивчення основ математики;
- Розробити алгоритм роботи інформаційної технології гейміфікації вивчення основ математики;
- Реалізувати програмне забезпечення гейміфікації вивчення основ математики;
- Провести тестування розробленого програмного забезпечення;
- Здійснити економічні розрахунки доцільності розробки інформаційної технології гейміфікації додатку для вивчення основ математики.

**Об’єкт дослідження** – процес гейміфікації додатку для вивчення основ математики.

**Предмет дослідження** – інформаційні технології гейміфікації додатку для вивчення основ математики.

**Методи дослідження.** У роботі використані такі методи наукових досліджень: методи системного аналізу, теорія об’єктно-орієнтованого програмування, методи дизайну та стилізації десктопних додатків, методи оптимізації програмного забезпечення, методи роботи з файловою системою, методи організації роботи даних та знань.

**Наукова новизна одержаних результатів.** Набула подальшого розвитку інформаційна технологія гейміфікації додатку для вивчення основ математики, що відрізняється від існуючих удосконаленою моделлю гейміфікації, що дозволяє підвищити ефективність вивчення математичних дисциплін.

**Практичне значення** одержаних результатів полягає в тому, що на основі проведених досліджень розроблено:

- алгоритм гейміфікації процесу вивчення основ математики;
- програмне забезпечення гейміфікації вивчення основ математики.

**Достовірність теоретичних положень** магістерської кваліфікаційної роботи підтверджується коректністю постановки завдання, коректністю використання математичного апарату методів дослідження, експериментальними дослідженнями тестування програмної реалізації інформаційної технології гейміфікації додатку для вивчення основ математики. Адекватність розроблених математичних моделей підтверджується результатами експериментальних досліджень.

**Особистий внесок здобувача.** Усі результати, що наведені у магістерській кваліфікаційній роботі, отримані самостійно. У працях, які написано у співавторстві, здобувачу належать: огляд застосунків для гейміфікації вивчення основ математики [1].

**Апробація результатів.** Результати дослідження були представлені та обговорені на ЛІІ науково-технічній конференції підрозділів ВНТУ [1].

**Публікації.** За основними результатами досліджень опубліковано тези доповіді наукової конференції «ЛІІ Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2023)» [1]

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ГЕЙМИФІКАЦІЇ

## 1.1 Аналіз предметної області розробок геймифікації

Як і в будь-якій іншій області, основу в галузі геймифікації становить досить велика кількість психологічних та поведінкових принципів, серед яких можна виділити 4 базові:

- принцип мотивації;
- принцип несподіваних відкриттів та заохочень;
- принцип статусу;
- принцип винагороди.

Далі проаналізуємо сутність та особливості цих принципів.

Принцип свідчить, що гравці мають бажати комунікувати і взаємодіяти, тобто бути мотивованими. Найсильнішими мотиваторами у будь-якій діяльності є прагнення досягти задоволення і водночас уникнути неприємних ситуацій.

У першому випадку можна використовувати як реальний, так і ігровий приз, головне, щоб клієнт відчув заохочення.

Такі знання формуються на даних про цільову аудиторію – чим краще ви знаєте про те, що мотивує потенційних учасників (клієнтів, співробітників, колег, друзів тощо), тим ефективніше ви зможете на них впливати.

Будь-які заохочення за пройдені квести, у тому числі бонуси, знижки, специфічні нагороди, викликає у гравців не тільки яскраві емоції, але й цікавість, що важливо для досягнення кінцевої мети, будь то змагання, рейтингова гра або завдання.

Важливість набути статусу у будь-якому виді діяльності – це природний інстинкт, який веде до бажання досягти чогось більшого. При грамотній структуризації ігрового процесу з'являється багато опцій демонстрації успіху та прогресу, що дозволяє людині довести свої плюси оточенню та суспільству в цілому. Формування високої самооцінки сприяє

розвитку такої моделі лідера, яку клієнт справді мріє досягти.

Стимулами та індикаторами успіху можуть бути такі дріб'язкові предмети, як бейджи, значки, медальки та звання. Також цьому сприяє відображення інформації в ході ігрового процесу, адже учасники зможуть порівнювати особисті успіхи та досягнення з успіхами та досягненнями інших.

Ігрофікація різних процесів та явищ схиляє розробників задуматися про систему нагород. Проблема в тому, що невелика частина розробників здатна придумати нагороди, що підходять до предметної області цільової аудиторії.

Винагороди бувають такими, що підвищують статус, персональними, фізичними або емоційними, але багато залежить від додаткових параметрів, що базуються на інтересах і поглядах. Якщо, наприклад, брати до уваги щось фізично відчутне, потрібно бути впевненим, що цінність нагороди вартує тих зусиль, які потрібно докласти гравцям для її досягнення, інакше бажаючих взяти участь у проєкті просто не буде.

Залученість, експеримент та результат – це три «кити» гейміфікації, на яких засноване її застосування. Розберемо кожен із них детальніше.

Залучення до проєкту нових учасників, утримання вже бувалих гравців, натхнення всіх заглиблюватися в таку активність – це основи гейміфікації.

Дофамін – гормон радості, що часто виробляється протягом будь-якого ігрового процесу. А значить, створення спеціального гейм-середовища робить більш цікавим процес вирішення завдань. Втім, важливо не лише концентруватися на самій мотивації, а й опрацьовувати найдрібніші деталі тієї чи іншої предметної області – тільки так можна підняти охоплення та значно покращити кількість та якість аудиторії. Важливим є прагнення до створення внутрішньої мотивації, тобто, робити так, щоб виконання дій та розв'язання задач стало важливим вже саме собою.

Гейміфікація може мотивувати до занять спортом, екології,

раціоналізму. При цьому можна виконувати завдання командами і таким чином розвивати софт скілс.

Вдосконалення – це найчастіше результат ігрових експериментів. Кожен розуміє, що в перебігу гри він колись програє, але в той же час він знає, що у нього є можливість завжди почати спочатку, а тому перспектива поразки зовсім не страшна. У будь-якій грі можна так чи інакше здобути перемогу, але програти без можливості розпочати знову неможливо. Ігри, які тримають баланс між високою складністю та її відсутністю, є мотиваторами до самовдосконалення та розвитку, а також до креативності, пошуку нових цікавих та інноваційних рішень. Таким чином, ігрофікація важлива у бізнесі, науці, освіті, сфері послуг і не тільки.

Гейміфікація дає величезні результати. Успішні стартапи щодня впроваджуються корпораціями Samsung, American Express, Microsoft, Nike, що є вагомим аргументом до того, щоб брати приклад із цих організацій.

Розберемо застосування гейміфікації на окремих прикладах. Розглянемо спершу область email-маркетингу.

Впровадження в email-маркетинг має на увазі:

- Нагорода – головна причина, через яку клієнт підписується на ту чи іншу розсилку. Подарунок дозволяє підвищити інтерес клієнта.
- При створенні проморозсилок потрібно наголошувати на знижках, сезонних пропозиціях та інших новинках. Можна влаштувати конкурс на ту чи іншу онлайн знижку.
- При створенні веселої гри ви використовуєте різні новинки, і як наслідок, покращуєте настрій клієнта.

Гейміфікація має бути досить примітивною, оскільки розрахована на різних людей, а їх треба розважити, а не нервувати. Умови участі та отримання нагороди повинні бути прозорими.

Завдяки впровадженню елементів гейміфікації інтернет-магазину Stylus вдалося збільшити прибутковість email-маркетингу в 3 рази та розширити основу контактів у 2,5 рази (рис. 1.1).

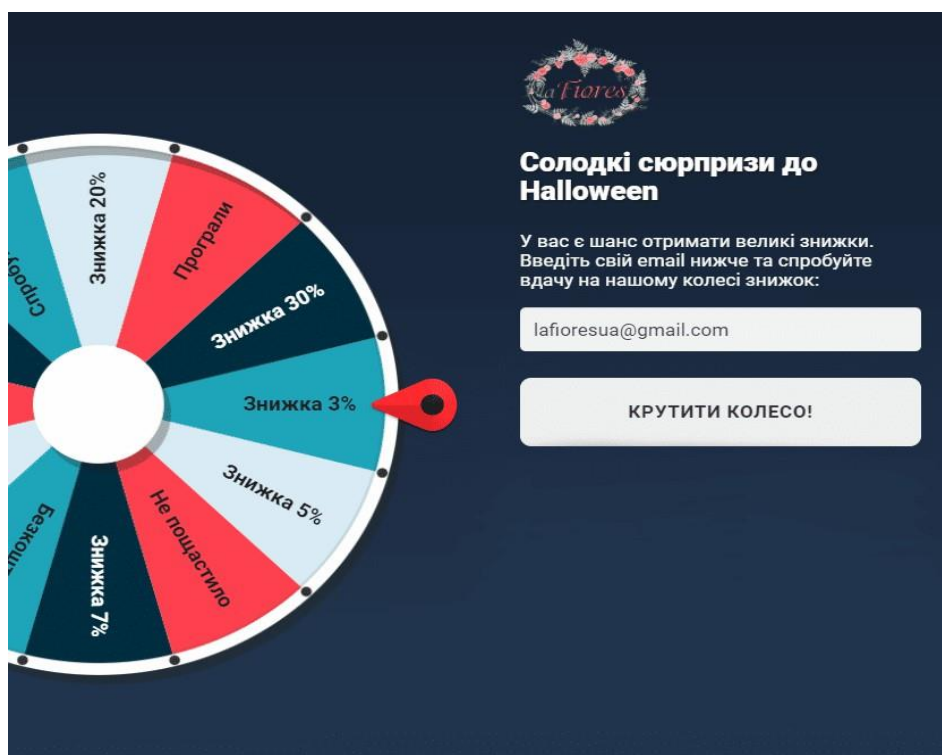


Рисунок 1.1 – Приклад гейміфікації у email-маркетингу

Гейміфікація має наступні проблеми, розглянемо їх нижче.

Існує поширена проблема, коли співробітники роблять все, щоб заробити досягнення або активно борються за право зайняти верхній рядок рейтингу, але якщо це єдина причина, що спонукає до дії, така поведінка не триватиме довго. Зовнішні мотиватори результативні і добре залучають. Але швидко зникає ефект новизни, люди схильні втрачати інтерес і повертатися до своїх старих звичок. Якщо не вірите – спробуйте мотивувати семирічної дитини виконувати нудне завдання за шоколадку. Дуже скоро ви помітите, що шоколад вважається само собою зрозумілим, і вам доведеться збільшити приз, щоб зберегти мотивацію для виконання завдання.

Слід переконатися, що ви звертаєтеся до внутрішніх мотиваторів, а не використовуєте тільки зовнішні. У наведеному вище прикладі, відчутно кращим варіантом було б зробити завдання цікавим або, якщо це неможливо, переконати учня в цінності виконання даного завдання для нього.

Хоча заохочення та нагороди безперечно мотивують людей, іноді

гейміфікація призводить до несподіваних і навіть небажаних побічних наслідків. Візьмемо, наприклад, Ефект Кобри [2], термін, що виник на просторах колоніальної Індії. Британський уряд, стурбований зростанням популяції отруйних кобр у Делі, оголосив нагороду за кожен мертву кобру. Спочатку стратегія виявилася успішною і багато отруйних змій було знищено. Проте заповзятливі люди незабаром почали спеціально розводити кобр у пошуках легкого додаткового прибутку. Розуміючи це, уряд відмовився від програми, що ще більше посилює проблему, оскільки кобри з розплідників через непотрібність виявилися на вулицю.

Виділіть час, щоб продумати систему мотивації, заохочення та варіанти можливих наслідків. Переконайтеся, що зовнішні стимули, такі як фізичні нагороди та грошові стимули, використовуються дуже обережно.

Це, на жаль, популярна проблема. Наприклад, у деяких компаніях хочуть заохотити командну роботу, але винагороджують окремих зірок усередині колективу. Або розвиваючи культуру обслуговування, нагороджують працівників за швидкість закриття клієнтських запитів та скарг. Або коли у компанії оголошується час пошуку ідей та інновацій, але за кожен помилку співробітників карають та штрафують. Також це заохочення швидкості проходження та завершення курсу замість акцентів на ефективності навчання та використанні отриманих навичок на робочому місці. Список можна продовжувати нескінченно.

Для уникнення важливо конкретно визначити цілі та результати, яких ви прагнете. Вимірюйте та винагороджуйте саме те, що вам важливо, а не те, що простіше порахувати.

Уявіть, що ви отримали подарунок, який вручили вам із любов'ю як велику цінність. І намагаєтесь сказати, що він вам не подобається або ви не уявляєте, як вписати його в своє життя. Приблизно те саме станеться, якщо ми висипатимемо купу різносортних бейджів на користувачів, не переймаючись, чи будуть вони для них цінними.

Завданням гейміфікації є побудова такої моделі гри, що якнайкраще

мотивує користувача до здійснення певних дій. Технологія гейміфікації – це трансформація матеріального на духовне. Внутрішньою нагородою може стати моральне задоволення, порівняння себе з іншими та виділення на їхньому фоні.

На основі ігрових механік сформувався перелік елементів, які найчастіше використовуються для того, щоб гейміфікувати ті чи інші процеси. Елементи гейміфікації можуть включати [3]:

#### 1) Програми лояльності

Це набір різних карток – клубних, купонів на знижку, конкурсних пропозицій, промокодів. Мета таких програм – встановити зв'язок між брендом товару та клієнтом, зацікавити його у подальших покупках. Стимулом для покупця є нагорода, яка з'являється знову та знову. Щоразу, коли покупець проходить через цей процес, він вкладає свої гроші та час, що у свою чергу посилює зв'язок із брендом.

#### 2) Рейтингова таблиця

Рейтингові таблиці застосовуються в іграх і не тільки, дозволяють оцінити людині своє місце по відношенню до інших. Чудово працює психологічний прийом, коли показують, яке місце займає користувач серед інших. Бажання піднятися вище в рейтингу і отримати публічне визнання може добре підживлювати мотивацію.

#### 3) Шкала прогресу

Візуалізація – відмінний трюк, щоб показати користувачеві, як далеко він просунувся в чомусь. Шкала може містити проміжні результати, розбивати шлях на частини, завдяки чому його проходження здається більш здійсненним.

#### 4) Очки та значки відмінності (беджі)

Очки – це форма нагороди за дії, що передбачає зворотний зв'язок та стимуляцію підвищення рівня. Значки є умовними символами, що позначають досягнення.

#### 5) Просування за рівнями



Рівні також відображають прогрес користувача. Вони корисні як винагороди, і у тому, щоб стимулювати бажання просунутися більш складний рівень. Насамперед їх широко застосовують для гейміфікації навчання, де поліпшення успішності студентів має бути завжди помітним і вимірюваним.

Математична головоломка — завдання захоплюючої математики з ігровими елементами (правилами можливих дій, іноді — сюжетом), що вимагає більшою мірою кмітливості, ніж математичної підготовки або спеціальних знань. Багато з загальновідомих головоломок так чи інакше містять математичний зміст, наприклад, в «Пентаміно» важливими є форми і розташування фігур, а в «Sudoku» — властивості графів. Гру Конвея «Життя» і задачу побудови фракталів можна також розглядати як математичні головоломки, хоча гравець оперує з ними тільки шляхом завдання початкових конфігурацій, а після того, як початкові умови встановлені, правила головоломки визначають всі подальші зміни і ходи.

Багато головоломок добре відомі, оскільки обговорювалися Мартіном Гарднером в колонці «Mathematical games» в журналі Scientific American. Математичні ігри іноді використовуються для того, щоб залучити студентів та школярів набути певних практичних навичок у вирішенні задач специфічного характеру, тобто мають розвивальну і водночас розважальну мету.

Ігрові математичні додатки можна поділити за тематикою на наступні класи:

- алгебраїчні та арифметичні ігри (числові кросворди, ребуси, китайська арифметична гра 24, знаходження циклічних чисел);
- комбінаторні ігри (Sudoku, Ханойські вежі);
- логічні ігри (різноманітні головоломки на логіку, знаходження закономірностей, кмітливість);
- ігри на застосування теорії ймовірності (парадокс Монті Холла);
- мозаїки та розбиття (Пентаміно, головоломка Конвея);

- ігри на дошці (доміно, «Життя» Конвея);
- ігри на два гравця (Хрестики-нулики);
- шахові задачі (задача про 8 ферзів);
- ігри з теорії графів (задача про 7 мостів);
- механічні головоломки (Кубик Рубіка).

Ці математичні головоломки, незважаючи на свою різноманітність, спільні у тому, що вони стимулюють розвиток логічного мислення та креативного підходу до вирішення завдань. Це ігри, що не лише розважають, а й розвивають мозок, навчаючи шляхів аналізу, систематизації та вирішення складних завдань.

Серед них алгебраїчні та арифметичні ігри, такі як числові кросворди або гра 24, спонукають до глибокого осмислення числових властивостей та алгоритмів. Комбінаторні головоломки, такі як Sudoku чи Ханойські вежі, розвивають навички логічного мислення та стратегічного планування. Логічні ігри допомагають у виявленні закономірностей та розв'язанні складних задач.

Ігри на дошці, шахові задачі, задачі з теорії графів чи механічні головоломки — кожна з них вимагає унікального підходу та вміння швидко знаходити оптимальні рішення.

## **1.2 Аналітичний огляд відомих реалізацій систем гейміфікації**

Існує безліч реалізацій математичних головоломок, вони можуть бути як розрахованими на проходження однією особою, так і мультиплеєрними, різнитися за тематичним змістом та метою розробки. Загалом, такі ігри поділяються на 2 великих класи – математичні додатки, в яких основна мета навчальна та додатки, в яких головним є тренування мозку задля розваги. До першого класу належить ПЗ у вигляді тестувальних програм онлайн, мобільних ігор та подібних навчальних додатків, що допомагають краще опанувати певну тему чи розділ математики. Другий клас містить в собі

ігрові розважальні програми, задачі, для вирішення яких необхідно застосовувати комбіновані та вже набуті математичні знання. Розглянемо деякі з комп'ютерних реалізацій математичних додатків обох видів.

– Мобільний ігровий додаток «Brain Champ». Гра є одним з найперших ігрових мобільних додатків на тренування мозку та розвиток інтелекту, була присутня ще в прошивці старих сотових телефонів «Nokia». «Brain Champ» являє собою набір різноманітних завдань на логіку, пам'ять, увагу і математичне мислення. В грі розроблено три основних режими для користувача – «Щоденне тренування», «Одиночне тренування» і «Бонусні ігри». Професор Лабідібі – персонаж, що допомагатиме фіксувати тренування в кожному зареєстрованому профілі. Спеціальний календар показує прогрес розвитку кожної частини мозку, рівень гравця з кожним тренуванням підвищується і професор Лабідібі порівнює його з одним з відомих геніїв. При досягненні останнього рівня у режимі «Щоденне тренування» відкривається доступ до «Бонусних ігор» – «Sudoku» та «Kakuro» (рис. 1.2).

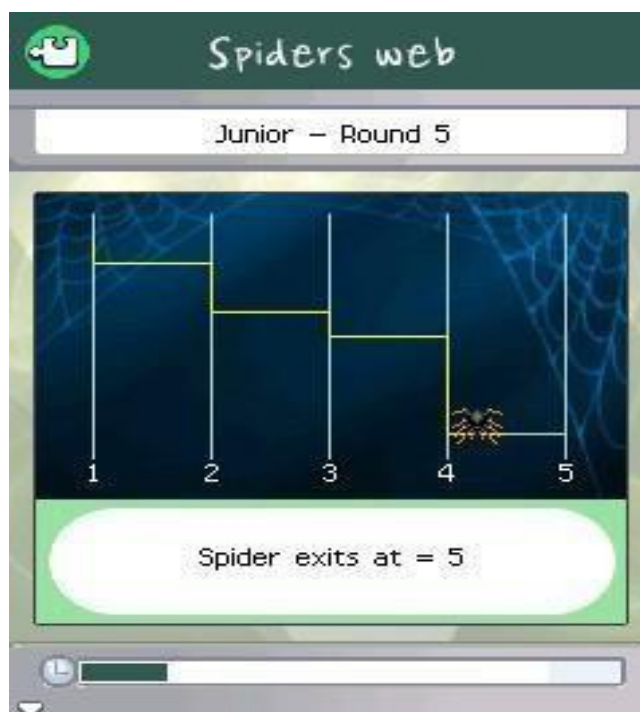


Рисунок 1.2 – Приклад тренування на логіку в «Brain Champ»

– Онлайн-гра «Іконт». Гра дозволяє покращити свої навички швидкого обрахунку. Додаток має мінімалістичний дизайн і в ньому присутні лише два режими – «Practice» та «Challenge». Потрібно вирішувати прості приклади на чотири арифметичні дії (в режимі «Challenge» наявний таймер) і таким чином підвищувати власний рівень та встановлювати рекорди. Додаток добре підходить для школярів молодших класів та стимулює вивчення ними арифметики (рис. 1.3).

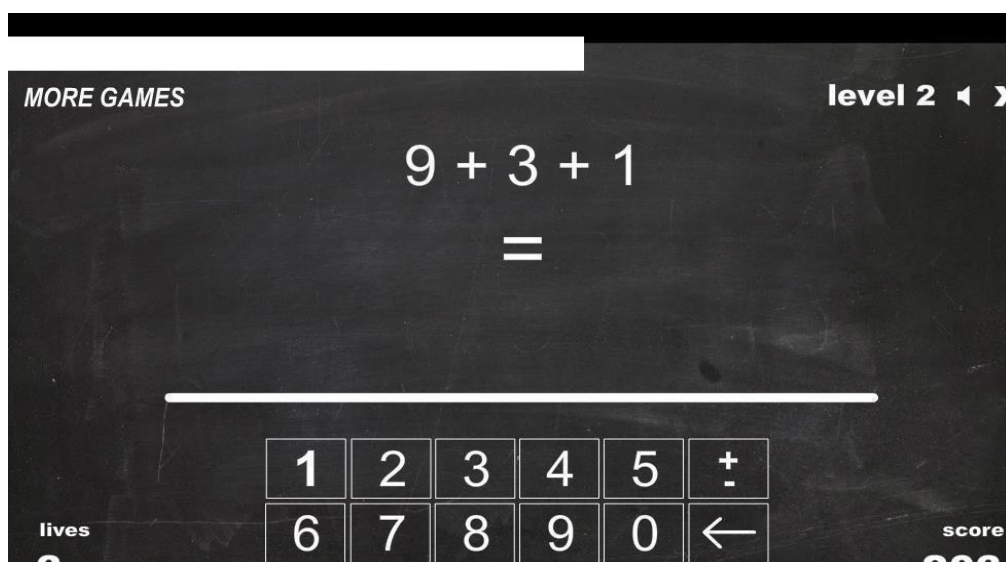


Рисунок 1.3 – Режим «Challenge» онлайн-гри «Іконт»

– Браузерний та мобільний ігровий додаток «2048». Гра починається на квадратному полі з 16-ти клітинок, де дві клітинки зайняті плитками номіналом «2» і «4». Кожну плитку можна перемістити по горизонталі чи вертикалі. Щоразу як гравець переміщує плитку, на полі з'являється додаткова плитка номіналу «2» (з ймовірністю 90 %) або «4» (з ймовірністю 10 %). Дві плитки одного номіналу, будучи поміщеними на одну клітинку, зливаються в одну, номінал якої дорівнює сумі злитих. Якщо в одній лінії після руху поряд стоять понад дві плитки одного номіналу, то вони зливаються автоматично. Кожному номіналу відповідає свій колір, що більший номінал, то «гарячіший» колір. За кожне злиття ігрові очки збільшуються на номінал новоутвореної плитки. Метою є отримати плитку

«2048», після чого дозволяється продовжити. Гра закінчується, якщо після чергового переміщення неможливо виконати жодне злиття плиток (рис. 1.4).



Рисунок 1.4 – Ігровий процес «2048»

### 1.3 Методи оцінювання ефективності застосування гейміфікації у навчанні

Методи колективної експертної оцінки засновані на виявленні узагальненої оцінки експертної групи шляхом аналізу та обробки індивідуальних незалежних оцінок експертів, що входять до складу групи.

В основі застосування методів колективної експертної оцінки лежить гіпотеза щодо наявності у експертів умінь оцінити з достатнім ступенем вірогідності: важливість і знання проблеми фактора, параметра, напряму розвитку, ознаки тощо; час здійснення тієї чи іншої події; значення параметрів, які прогнозуються; доцільність вибору одного з альтернативних шляхів розвитку об'єкта прогнозування та інші.

Методи колективної експертної оцінки за ознакою способу

отримання інформації від експертів умовно можна розподілити на дві великі групи: методи групової експертизи і методи анкетування.

Найбільш часто використовуються такі методи групової експертизи: експертних комісій, колективної генерації ідей (метод "мозкової атаки"), оперативних ігор.

Метод експертних комісій — це стратегія колективної експертної оцінки, де експерти об'єднуються для спільного визначення прогнозів чи прийняття рішень. Цей підхід спирається на принцип круглого столу, де кожен експерт вносить свої варіанти рішень і аргументує їх.

Під час засідань експертів у методі експертних комісій основна ідея полягає в тому, щоб досягти консенсусу або єдиної точки зору відносно розглянутих прогнозів або рішень. Важливою складовою є взаємне обговорення та обґрунтування пропозицій, що сприяє розумінню різноманітних підходів та поглядів.

Однак, цей метод може мати певні недоліки. Вироблення єдиної думки може ґрунтуватися на логіці компромісу, що призводить до ситуації, коли вплив окремих експертів, особливо тих, хто має великий авторитет, може переважати над іншими. Це може призвести до узурпації поглядів та надмірного впливу на прийняття рішення через авторитет окремих учасників.

Враховуючи ці моменти, важливо забезпечити баланс висловлених поглядів, враховуючи різноманітність експертних думок, а також розвивати механізми запобігання переважанню думки однієї особи над іншими в рамках колективного прийняття рішень.

Метод генерації ідей ("мозкових атак") заснований на стимулюванні творчої діяльності експертів шляхом сумісного обговорення конкретної проблеми. При цьому обговорення регламентується певними правилами: забороняється оцінка ідей, що висуваються; обмежується час одного виступу; припускаються багаторазові виступи кожного експерта; пріоритет щодо виступу має експерт, який розвиває попередню ідею; обов'язково

фіксуються всі ідеї; оцінка ідей, що висунуті, здійснюється на наступних етапах.

Метод "мозкових атак" або генерації ідей є потужним інструментом для збільшення креативності та продуктивності експертів у вирішенні проблеми. Основна ідея полягає в сприянні активному та безкритичному висуванню ідей, спонуканні до розвитку думок та стимулюванні творчості.

В процесі "мозкових атак" учасники зосереджуються на масовому висуванні ідей, уникненні будь-якої оцінки або критики, що дозволяє експертам вільно та необмежено висловлювати свої думки. Це сприяє появі нових, навіть нестандартних поглядів на проблему, розширенню горизонтів мислення та збагаченню ідейного спектру.

Однією з ключових особливостей методу є зосередженість на кількісних результатах, а не на оцінці або відборі ідей під час їх виникнення. Це дозволяє максимально розширити спектр можливих рішень та підходів до проблеми. Важливо також враховувати, що кожна ідея фіксується без будь-яких обмежень, навіть якщо вона здається непрактичною.

Завдяки "мозковим атакам" експерти можуть швидше та більш ефективно надавати творчі рішення для складних проблем, а також створювати плідне середовище для подальшої обробки та оцінки ідей на наступних етапах роботи.

Важливою частиною "мозкових атак" є створення атмосфери відкритості та взаєморозуміння, що стимулює обмін думками й ідеями між учасниками групи. Відсутність критики на етапі генерації ідей дає змогу кожному експерту вільно висловлюватись та вносити свій внесок у вирішення проблеми без страху перед оцінкою.

Метод "мозкових атак" базується на креативності та вільному потоці ідей, але має свої особливості. Він покликаний стимулювати творчий процес, проте при цьому передбачає певні правила та умови, щоб ефективно використовувати потенціал групи експертів:

1. Розуміння цілей. Учасники повинні розуміти мету і завдання

"мозкових атак", тобто генерацію максимальної кількості ідей для вирішення конкретної проблеми.

2. Заборона оцінки. Оцінка чи критика ідей заборонена на початкових етапах. Це дозволяє кожному експерту вільно висловлюватися та запропоновувати будь-які концепції.

3. Часові рамки. Встановлення обмеження на час для кожної ідеї. Це сприяє продуктивності та запобігає занадто довгим виступам.

4. Запис ідей. Всі ідеї фіксуються без винятку, навіть якщо на перший погляд вони здаються неперспективними.

5. Підкреслення спільності. Заохочується колективне сприйняття ідей. Один експерт може розвинути ідею, запропоновану іншим.

6. Аналіз і оцінка на наступних етапах. Оцінка та аналіз ідей відбувається на наступних етапах роботи над проблемою.

В цілому, "мозкові атаки" сприяють вільному висловленню ідей та стимулюють творчий процес, що дозволяє залучити різноманітні точки зору та концепції для досягнення оптимального рішення.

Метод "мозкових атак" розглядається як ефективний інструмент для стимулювання творчості та генерації ідей, проте він має свої унікальні аспекти, які важливо враховувати під час застосування. Важливо, щоб учасники розуміли, що мета цього методу - не критика чи оцінка ідей, а максимальне висловлення всіх можливих концепцій для вирішення конкретної проблеми. Обмеження в часі для виступу кожного допомагає уникнути занадто тривалих дискусій та сприяє концентрації на кожній ідеї. Аналіз та оцінка ідей зазвичай здійснюються на подальших етапах роботи над проблемою, що дозволяє уникнути приписування вартості конкретним концепціям на початковій стадії. Вільне висловлення ідей та підкреслення спільності під час розвитку концепцій забезпечують різноманітність поглядів та сприяють залученню різноманітних точок зору для досягнення оптимального рішення.

Відомі три різновиди методу колективної генерації ідей ("мозкової



атаки"): метод прямої групової експертизи ("пряма мозкова атака "), метод "групової згоди" і метод "оперативної творчості".

Метод прямої групової експертизи заснований на гіпотезі про те, що серед ідей, запропонованих експертами, є принаймні декілька добрих. Порядок проведення прямої групової експертизи такий: перед групою експертів різних спеціальностей ставиться проблема, яка може мати декілька варіантів рішень. Мета методу полягає в збиранні ідей щодо поставленої проблеми. Потім в декілька етапів проводиться їх аналіз, відбираються найбільш раціональні і виключаються непридатні. Найбільш часто цей метод застосовується на рівні зовнішньополітичних і стратегічних рішень, а також при воєнно-економічному і воєнно-технічному прогнозуванні.

Метод "групової згоди" заснований на проведенні групової експертизи з метою визначення згоди й єдності поглядів щодо прогнозованого питання. У цьому процесі беруть участь, як правило, шість експертів. Встановлено, що збільшення кількості експертів більше шести веде до збільшення часу та ускладнює одержання згоди, не збільшуючи суттєво кількості запропонованих ідей. При зменшенні кількості експертів можливо, що будуть враховані не всі обставини та знижений ступінь точності прогнозу.

Метод групової згоди активно використовується для обговорення та узгодження поглядів з метою прийняття спільного рішення. Це дозволяє уникнути екстремальних позицій окремих експертів та врахувати думки більшості учасників експертизи. Проте важливо враховувати, що цей метод може бути обмежений у вираженні різноманітності поглядів та стимулюванні критичного мислення, оскільки пошук компромісу може призвести до загального прийняття рішення, не враховуючи всіх альтернативних точок зору.

При роботі за методом групової згоди, слід враховувати, що цей підхід корисний для узгодження думок та уникнення великих розбіжностей, проте

він може потребувати більш довгих обговорень та компромісів для досягнення спільної точки зору.

Однак, слід враховувати, що багатоекспертний підхід має свої обмеження. Наприклад, різноманіття точок зору може бути обмежене через потребу досягнення згоди, що може звести до узгодженості на компромісний варіант. Крім того, багатоекспертна група може ускладнити процес ухвалення рішень через необхідність враховувати думки кожного учасника.

Загалом, метод "групової згоди" є корисним інструментом для досягнення спільної точки зору та уникнення радикальних позицій. Однак його ефективність може бути обмеженою у вирішенні складних або контроверсійних питань через необхідність згоди між учасниками.

Метод "оперативної творчості" припускає, що суть проблеми в цілому і можливі підходи до її вирішення відомі тільки керівнику роботи. У цьому випадку опитування експертів потрібне для того, щоб переконатися, що підхід до вирішення проблеми є правильний, а також отримати упевненість в роботі.

Цей метод зазвичай використовується у випадках, коли прийняття рішень потребує стрімкості та оперативності, і коли особа, яка відповідає за процес, володіє достатнім досвідом та компетентністю для оцінки ситуації. Опитування може використовуватися як засіб залучення підтримки для обраного рішення або стратегії, а також для переконання у його правильності та доцільності.

На відміну від методів, що базуються на колективному аналізі або обговоренні рішень, метод "оперативної творчості" ставить акцент на індивідуальному рішенні керівника або особи, яка відповідає за процес. Цей підхід передбачає, що саме ця особа має всі необхідні знання, досвід та кваліфікацію для прийняття правильних рішень щодо проблеми чи стратегії.

Експертне опитування в рамках цього методу використовується як інструмент для збору відповідей та підтвердження у відповідності обраного

шляху вирішення проблеми. Однак, це не завжди означає, що інші можливі альтернативи виключаються – скоріше, експертне опитування фокусується на переконанні у виборі і прийнятті обраного шляху.

Цей метод часто використовується в ситуаціях, коли потрібне негайне прийняття рішення та відсутній час на розгортання довгих обговорень чи колективного аналізу. Оскільки вирішення приймається на основі досвіду та знань однієї особи, цей метод може бути швидким та ефективним в контексті управління стратегічними питаннями або вирішення проблем швидкого реагування.

Однак, він може викликати обмеження у виборі альтернативних шляхів розв'язання проблеми та унеможливити врахування різноманітних точок зору.

Цей метод часто застосовується в управлінні ситуаціями кризового характеру, коли необхідно негайно прийняти рішення і час на широкий аналіз обмежений. Однак, він може бути неефективним у випадках, коли потрібно враховувати різноманітність точок зору та залучення великої кількості експертів для комплексного аналізу проблеми.

Гейміфікація у навчанні - це застосування елементів гри та геймінгу у навчальному процесі з метою стимулювання мотивації, залучення уваги та підвищення ефективності навчання. Для оцінки її ефективності можна використовувати різні методи:

1. Підвищення мотивації та участь у навчанні:

- Аналіз залучення: Вимірювання активності та участі учнів у навчальному процесі перед та після впровадження гейміфікації. Чи зросла активність учнів, чи збільшилася участь у завданнях через елементи гри?
- Опитування та анкети: Оцінювання думок учасників щодо ефективності гейміфікації, їхня мотивація та задоволеність процесом навчання.

2. Підвищення засвоєння матеріалу:

- Вимірювання успішності: Аналіз результатів учнів під час тестів або оцінка їхніх знань перед та після використання гейміфікації. Чи покращилися їхні результати?
- Спостереження за учнями: Оцінка взаємодії учнів з матеріалом під час виконання завдань гри. Чи активніше вони залучаються до вивчення матеріалу через ігрові елементи?

### 3. Стимулювання співпраці та командної роботи:

- Оцінка рівня співпраці: Аналіз того, як гейміфікація сприяє співпраці між учнями, чи підтримує командну роботу та спільні досягнення.
- Оцінка взаємодії: Спостереження та аналіз взаємодії між учнями під час гейміфікованих завдань чи викликів. Чи сприяє гра досягненню спільних цілей?

### 4. Вплив на зацікавленість учнів:

- Оцінка відновлення інтересу: Спостереження за змінами в зацікавленості учнів до предмета після використання гейміфікації.
- Аналіз динаміки участі: Вимірювання динаміки участі та зацікавленості під час навчального процесу з використанням гейміфікації.

Гейміфікація в навчанні не лише заохочує учнів активно брати участь у процесі навчання, але й сприяє збільшенню їхньої мотивації досягати певних цілей. Вона створює стимули для залучення уроків та самостійного опанування матеріалу. Ігрові елементи, такі як бали, рівні складності, або винагороди, можуть бути використані для створення чітких цілей та поетапного досягнення навчальних завдань.

Крім того, гейміфікація може поліпшити сприйняття матеріалу, оскільки вона надає можливість використовувати різноманітні методи інтерактивного навчання. Вона дозволяє вчителям використовувати інноваційні підходи та інструменти для ефективного передачі знань.

Застосування гейміфікації в навчанні вимагає гнучкості та уваги до потреб учнів, щоб відповідати їхнім особистим навчальним стилям та запитам. Ключовим фактором є збалансованість між використанням ігрових елементів та академічної основи, щоб досягти максимальної ефективності.

Гейміфікація в навчанні привертає увагу через свою ефективність у залученні учнів до вивчення. Вона може стати корисним інструментом для збільшення зацікавленості та участі, особливо в контексті сучасного світу, де великий обсяг інформації конкурує за увагу студентів.

Цей підхід впроваджує інтерактивність, яка додає елементів розваги до навчального процесу. При цьому важливо пам'ятати, що успішність гейміфікації в навчанні значною мірою залежить від здійснення балансу між розвагою та навчанням. Такий підхід вимагає вміння вчителів використовувати гейміфікацію як інструмент, який сприяє досягненню навчальних цілей.

Невідомість результатів застосування гейміфікації в навчанні на практиці в нинішній час може становити певні виклики. Однак, наявність позитивних свідчень та успішних прикладів використання цього методу свідчать про його потенційну ефективність у поліпшенні навчального процесу.

Гейміфікація в навчанні сприймається як потужний механізм для стимулювання мотивації, залучення учнів та збільшення ефективності процесу вивчення. Вона перетворює навчання на цікаву та захопливу пригоду, яка може значно полегшити засвоєння складних концепцій.

Основними складовими гейміфікації є використання ігрових елементів, таких як завдання, рівні складності, нагороди та лідерборди. Ці елементи не лише мотивують учнів, а й сприяють створенню здорової конкуренції та співпраці, що може стимулювати бажання вчитися та розвиватися.

Застосування гейміфікації в навчанні також може розвивати навички самодисципліни та самоконтролю. Через цікавість та емоційну залученість до

процесу, учні частіше беруть на себе відповідальність за власне навчання, що може призвести до покращення їхнього успіху в навчанні.

Ще однією перевагою гейміфікації є можливість персоналізації навчання. Вона дозволяє створювати індивідуальні шляхи для кожного учня, враховуючи їхні потреби, рівень знань та індивідуальні особливості, що сприяє більш ефективному та цікавому процесу навчання.

Гейміфікація в навчанні також може сприяти формуванню навичок розв'язання проблем та критичного мислення. Вона часто включає завдання, які потребують аналізу та прийняття рішень, що допомагає учням розвивати ці важливі навички.

Крім того, використання гейміфікації може підвищити рівень участі учнів у навчальному процесі. Ігрові елементи, такі як бейджі, рівні досягнень та нагороди, можуть заохочувати учнів до більш активного і стабільного виконання завдань, сприяючи позитивному настрою та задоволенню від навчання.

Такий підхід також може сприяти покращенню співпраці та комунікації між учнями. Колективні завдання та можливість спільної роботи над гейміфікованими завданнями спонукають учнів до співпраці, обміну думками та допомоги одне одному.

Гейміфікація в точних дисциплінах виявляється особливо цікавою через можливість застосування ігрових елементів для покращення навичок вирішення складних завдань. В предметах, де логіка, математика або наука важливі, гейміфікація може стимулювати учнів до розгадування складних проблем шляхом поетапних завдань, що надихає на аналіз та пошук ефективних рішень.

Гейміфікація в точних науках може включати симуляції, де учні мають можливість експериментувати з різними параметрами, виконувати реальні вимірювання або моделювати складні процеси. Такий підхід дозволяє краще зрозуміти та запам'ятати складні концепції через активну участь.

Також важливо відзначити, що гейміфікація в точних науках дозволяє

створити стимулююче навчальне середовище, де учні можуть вирішувати викликові завдання, отримувати бейджі та нагороди за досягнення та зацікавлення. Це сприяє залученню та мотивації учнів, оскільки вони бачать конкретні цілі та досягають їх через активну участь у навчальному процесі.

Гейміфікація в точних науках може також використовуватися для спрощення або ускладнення матеріалу залежно від рівня учня. Наприклад, у математиці можуть бути створені ігри або завдання на різні рівні складності для різних груп учнів. Це дозволяє кожному учневі працювати у своєму темпі та займатися матеріалом, що відповідає його поточному рівню знань.

Крім того, графічні симуляції та інтерактивні вправи можуть сприяти кращому усвідомленню складних фізичних або хімічних концепцій. Це дозволяє учням бачити та відчувати вплив змін у параметрах системи, що може значно полегшити розуміння складних аспектів науки.

Застосування гейміфікації в точних дисциплінах також може сприяти розвитку проблемного мислення та критичного аналізу. Учні, вирішуючи завдання, які вимагають логічного мислення та аналізу даних, розвивають важливі навички, необхідні для успішного опанування точних наук.

Гейміфікація в точних дисциплінах може стимулювати учнів до самостійного дослідження. Наприклад, у фізиці чи біології можуть створюватися інтерактивні завдання, які дозволяють учням відчути себе дослідниками, експериментаторами. Вони можуть вивчати закони природи через власні експерименти та віртуальні моделі, що сприяє глибокому розумінню концепцій.

Також, графічні представлення складних математичних або фізичних принципів можуть полегшити їх сприйняття. Через використання візуалізації учні можуть більш чітко уявляти та розуміти складні абстрактні концепції, такі як тригонометрія у геометрії або молекулярна структура у хімії. Такий підхід робить процес навчання більш доступним та зрозумілим для учнів, допомагаючи їм легше засвоювати матеріал.

Підсумовуючи, гейміфікація в навчанні є перспективним

інструментом, що може стати цікавим доповненням до сучасних методів навчання. Однак, для досягнення найбільшої користі від цього підходу, необхідно продовжувати досліджувати та адаптувати його до конкретних потреб навчального середовища та учнівських груп.

Загальний метод оцінки ефективності дозволяє порівняти нову розробку з уже існуючими аналогами в різних аспектах. Він стає корисним інструментом у випадках:

1. Покращення продукту: Дозволяє розробникам зрозуміти, наскільки їхні нові системи або продукти переважають над існуючими аналогами. Це дозволяє ідентифікувати сильні та слабкі сторони нової розробки.

2. Оцінка конкурентоспроможності: Порівняння з аналогами дає змогу з'ясувати, наскільки ваш продукт вигідний на ринку порівняно з конкурентами. Це може допомогти визначити стратегію позиціонування продукту на ринку.

3. Удосконалення маркетингових стратегій: Розуміння переваг нової системи в порівнянні з аналогами дозволяє створювати маркетингові повідомлення, які акцентують на основних перевагах вашого продукту.

4. Покращення користувацького досвіду: Аналізуючи різні аспекти (наприклад, зручність роботи, відсутність багів, ефективність вивчення дисциплін, графічний інтерфейс), можна зрозуміти, як саме поліпшити користувацький досвід вашого продукту.

Обґрунтуємо розрахунок даного методу.

- Середня оцінка розробленої системи в кожній категорії через  $R_s$ .
- Середня оцінка кожного аналогу в категорії через  $R_a$ .

Тоді різниця між середніми оцінками системи та аналогів  $\Delta R$  у кожній категорії буде виражена як  $\Delta R = R_s - R_a$ .

Щоб виразити відсоток, на який розроблена система краща за аналоги, можна скористатися формулою:

$$\% \text{Кращості} = \Delta R / R_a \times 100\%.$$



## 1.4 Постановка задачі дослідження

Постановка задачі - це опис питання, яке потрібно вирішити або умови, яку потрібно покращити. Воно визначає розрив між поточною проблемою та ціллю. Постановка задачі має відповідати п'яти "W" - хто, що, коли, де, чому. Перша умова вирішення задачі - розуміння проблеми, що може бути зроблено через постановку задачі.

Постановка задачі використовується більшістю підприємств та організацій для виконання проектів з покращення процесів. Основна мета постановки задачі - ідентифікувати та пояснити проблему. Це включає опис поточного середовища, де виникає проблема, та її впливу. Крім того, постановка задачі використовується для пояснення того, як повинне виглядати очікуване середовище.

Ще одна важлива функція постановки задачі - служити засобом комунікації. Постановка задачі допомагає отримати підтримку від тих, хто задіяний у проекті. Перед початком проекту зацікавлені сторони перевіряють, чи коректно описані проблема та цілі у постановці задачі. Після отримання цього затвердження команда проекту переглядає це, щоб кожен зрозумів суть проблеми та що вони намагаються досягнути. Це також допомагає визначити обсяг проекту, що допомагає сконцентруватися на загальній меті.

Постановка задачі використовується під час всього проекту для установлення фокусу в команді проекту та підтвердження, що вони залишаються на правильному шляху. В кінці проекту до неї повертаються, щоб підтвердити, що впроваджене рішення дійсно вирішує проблему. Чітко визначена постановка задачі також може допомогти в проведенні аналізу причин, щоб з'ясувати, чому виникла проблема та забезпечити можливість прийняти заходи для її уникнення в майбутньому.

Важливо зауважити, що постановка задачі не визначає рішення або методи досягнення рішення. Вона просто визначає розрив між проблемою

та ціллю.

Перш ніж буде сформульовано постановку задачі, необхідно визначити саму проблему.

Процес визначення проблеми часто включає участь групи. Він розпочинається зустріччю з зацікавленими сторонами, клієнтами або користувачами, які стикаються з цією проблемою (якщо це можливо), та вивченням їх проблем. Оскільки люди часто мають складнощі з ефективним спілкуванням своїх проблем, особливо з кимось, хто не розуміє процесу, корисно задавати серію запитань "чому", поки не буде виявлено основне обґрунтування. Цей метод, відомий як "п'ять чому", допомагає проникнути до основної проблеми, оскільки багато відчутних роздратувань можуть бути лише симптомами фактичної проблеми. Поставляючи ці додаткові питання та парафразуючи сказане зацікавленими сторонами, демонструється рівень співчуття та розуміння проблеми.

Інформація, зібрана під час цих початкових інтерв'ю, це лише одна частина аналізу проблеми. Часто проблема поширюється на кілька сфер або функцій, які зацікавлені сторони, клієнти та користувачі не усвідомлюють. Вони також можуть бути знайомі з тим, що відбувається на поверхні, але не обов'язково знають справжню причину. Тому так само важливо збирати знання, інформацію та висновки від членів команди проєкту та експертів у цій галузі щодо проблеми. Можливо, знадобиться також додаткова дослідницька матеріальна база, така як інструкції з роботи, користувацькі посібники, технічні характеристики продукту, діаграми робочих процесів та попередні плани проєкту. Як і більшість інших етапів проєкту по вдосконаленню процесу, визначення проблеми часто є ітеративним, оскільки може знадобитися кілька раундів обговорень, щоб отримати повну картину.

Після того, як проблема зрозуміла, і обставини, що визначають початок проєкту, чіткі, прийшов час для написання постановки задачі.

Постановка задачі буде використовуватися для отримання підтримки проєкту та затвердження зацікавлених сторін. Отже, вона повинна бути

спрямована на дії. Важливо, щоб постановка задачі була написана чітко та точно для досягнення успішних результатів. Неправильно сформульована або неточна постановка задачі призведе до неправильного рішення, а також витрат часу, грошей та ресурсів.

Є кілька основних елементів, які можна включити до кожної постановки задачі, щоб зменшити ризик невдачі проєкту. По-перше, постановка задачі повинна фокусуватися на кінцевому користувачеві. Часта помилка полягає в тому, що акцент робиться на тому, як буде вирішено проблему, а не на поточній прогалині. По-друге, постановка задачі не повинна бути занадто широкою. Перевагою використання підходу "п'ять чому" є уникнення занадто спрощеного підходу, надаючи необхідні деталі для розуміння проблеми та розробки відповідного рішення. Нарешті, постановка задачі не повинна бути занадто вузькою. Упередженість до рішень стримує творчість, яка виникає під час мозкового штурмування рішення, що може призвести до гіршого досвіду для користувача.

Корисно створити й дотримуватися певного формату при написанні постановки задачі. Хоча є кілька варіантів цього, ось простий та прямий шаблон, який часто використовується в бізнес-аналізі для збереження фокусу на визначенні проблеми.

Ідеал. Цей розділ використовується для опису бажаного або "майбутнього" стану процесу або продукту. Він визначає цілі зацікавлених сторін та клієнтів, а також допомагає у визначенні обсягу. У цілому, цей розділ повинен ілюструвати, яким буде очікуване середовище після впровадження рішення. Реальність: Цей розділ використовується для опису поточного або "наявного" стану процесу або продукту. Він пояснює труднощі, висловлені зацікавленими сторонами та клієнтами. Він також повинен включати інсайти та експертність команди проєкту та експертів у цій галузі, надані під час аналізу проблеми. Наслідки: Цей розділ використовується для опису впливу на бізнес, якщо проблему не виправити чи не поліпшити. Це включає витрати, пов'язані з втратою грошей, часу, продуктивності, конкурентних переваг та інше. Величина цих наслідків

також допоможе визначити пріоритет проєкту. Пропозиція: Цей розділ використовується для опису потенційних рішень. Після завершення, розуміння та затвердження розділів "ідеал", "реальність" та "наслідки", команда проєкту може почати пропонувати варіанти вирішення проблеми. Він також може включати пропозиції зацікавлених сторін та клієнтів, хоча для визначення конкретного плану дій потрібні подальші обговорення та дослідження.

Необхідно розробити технічне завдання для системи з наступними основними вимогами:

1. Підтримка файлової системи. Система повинна мати можливість працювати з файлами, зокрема зберігати та завантажувати тестові дані. Підтримка декількох режимів роботи:

- Проходження вікторини за певний час.
- Проходження вікторини з врахуванням кількості чекпойнтів.
- Проходження вікторини в режимі "відповідь є у питанні".

2. Підтримка до 100 питань. Система повинна бути здатна працювати з великою кількістю питань без проблем з продуктивністю.

3. Висока масштабованість. Система повинна витримувати навантаження від багатьох користувачів, які одночасно проходять вікторини.

4. Реалізація привітань в кінці гри. Після закінчення вікторини, система повинна надавати різні привітання залежно від відповідей користувача.

5. Підтримка ОС Windows. Система повинна бути сумісною з операційною системою Windows.

Додатково, система повинна мати такі можливості:

- Доступ для будь-якого користувача без необхідності реєстрації та авторизації.
- Завантаження тестових даних у текстовому форматі шляхом їх попереднього створення та форматування.

Виконання цих вимог дозволить розробити систему гейміфікації для вивчення основ математики.

Ця система гейміфікації для вивчення математики буде створена з урахуванням ключових потреб користувачів. Однією з головних функцій є підтримка файлової системи, що дозволить зручно зберігати та завантажувати тестові дані. Це важливо, оскільки гнучкість у керуванні контентом гри є ключовою для забезпечення актуальності та різноманітності запитань.

Додатково, система буде функціонувати у декількох режимах, що дасть можливість гравцям обирати спосіб проходження вікторини - чи то за певний час, чи за кількістю питань, чи у режимі "відповідь є у питанні". Це дозволить створити індивідуальний досвід для різних типів користувачів.

Однак, щоб забезпечити комфортну роботу системи, вона має підтримувати до 100 питань та бути готовою працювати із великою кількістю користувачів одночасно. Це забезпечить ефективність та збереже продуктивність під час гри.

Окрім цього, система надасть вітальні повідомлення в кінці гри, додатково заохочуючи гравців та підкреслюючи їхні досягнення. Для широкого кола користувачів, система буде доступною на операційній системі Windows, що забезпечить широку доступність та зручність у використанні.

## **1.5 Висновок до розділу 1**

Створення додатку гейміфікації для вивчення основ математики є обґрунтованим з урахуванням проведеного аналізу предметної області систем гейміфікації в освітньому профілі. Оцінено особливості та принципи таких систем. Також були проаналізовані програми-аналоги, які пропонують навчання через гру, рейтинги та нагороди, спонукаючи до активного вивчення. Проте виявлено недоліки цих систем, такі як обмежена універсальність додатків, необхідність авторизації та складність інтерфейсу. В результаті поставлено задачу дослідження та сформульовано основні вимоги до розробки програмного забезпечення для гейміфікації вивчення основ математики. Також обґрунтовано метод оцінювання ефективності ПЗ.

## 2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ГЕЙМІФІКАЦІЇ ДОДАТКУ ДЛЯ ВИВЧЕННЯ ОСНОВ МАТЕМАТИКИ

### 2.1 Проектування UML діаграм

UML – мова графічного опису для об'єктного моделювання в області розробки програмного забезпечення, моделювання бізнес-процесів, системного проектування та відображення організаційних структур.

UML є мовою широкого профілю, це – відкритий стандарт, який використовує графічні позначення для створення абстрактної моделі системи. UML був створений для визначення, візуалізації, проектування та документування програмних систем. UML не є мовою програмування, але на підставі UML-моделей можлива генерація коду.

Для розробки програмного додатку буде використовуватись з усіх видів діаграм, наявних в UML, діаграма класів. Діаграма класів (Class diagram) - статична структурна діаграма, що описує структуру системи, демонструє класи системи, їх атрибути, методи і залежності між класами.

Існують різні точки зору на побудову діаграм класів в залежності від цілей їх застосування:

- концептуальна точка зору – діаграма класів описує модель предметної області, в ній присутні тільки класи прикладних об'єктів;
- точка зору специфікації – діаграма класів застосовується при проектуванні інформаційних систем;
- точка зору реалізації – діаграма класів містить класи, використовувані безпосередньо в програмному коді (при використанні об'єктно-орієнтованих мов програмування).

Для проектування даного програмного додатку найдоцільнішим буде обрати застосування діаграми класів з точки зору реалізації.

Далі необхідно застосувати відомі знання про типи взаємозв'язків між класами. Взаємозв'язки надзвичайно полегшують процес розробки додатку, вони надають можливість відслідкувати певні залежності між класами, побачити діаграму класів як єдине ціле. Розглянемо основні типи зв'язків. Асоціація – це використання об'єктом одного класу об'єкт іншого. Композиція – зв'язок схожий на асоціативний, за винятком того, що створення об'єкту класу, який використовуються, виконується безпосередньо в першому класі. Залежність – підвид композиції; об'єкт створюється в певному методі. Генералізація – розширення одного класу іншим (наслідування). Реалізація – реалізація певним класом методів, які задані в інтерфейсі.

При проектуванні відносин між класами треба враховувати деякі загальні рекомендації. Зокрема, замість успадкування слід надавати перевагу композиції, а замість композиції – агрегації.

Після реалізації всіх необхідних зв'язків в додатку можна переходити до написання лістингу програм окремих класів, компонування компонентів програми в єдине ціле, обробки виключень.

UML є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, яка називається UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

UML може бути застосовано на всіх етапах життєвого циклу аналізу бізнес- систем і розробки прикладних програм. Різні види діаграм які підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, такі ділових систем.

Діаграми дають можливість представити систему (як ділову, так і

програмну) у такому вигляді, щоб її можна було легко перевести в програмний код.

Основною причиною використання мови UML є спілкування розробників між собою.

Крім того, UML спеціально створювалася для оптимізації процесу розробки програмних систем, що дозволяє збільшити ефективність їх реалізації у кілька разів і помітно поліпшити якість кінцевого продукту.

UML чудово зарекомендувала себе в багатьох успішних програмних проєктах. Засоби автоматичної генерації кодів дозволяють перетворювати моделі мовою UML у вихідний код об'єктно-орієнтованих мов програмування, що ще більш прискорює процес розробки.

Практично усі CASE-засоби (програми автоматизації процесу аналізу і проєктування) мають підтримку UML. Моделі розроблені в UML, дозволяють значно спростити процес кодування і направити зусилля програмістів безпосередньо на реалізацію системи.

Діаграми підвищують супроводжуваність проєкту і полегшують розробку документації.

UML необхідний:

- керівникам проєктів, які керують розподілом завдань і контролем за проєктом
- проєктувальникам інформаційних систем які розробляють технічні завдання для програмістів;
- бізнес-аналітикам, які досліджують реальну систему і здійснюють інжиніринг і реінжиніринг бізнесу компанії;
- програмістам які реалізують модулі інформаційної системи.

При модифікації системи об'єктний підхід дозволяє легко включати в систему нові об'єкти і виключати застарілі без істотної зміни її життєздатності. Використання побудованої моделі при модифікаціях системи дає можливість усунути небажані наслідки змін, оскільки вони не ламають структури системи, а тільки змінюють поведінку об'єктів.



Діаграма класів (англ. class diagram) — статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення. Діаграма класів може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм.

Діаграма класів служить для представлення статичної структури моделі системи в термінології класів об'єктноорієнтованого програмування. На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини.

Для того, щоб реалізувати інформаційну технологію гейміфікації додатку для вивчення основ математики необхідно розробити такі діаграми: діаграму класів, діаграму прецедентів.

До розробленої діаграми класів належать наступні класи: App, Expression, Expression Content, Functions, FunctionsContent, LinSystem, LinSystemContent, MainWindow, QuadEquation, QuadEquationsContent, Test Content. Основний клас MainWindow відповідає за запуск додатку та взаємодію користувача з графічним інтерфейсом. Класи Expression, Expression Content, Functions, FunctionsContent, LinSystem, LinSystemContent, QuadEquation, QuadEquationsContent відповідають за чотири базові види завдань та їх реалізацію в системі. Загалом, серед зв'язків між класами використовується успадкування та генералізація.

-public partial class App – відповідає за запуск додатку засобами WPF;

*Клас Expression*

-public void Generate – генерує задану користувачем кількість прикладів;

-public bool isCorrect – перевіряє відповідь користувача на коректність;

-public string Show – відповідає за виведення прикладів на екран;

*Клас ExpressionsContent*

-public string Expression – виведення певного прикладу на екран;

-public string Score – виведення набраних гравцем балів на екран;  
 -private void Generate\_Click – відповідає за введення користувачем коректних даних для кількості генерованих прикладів;  
 -private void Answer\_Click – відповідає за надання коректної відповіді гравцем;

#### *Клас Functions*

-public string answer – надання відповіді гравцем скільки потрібно згенерувати функцій;  
 -public string Generate()– генерує задану користувачем кількість функцій;  
 -public bool IsCorrect – перевіряє запит користувача на коректність;

#### *Клас FunctionsContent*

-public string Function – виведення певної функції на екран;  
 -public string Score – виведення набраних гравцем балів на екран;  
 -private void Generate\_Click – відповідає за введення користувачем коректних даних для кількості генерованих прикладів;  
 -private void Answer\_Click – відповідає за надання коректної відповіді гравцем;

#### *Клас LinSystem*

-private int x, y – задання двох змінних СЛАР;  
 -private int a, b, c, d – задання коефіцієнтів змінних;  
 -public int res1, res2 – ініціалізація результатів СЛАР;  
 -public void Generate() – генерація СЛАР;  
 -public string Show() – виведення на екран СЛАР;  
 -public int IsCorrect – перевірка відповіді наданої гравцем на коректність;

#### *Клас LinSystemContent*

-public string LinSystem – виведення певної СЛАР на екран;  
 -public string Score – виведення набраних гравцем балів на екран;

#### *Клас MainWindow*

-public MainWindow – відповідає за графічний інтерфейс головного вікна програми;

-private void Expressions\_Click – відповідає за вибір користувачем арифметичних прикладів;

-private void QuadraticEquations\_Click – відповідає за вибір користувачем квадратних рівнянь;

-private void LinSystem\_Click – відповідає за вибір користувачем СЛАР;

-private void Functions\_Click – відповідає за вибір користувачем функцій;

-private void Exit\_Click – відповідає за вихід з системи;

*Клас QuadEquation*

-public double D, x1, x2, a, b, c – відповідає за відповідні змінні та коефіцієнти квадратного рівняння;

-public string answer – надання гравцем відповіді;

-public void GenerateRandom – генерація квадратного рівняння;

-public bool IsDSqrt – перевірка на корінь у детермінанта;

-public string Show() – виведення на екран квадратного рівняння;

*Клас QuadEquationsContent*

-public string QuadEquation – виведення певного квадратного рівняння на екран;

public string Score – виведення набраних гравцем балів на екран;

*Клас TestContext*

-public TestContext – логіка взаємодії програми з XAML.

Ці класи складають основу програми для математичної вікторини з використанням платформи WPF (Windows Presentation Foundation). Ось короткий опис кожного класу:

1. App. Цей клас відповідає за запуск додатку засобами WPF. Він управляє головними властивостями та функціоналом програми.
2. Expression. Клас, що містить методи для генерації математичних

виразів, перевірки правильності відповідей користувача та виведення прикладів на екран.

3. ExpressionsContent. Він відповідає за зберігання та відображення певних виразів та балів гравця. Містить методи для генерації та обробки введених даних.

4. Functions. Цей клас містить функції для генерації та перевірки функцій, введених користувачем, і взаємодії з ними.

5. FunctionsContent. Аналогічно до ExpressionsContent, але для функцій.

6. LinSystem. Він містить логіку для генерації та відображення систем лінійних алгебраїчних рівнянь (СЛАР) і перевірки правильності відповідей користувача.

7. LinSystemContent. Аналогічно до інших Content-класів, але для систем лінійних рівнянь.

8. MainWindow. Клас, що відповідає за графічний інтерфейс головного вікна програми. Містить обробники подій для різних опцій вибору гравця та логіку взаємодії з іншими класами.

9. QuadEquation. Він відповідає за генерацію, перевірку та відображення квадратних рівнянь.

10. QuadEquationsContent. Аналогічно до інших Content-класів, але для квадратних рівнянь.

11. TestContext. Цей клас містить логіку взаємодії програми з XAML, що використовується для розмітки та дизайну графічного інтерфейсу.

Визначимо основні класи, що будуть наявні в структурі системи, що розробляється та будемо по чергово їх створювати. Необхідно надати кожному з класів потрібні методи, властивості, побудувати зв'язки між ними, визначити головні класи та другорядні.

Результатом моделювання буде розроблена UML-діаграма, що міститиме в собі одинадцять класів та одну структуру (рис. 2.1). У побудованій ієрархії зв'язків найпоширенішим типом відносин є асоціація.

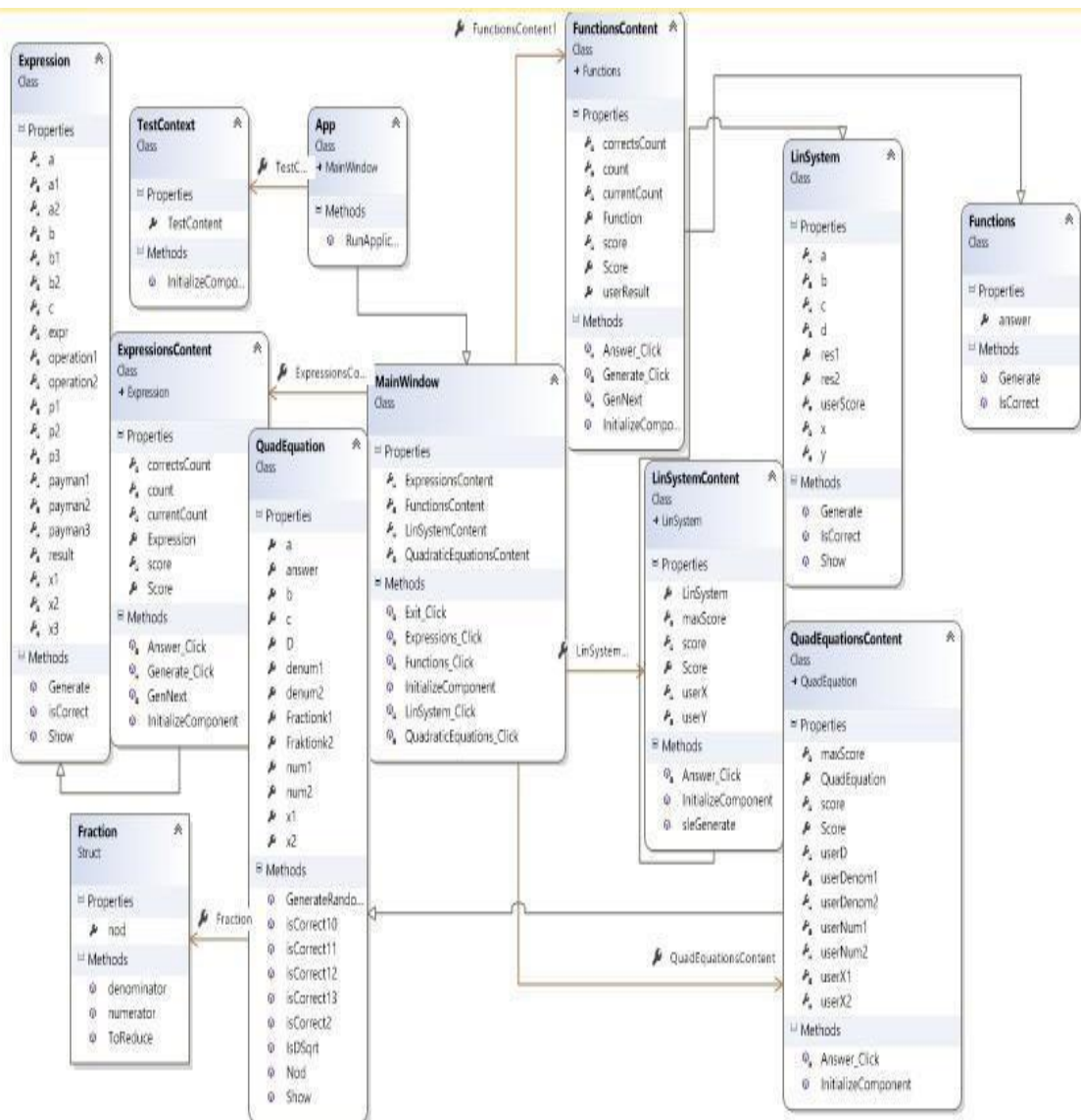


Рисунок 2.1 – UML-діаграма класів ігрової математичної програми

Спроектуємо також діаграму прецедентів, зображену на рисунку 2.2.

Діаграми прецедентів – це важливий інструмент у системному аналізі, що використовується для моделювання взаємодії між користувачами системи та самою системою. Ці діаграми відображають функціональність системи через визначення основних дій (прецедентів), які можуть бути виконані її користувачами.

У них кожен прецедент представляє конкретну дію або функцію, яку може виконати користувач. Це може бути звернення до системи для отримання певної послуги, обробки інформації чи взаємодії з іншими користувачами.

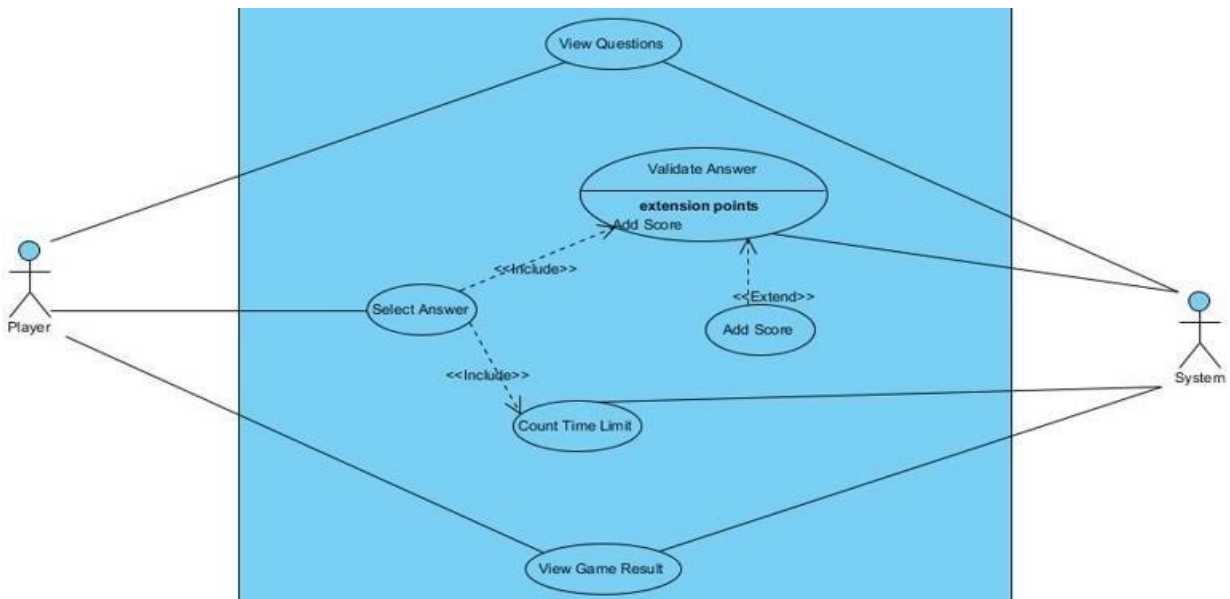


Рисунок 2.2 – Діаграма прецедентів ігрової математичної програми

Діаграми прецедентів складаються з прецедентів (дій користувачів), акторів (ролей, які взаємодіють з системою), а також зв'язків між ними. Ці зв'язки показують, які прецеденти доступні для кожного актора і як вони взаємодіють між собою.

Завдяки діаграмам прецедентів можна чітко визначити функціональність системи, зрозуміти потреби користувачів та встановити основні етапи взаємодії між ними та системою. Це сприяє кращому розумінню вимог до системи та визначенню основних сценаріїв її використання

Діаграми прецедентів допомагають у визначенні потреб користувачів та визначенні функціональних вимог до системи. Вони спрощують комунікацію між розробниками системи та її користувачами, оскільки графічне зображення прецедентів дозволяє легше сприймати взаємодію між системою та користувачами.

Ці діаграми допомагають уникнути непорозумінь щодо очікувань від системи, оскільки вони чітко показують, які дії можуть бути виконані кожним користувачем та які функції доступні в межах системи. Крім того, вони можуть служити основою для подальшої розробки системи, дозволяючи команді проекту краще розуміти, як система повинна працювати та які

можливості вона має надати користувачам.

Використання діаграм прецедентів у системному аналізі сприяє покращенню якості проектування системи, дозволяючи врахувати потреби користувачів та визначити ключові функції системи ще на ранніх етапах розробки. Діаграми використовуються для уточнення вимог до системи шляхом виділення ключових акторів (стейкхолдерів) та прецедентів, які представляють сценарії взаємодії між користувачами та системою. Актори можуть бути різними типами користувачів, такими як адміністратори, клієнти або зовнішні системи, тоді як прецеденти відображають конкретні дії, які вони можуть виконати. Ця методика моделювання дозволяє не лише ідентифікувати основні функції системи, але й визначити залежності між різними прецедентами, що сприяє кращому розумінню логіки роботи програми та розробці зв'язків між різними її частинами.

Крім того, діаграми прецедентів використовуються не лише на етапі проектування, але й можуть служити базою для тестування системи, оскільки вони дозволяють визначити сценарії та очікувані результати взаємодії користувачів з системою.

Діаграми прецедентів допомагають створити загальне уявлення про те, як користувачі будуть взаємодіяти з системою. Це важливо для розробки ефективного та зручного для використання програмного забезпечення. Вони дозволяють виявити потреби користувачів, описати, яким чином користувачі взаємодіють із системою, та визначити основні функціональні можливості системи.

Кожен прецедент - це конкретний сценарій взаємодії між користувачем і системою. Вони описують, як користувачі використовуватимуть систему для досягнення своїх цілей та як система реагуватиме на їх дії. Це можуть бути дії вводу даних, перегляду інформації, обробки запитів тощо.

Актори - це ролі або сутності, які взаємодіють з системою. Це можуть бути різні типи користувачів, зовнішні системи або навіть інші програми. Вони використовують прецеденти для досягнення своїх мет.

Діаграми прецедентів допомагають команді розробників легше розуміти потреби користувачів, визначити пріоритети функціональності та логіку взаємодії між користувачами та системою. Вони стають основою для подальшого аналізу системи, проектування та реалізації програмного забезпечення.

У системі математичної ігрової вікторини, діаграма прецедентів розглядає взаємодію двох основних учасників: гравця та системи.

1. Гравець:

- Проходження вікторини. Гравець може взаємодіяти з системою, вибираючи теми або типи завдань для вікторини (наприклад, арифметика, квадратні рівняння, системи лінійних рівнянь).
- Надання відповідей. Гравець вводить свої відповіді на питання або завдання, які генеруються системою.

2. Система:

- Генерація питань: Система генерує математичні завдання для вікторини відповідно до вибору гравця та обраної теми.
- Перевірка відповідей. Система аналізує введені гравцем відповіді на коректність та оцінює їх правильність.
- Надання результатів. Після завершення вікторини система показує гравцеві його результати та набрані бали.

Діаграма прецедентів демонструє взаємодію між гравцем та системою, визначаючи основні функції кожного учасника та їх способи взаємодії. Це важливий інструмент при проектуванні системи, що допомагає зрозуміти, як користувачі будуть використовувати програму та як вона буде взаємодіяти з їхніми діями. В нашій діаграмі прецедентів є всього два актори – власне користувач (гравець) і сама система. Гравець може переглядати запитання, обирати відповідь, перевіряти відповіді та отримувати результат вікторини, а система може робити рахунок відповідям і обраховувати результат, також перевіряти відповіді гравця на коректність.



## 2.2 Розробка алгоритмів функціонування інформаційної технології гейміфікації вивчення основ математики

Алгоритм функціонування системи “Brain Strain” має досить просту структурну схему і складається з декількох великих модулів (рис.2.3):

- модуль Main Window;
- модуль графічного інтерфейсу на основі WPF;
- модуль App, що відповідає за запуск програми;
- модуль Expressions;
- модуль Quadratic Equations;
- модуль Linear Systems;
- модуль Functions.

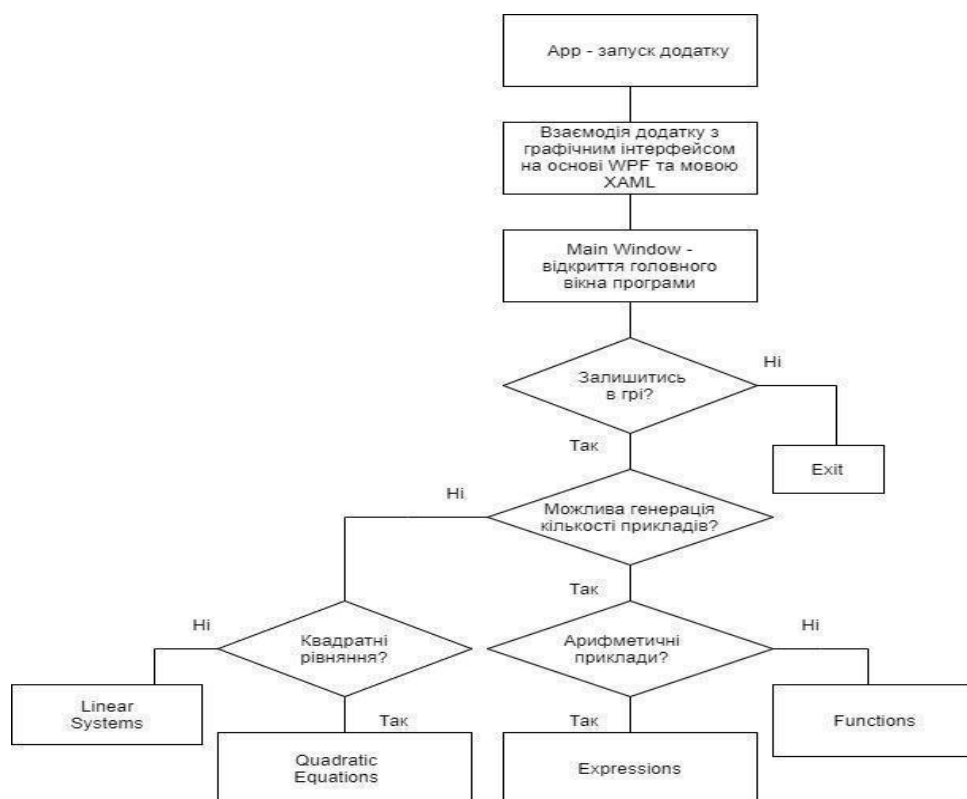


Рисунок 2.3– Алгоритм функціонування інформаційної технології гейміфікації вивчення основ математики

Спершу користувач запускає додаток, потім встановлює взаємодію з графічним інтерфейсом користувача через бібліотеку WPF, після чого обирає

один з чотирьох можливих класів завдань.

На рисунку 2.4 можна побачити відображення роботи системи з програмної точки зору, а саме запуск головного модуля, після чого відбувається взаємодія з графічним інтерфейсом користувача, запуск модулів обробників подій та класів математичних задач, відбувається генерація прикладів і розрахунок результату вікторини системою.



Рисунок 2.4 – Алгоритм функціонування основних модулів системи

На рисунку 2.5 можна побачити структурну схему роботи додатку гейміфікації. Дана структурна схема складається з блоків винагорода,

ексклюзивна винагорода, статус, ціль та зв'язки. Кожен з цих блоків містить підблоки, що є структурними частинами базових одиниць. Одиниця Ціль відповідає за призначення роботи додатку, зв'язки – за коректність зв'язків інтерфейсу користувача з користувачем. До розділу статус належить статистика і бонуси, до лояльності – ексклюзивні винагороди, а до винагород – очки і статуси-бейджі.



Рисунок 2.5 – Структура інформаційної технології гейміфікації вивчення основ математики

### 2.3 Висновок до розділу 2

В даному розділі спроектовано простий алгоритм роботи додатку гейміфікації основ математики, що враховує всі необхідні елементи, які мають бути залучені до навчального процесу, а також окреслено ряд методик, що задовольняють всі необхідні стандарти навчання, для проведення тестування майбутнього додатку, який буде створено згідно зі всіма поставленими вимогами та деталями до технічного завдання. Побудовано декілька наочних UML-діаграм, що відображають елементи та можливості майбутнього додатку, для кращого розуміння структури програми та візуалізації потреб користувача, який буде брати участь у навчальному процесі з використанням технології гейміфікації та додатку, що ці технології вводить в навчальний процес.

### 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ГЕЙМИФІКАЦІЇ ДЛЯ ВИВЧЕННЯ ОСНОВ МАТЕМАТИКИ

#### 3.1 Обґрунтування середовища і мови програмування

Існує безліч мов програмування, але, незважаючи на таку розмаїтість, кількість мов, які широко використовуються, насилу перевищує за десяток. Найпопулярнішими мовами на даний момент є Java, C#, JavaScript та PHP. Найбільш розповсюджені мови програмування це: C#, Java, C/C++, PHP [6].

З усього різноманіття мов програмування кожен з обирає ту мову, яка для нього є найбільш зручною й краще підходить для його роботи.

Як демонструють дані (рис. 3.1), C# є однією з найпопулярніших й найкращих мов програмування в наш час.



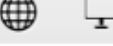





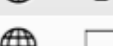

Language Rank	Types	Spectrum Ranking
1. C		100.0
2. Java		98.1
3. Python		98.0
4. C++		95.9
5. R		87.9
6. C#		86.7
7. PHP		82.8
8. JavaScript		82.2
9. Ruby		74.5
10. Go		71.9

Рисунок 3.1 - Рейтинг мов програмування за версією журналу IEEE Spectrum

Для себе я обрав мову програмування C# тому, що це об'єктно-орієнтована мова програмування з безпечною системою типізації. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події,

властивості, винятки, коментарі у форматі XML. Переїнявши багато що від своїх попередників – мов C++, Delphi, Модула і Smalltalk – C#, спираючись на практику їхнього використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем: так, C# не підтримує множинне спадкування класів (на відміну від C++) або виведення типів (на відміну Haskell).

C# — це мова програмування, яка поєднує багато характеристик із попередніми мовами, але також впроваджує власні особливості, спрямовані на полегшення розробки програмних систем. Вона базується на об'єктно-орієнтованому підході і відрізняється від інших мов, наприклад, відсутністю множинного спадкування класів, що спрощує структуру програм та дозволяє уникнути деяких складнощів, які можуть виникнути при розробці.

C# має строгу статичну типізацію, що означає, що типи даних визначаються на етапі компіляції, що дозволяє виявляти помилки на ранніх етапах розробки. Ця мова підтримує поліморфізм, що дозволяє створювати методи з однаковими назвами, але з різною поведінкою в залежності від контексту використання. Також C# має підтримку перевантаження операторів, атрибутів, подій, властивостей та винятків, що дозволяє розробникам працювати зі складними структурами даних та обробляти помилки.

Однією з цікавих особливостей є можливість додавання коментарів у форматі XML, що сприяє покращенню документації коду і забезпечує зручний доступ до інформації про класи, методи та їх параметри.

Що стосується відмінностей від попередніх мов, таких як C++ та Haskell, C# відмовляється від деяких концепцій, які можуть призвести до складнощів у розробці програмних систем. Наприклад, вона не підтримує множинного спадкування класів, що може спростити структуру програм, або виведення типів, що використовується в мовах, таких як Haskell, для автоматичного визначення типів даних. Такі обмеження можуть забезпечити

більш простий та зрозумілий шлях при створенні програм.

Ця мова програмування створена компанією Microsoft і є ключовою для розробки додатків для платформи .NET. Однією з її переваг є велика екосистема інструментів, бібліотек і фреймворків, які дозволяють розробникам створювати різноманітні програмні продукти.

C# також має вбудовану підтримку асинхронного програмування, що дозволяє розробникам легко створювати програми, які працюють ефективно та швидко обробляють паралельні завдання, такі як мережеві запити або обробка великих обсягів даних.

Однією з ключових особливостей C# є його інтеграція з платформою .NET. Це дозволяє розробникам використовувати різноманітні бібліотеки і фреймворки .NET для швидкої та ефективної розробки програм, а також сприяє переносу програм між різними платформами.

Важливо зазначити, що C# є однією з популярних мов програмування у сфері розробки ігор. Вона використовується для створення різноманітних ігрових додатків і має велику спільноту розробників, яка активно ділиться знаннями та досвідом.

Узагальнюючи, C# — це мова програмування з великим набором функцій, що робить її привабливим вибором для різних видів програмної розробки, від корпоративних програм до ігор, завдяки своїй потужності, ефективності та простоті використання.

Visual Studio є інтегрованим середовищем розробки (IDE) від Microsoft, призначеним для створення програм на різних мовах програмування, включаючи C#, C++, F#, Visual Basic та інші. Це програмне середовище надає розробникам широкий спектр інструментів для створення різноманітних програмних продуктів.

Однією з ключових переваг Visual Studio є його різнобічність та багатофункціональність. Воно підтримує різні мови програмування та типи програм, включаючи веб-додатки, мобільні застосунки, десктопні програми, хмарні рішення та ігри. Крім того, Visual Studio дозволяє розробляти

програми для різних платформ, таких як Windows, Android, iOS та інші.

Середовище має велику кількість інструментів, серед яких редактор коду, відлагодження, система контролю версій, створення і відлагодження інтерфейсів користувача та інші. Крім цього, воно підтримує роботу в команді, сприяючи спільній роботі над проектами, обміну кодом та співпраці розробників.

Visual Studio також відкрите для розширень та плагінів, що дає можливість розширювати його функціональність та налаштовувати під потреби конкретного проекту чи розробника.

Це інтегроване середовище розробки є потужним інструментом для розробників будь-якого рівня та спрощує процес створення програм, надаючи доступ до широкого спектру функцій, інструментів та можливостей для швидкої та ефективної розробки програмних продуктів.

Visual Studio, розроблене Microsoft, є потужним інтегрованим середовищем розробки (IDE), що надає розробникам різні інструменти для створення програм на різних мовах програмування. Його основні переваги полягають у різноманітності функцій та гнучкості.

Інструментарій Visual Studio охоплює широкий спектр можливостей, включаючи редактор коду з різними функціями автодоповнення, відлагодження та підтримку версій контролю за кодом. Воно підтримує створення різноманітних типів програм - від веб-додатків до мобільних застосунків і ігор. Багатофункціональність Visual Studio дозволяє розробникам працювати над проектами для різних платформ, у тому числі Windows, Android, iOS та хмарних середовищ.

Це середовище сприяє спільній роботі в команді, забезпечуючи інструменти для спільного використання коду, контролю версій, обміну інформацією та співпраці. Крім цього, Visual Studio підтримує можливість розширення через плагіни та розширення, що дозволяє налаштувати інтерфейс та функціональність IDE під індивідуальні потреби.

Загалом, Visual Studio стала важливим інструментом для розробників

будь-якого рівня, надаючи велику кількість інструментів та можливостей для створення програмних продуктів на різних платформах і в різних областях розробки. Її зручний інтерфейс та широкий спектр функцій роблять її одним із лідерів серед інтегрованих середовищ розробки програмного забезпечення.

### **3.2 Розробка програмного модулю гейміфікації додатку для вивчення основ математики**

#### Клас App

Програми, які використовують платформу WPF (Windows Presentation Foundation) для створення графічного інтерфейсу, мають кілька особливостей:

Гнучкість та Дизайн.

WPF надає багатий інструментарій для розробки стильних інтерфейсів з використанням різноманітних елементів управління, графіки, анімації та адаптивності.

Розділення Логіки та Інтерфейсу.

Методологія MVVM (Model-View-ViewModel) часто використовується для розділення логіки програми від її представлення. Це сприяє покращенню тестованості, розширюваності та підтримці коду.

Асинхронність та Багатозадачність.

WPF дозволяє працювати з асинхронним кодом та багатозадачністю, що дозволяє програмі працювати ефективно й плавно навіть при великій кількості операцій.

Анімація та Ефекти.

Інструменти WPF дозволяють створювати різноманітні анімації та спеціальні ефекти, що роблять інтерфейс живим та привабливим для користувача.

Використання XAML.

XAML (eXtensible Application Markup Language) використовується



для розміщення та опису інтерфейсу користувача. Це розділення логіки та представлення спрощує роботу з дизайном і кодом.

Взаємодія з .NET Framework.

WPF побудована на базі .NET Framework, що дозволяє використовувати широкий спектр бібліотек та інструментів для розробки функціоналу програми.

Ресурси та Стили.

Використання ресурсів та стилів дозволяє однаково формувати та налаштовувати різні елементи інтерфейсу, що спрощує управління зовнішнім виглядом програми.

Windows Presentation Foundation (WPF) - це технологія для створення графічних інтерфейсів користувача в програмах для операційних систем Windows. Вона надає розробникам широкий набір інструментів для створення стильних, інтерактивних та ефективних інтерфейсів. WPF базується на .NET Framework і використовує XAML (eXtensible Application Markup Language) для створення розмітки інтерфейсу.

Основні переваги WPF полягають у його гнучкості та можливостях налаштування, широких можливостях графіки та візуалізації, анімації та візуальних ефектів. Він також підтримує розділення логіки і представлення за допомогою патерна MVVM (Model-View-ViewModel), що спрощує тестування та підтримку коду. Крім того, WPF дозволяє працювати з різноманітними типами мультимедіа, включаючи зображення, відео та аудіо.

Завдяки XAML, розробники можуть описувати вигляд інтерфейсу у вигляді розмітки, що спрощує роботу з дизайном і розділяє логіку від представлення. Це дозволяє швидко створювати та редагувати інтерфейси, зберігаючи гнучкість та легкість в підтримці програм.

Узагальнюючи, WPF є потужним інструментом для розробки сучасних програм з графічним інтерфейсом, що надає розробникам багатий функціонал для створення стильних та функціональних додатків для операційних систем Windows.

XAML (eXtensible Application Markup Language) - це декларативна мова розмітки, що використовується в технології WPF (Windows Presentation Foundation) для описування інтерфейсів користувача. Вона дозволяє розробникам створювати візуальні елементи та їх структуру безпосередньо у вигляді текстових файлів.

Основні особливості XAML:

- Декларативний Синтаксис. XAML використовує декларативний підхід до описування інтерфейсу. Це означає, що ви описуєте, як має виглядати ваш інтерфейс, а не яким чином його створити за допомогою коду.
- Розділення Відображення та Логіки. Використання XAML дозволяє відокремлювати дизайн інтерфейсу від логіки програми. Це полегшує спільну роботу розробників і дизайнерів, а також робить код більш організованим та зрозумілим.
- Універсальність. XAML використовується не тільки в WPF, а й в інших технологіях, таких як UWP (Universal Windows Platform), Xamarin, Silverlight та інші. Це робить його більш універсальним та переносимим між різними платформами.
- Підтримка Відображення Графічних Елементів. XAML дозволяє описувати різні графічні елементи, такі як кнопки, текстові поля, графіки, анімації, стилі та інші компоненти інтерфейсу.
- Легкість Редагування та Візуалізація. Файли XAML можна легко редагувати у текстовому редакторі, що дозволяє зручно налаштовувати вигляд інтерфейсу. Крім того, інструменти розробки WPF надають зручний візуальний редактор для роботи з XAML.
- Ресурси та Шаблони. XAML дозволяє використовувати ресурси та шаблони, що полегшує стандартизацію та повторне використання елементів інтерфейсу. MVVM (Model-View-ViewModel) - це шаблон проектування програмного забезпечення, що використовується в технології WPF (Windows Presentation Foundation) для розділення логіки програми від представлення і дозволяє покращити тестованість, розширюваність та

підтримку коду.

- **Модель (Model).** Модель представляє собою логіку додатку, яка містить бізнес-логіку, обробку даних та управління станом додатку. Вона незалежна від інтерфейсу користувача і взаємодіє з джерелами даних.
- **Представлення (View).** Представлення відображає дані моделі користувачу та забезпечує можливість взаємодії з додатком через інтерфейс. У WPF це може бути вікно, сторінка або інший елемент інтерфейсу.
- **ViewModel.** ViewModel є посередником між моделлю та представленням. Вона приймає дані з моделі та підготовлює їх для відображення у представленні. ViewModel також обробляє команди від користувача та виконує необхідні дії в моделі.

Основні переваги MVVM:

- **Розділення Відповідальностей.** MVVM дозволяє чітко розділити логіку додатку, його представлення та взаємодію з користувачем, що спрощує розробку та підтримку коду.
- **Тестованість.** Оскільки логіка знаходиться в окремих компонентах (модель та ViewModel), це полегшує написання та виконання автоматизованих тестів.
- **Підтримка Дизайну.** ViewModel можна пов'язати з дизайнерськими інструментами для візуального представлення та перевірки макету без необхідності функціонування всього додатку.
- **Реактивність Інтерфейсу Користувача.** Зміни в ViewModel автоматично оновлюють відображення в представленні завдяки механізму прив'язок даних (Data Binding) у WPF.

MVVM дозволяє покращити структуру коду, полегшити співпрацю між розробниками та дизайнерами, а також забезпечує більш гнучке та тестоване програмне забезпечення.

.NET Framework - це велика платформа розробки програмного забезпечення, яка надає розробникам зручний інструментарій для створення різноманітних додатків. Вона підтримує різні мови програмування, такі як

C#, VB.NET, F#, що дозволяє програмістам вибирати мову, яка найкраще відповідає їхнім потребам. .NET Framework надає доступ до великої кількості бібліотек та фреймворків, які полегшують роботу з різними аспектами програмування, такими як робота з базами даних, графікою, мережами та іншими.

Ця платформа є кросплатформеною, що означає, що програми, розроблені на .NET Framework, можуть працювати на різних операційних системах, таких як Windows, Linux та macOS. Її основним елементом є CLR (Common Language Runtime), яке забезпечує виконання програм та управління ресурсами під час виконання. Крім того, .NET Framework має інтегровані середовища розробки, такі як Visual Studio, які надають зручність у розробці та налагодженні програм.

Велике співтовариство розробників .NET та різноманітність ресурсів для навчання роблять .NET Framework популярним інструментом для розробки високоякісного програмного забезпечення.

### **3.3 Тестування розробленого програмного модуля гейміфікації додатку для вивчення основ математики**

При запуску гри з'являється початкове вікно додатку (рис. 3.2). Обираємо «Linear Systems» (рис.3.3) На наступному рисунку можна побачити вікно, де генерується кількість прикладів (рис.3.4). Розв'язання прикладів гравцем, обробка виключень при введенні некоректних даних (рис.3.5). Можна побачити виведення кількості правильних відповідей на екран (рис.3.6).

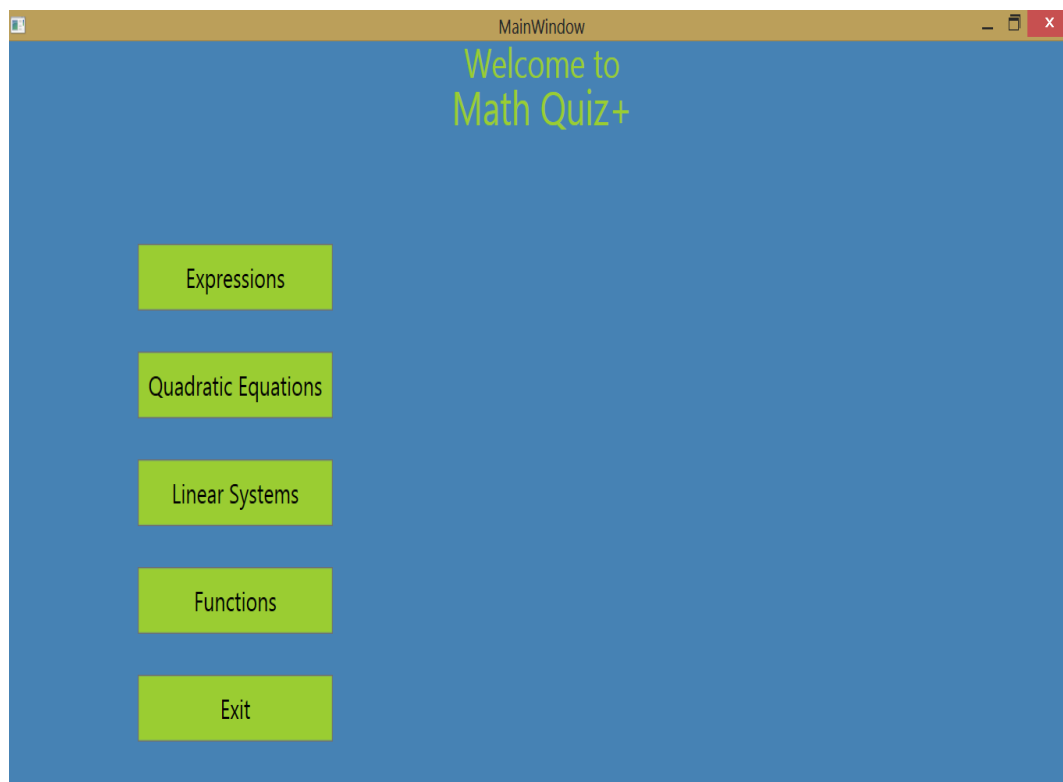


Рисунок 3.2 – Початкове вікно «Brain Strain (Math Quiz Plus)»

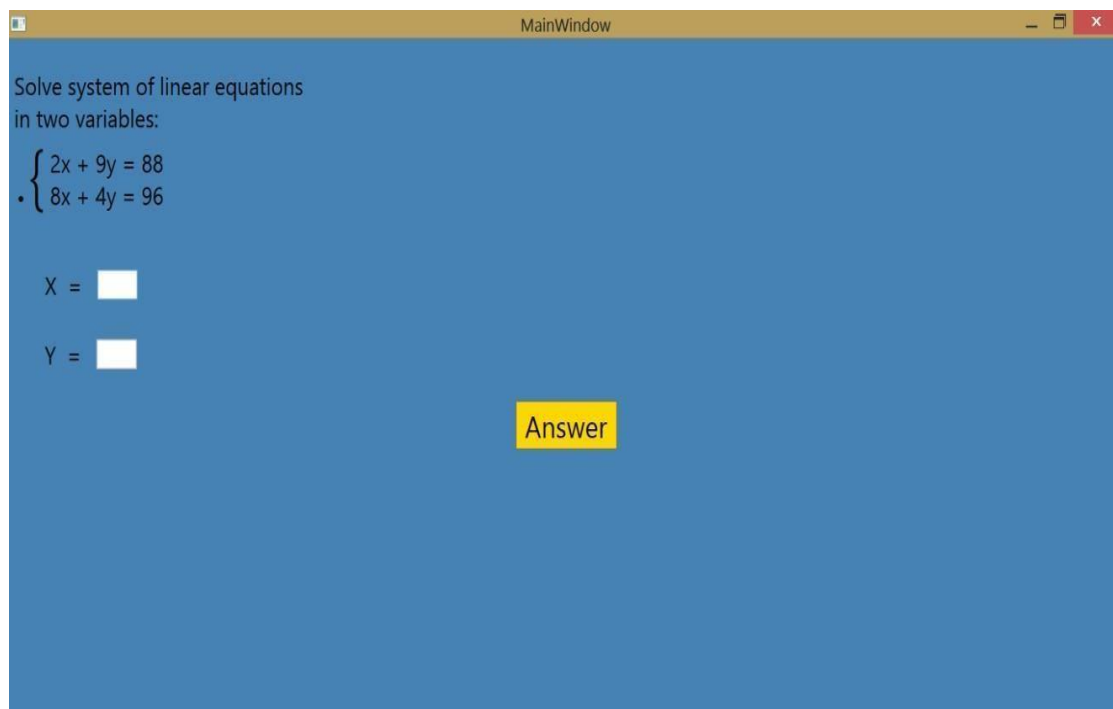


Рисунок 3.3 – Режим гри «Linear Systems»

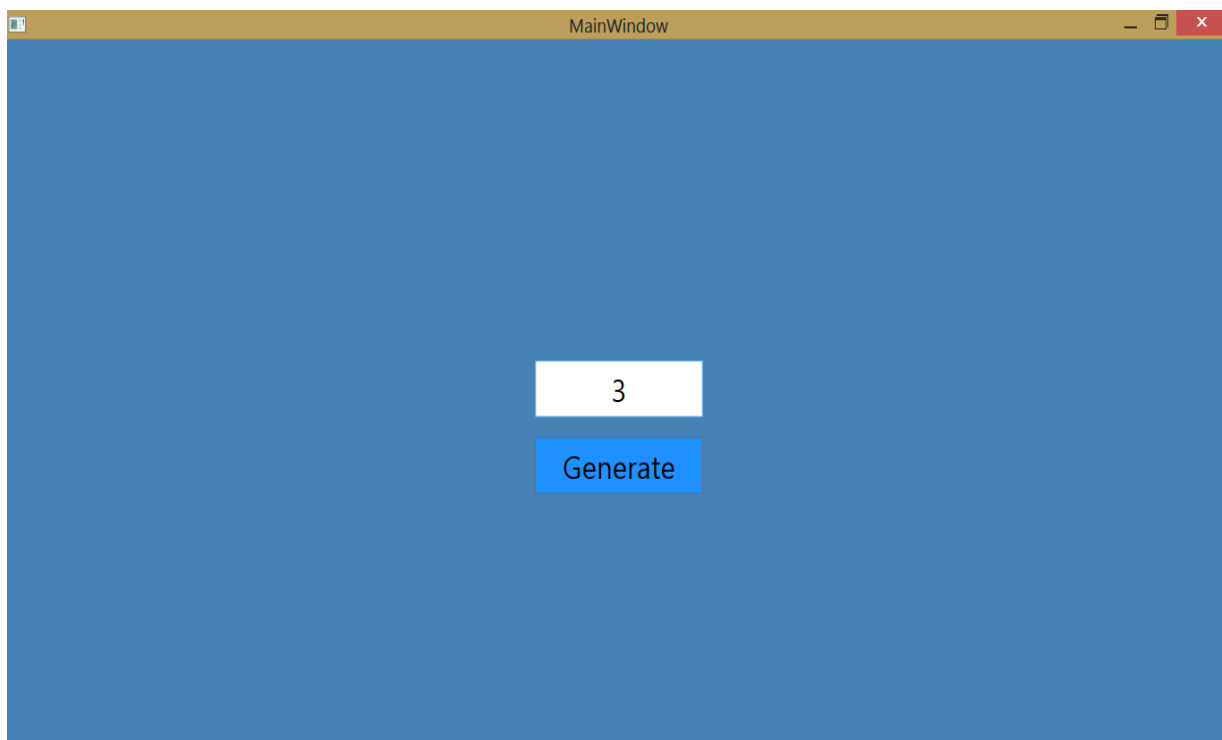


Рисунок 3.4 – Генерація кількості прикладів

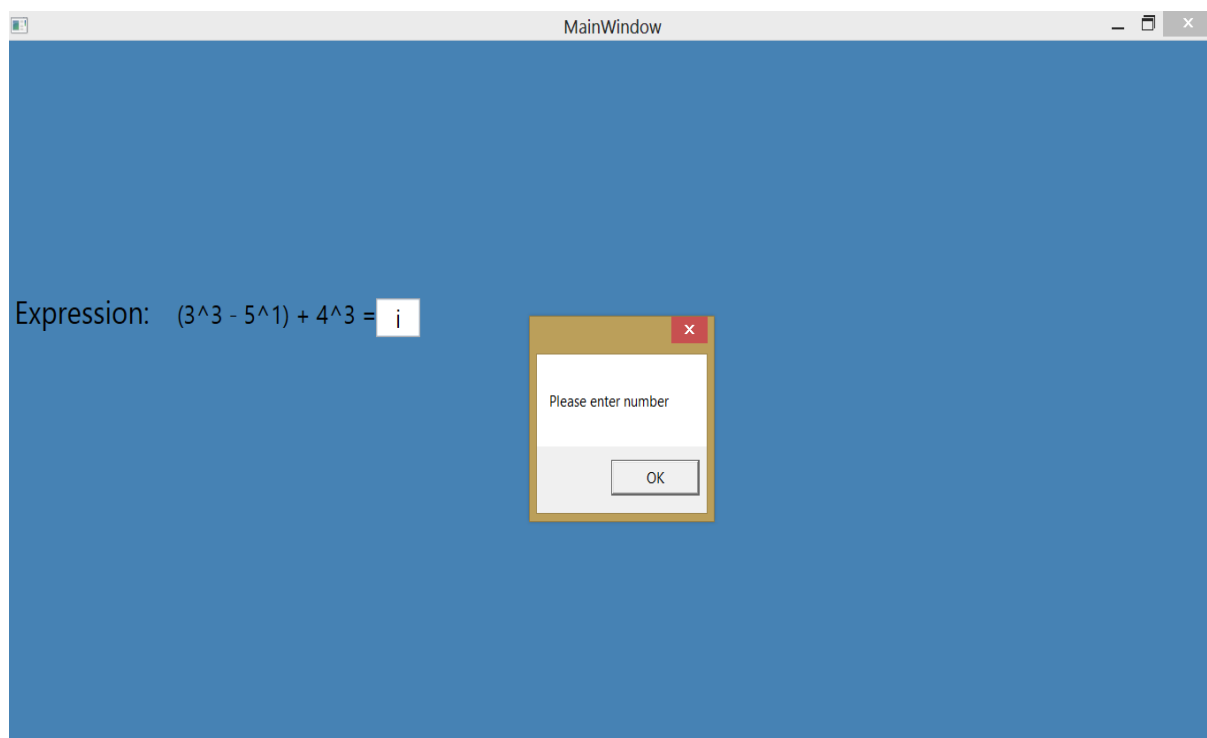


Рисунок 3.5 – Обробка виключень при некоректній відповіді

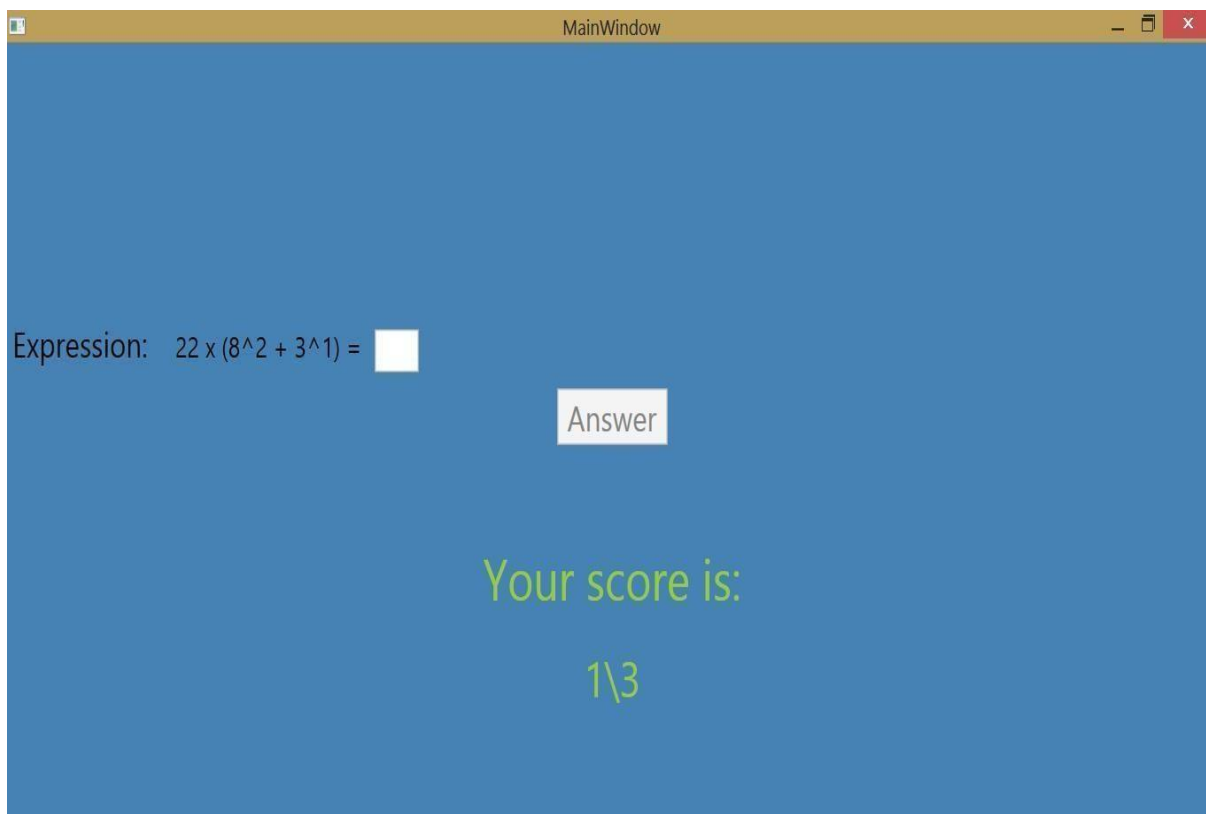


Рисунок 3.6 – Виведення кількості правильних відповідей

### **3.4 Порівняльна характеристика ефективності розробленої інформаційної технології гейміфікації додатку для вивчення основ математики**

Для візуалізації порівняльної характеристики розробленої системи гейміфікації порівняно з аналогами було проведено колективну експертизу з залученням десяти експертів (студентів) та оцінюванням роботи з чотирма подібними системами, а саме – даною системою, яка на відміну від існуючих, є інтуїтивно зрозумілою для користувачів та дозволяє пройти тестування без реєстрації, а також трьома її аналогами.

Було обрано метод прямої колективної експертизи для оцінки роботи додатків та її результати оформлено у таблиці 3.1 та на рис.3.7.

Таблиця оцінює різні параметри систем та їх аналогів. Розроблена система має високі показники у всіх категоріях, переважно перевершуючи аналоги. Її перевага може бути визначена через середні оцінки, що вищі, ніж у конкурентів,

на відповідних показниках.

Порахуємо ефективність для кожної категорії та загальну ефективність.

$$\text{Зручність} = \frac{4,8}{(4,77+4,5+4,7)/3} - 1 * 100\% = 2,56\%$$

$$\text{Відсутність багів} = \frac{4,76}{(4,78+4,7+4,71)/3} - 1 * 100\% = 0,64\%$$

$$\text{Графічний інтерфейс} = \frac{4,76}{(4,8+4,83+4,56)/3} - 1 * 100\% = 2,96\%$$

Для вирахування загальної ефективності

$$\frac{2.56+0.64+2.96}{3} = 2.05\%.$$

Отже, розроблена система ефективніша на 2.05% з аналогів.

Таблиця 3.1 – Результати колективної оцінки систем гейміфікації

	Середня оцінка за зручністю роботи	Середня оцінка за відсутністю багів	Середня оцінка за графічним інтерфейсом користувача
Розроблена система	4,8	4,76	4,87
Аналог 1	4,77	4,78	4,8
Аналог 2	4,5	4,7	4,83
Аналог 3	4,79	4,71	4,56

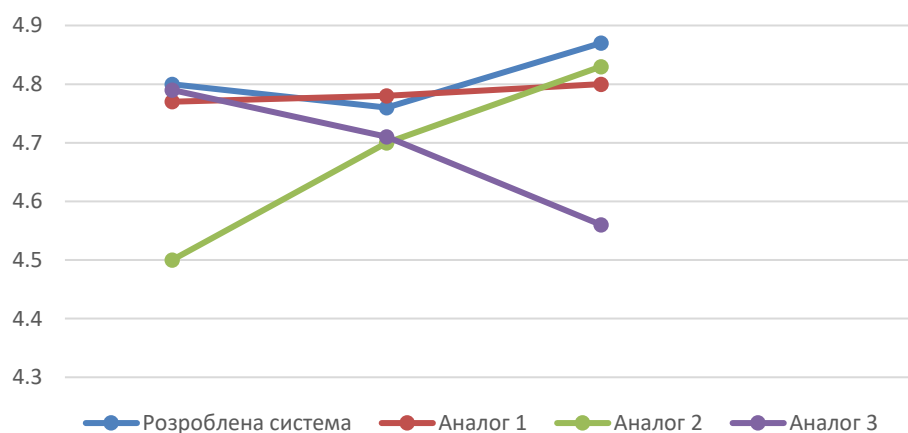


Рисунок 3.7 – Порівняльна характеристика систем гейміфікації



### 3.5 Висновок до розділу 3

Здійснено вибір мови програмування, після проведення порівняльного аналізу трьох мов об'єктно-орієнтованого програмування, що всі відповідають вимогам до виконання роботи. Обрано оптимальне середовище розробки системи, після детального дослідження всіх його можливостей та переваг, що залучені чи прямо використовуються у створенні необхідного додатку. Описано усі базові принципи розробки гейміфікованого додатку, створено прототип, що відповідає поставленим вимогам, протестовано його можливості та оцінено результати, отримані дані збігаються з вимогами до проекту, чим доведено його коректність. Доведено, що розроблена система ефективніша на 2.05% за аналоги.

## 4 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота «Інформаційна технологія гейміфікації додатку для вивчення основ математики» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

- 1) проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;
- 2) розраховано витрати на здійснення науково-технічної розробки;
- 3) розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

### **4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки**

Метою проведення комерційного і технологічного аудиту дослідження за темою «Інформаційна технологія гейміфікації додатку для вивчення основ математики» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в таблиці 4.1 [19].

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в	Технічні та споживчі властивості продукту значно кращі, ніж в
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає

Продовження таблиці 4.1

Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці 4.2.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	5	5	4
2. Ринкові переваги (наявність аналогів)	3	3	3
3. Ринкові переваги (ціна продукту)	4	4	3
4. Ринкові переваги (технічні властивості)	3	3	4
5. Ринкові переваги (експлуатаційні витрати)	2	2	3
6. Ринкові перспективи (розмір ринку)	3	3	3
7. Ринкові перспективи (конкуренція)	2	2	2
8. Практична здійсненність (наявність фахівців)	5	5	5
9. Практична здійсненність (наявність фінансів)	2	3	2
10. Практична здійсненність (необхідність нових матеріалів)	4	5	5
11. Практична здійсненність (термін реалізації)	3	4	5
12. Практична здійсненність (розробка документів)	4	5	4
Сума балів	40	44	43
Середньоарифметична сума балів $СБ_c$	42,3		

За результатами розрахунків, наведених в таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в таблиці 4.3 [19].

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів $СБ$ розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41...48	Високий
31...40	Вище середнього
21...30	Середній
11...20	Нижче середнього
0...10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія гейміфікації додатку для вивчення основ математики» становить 42,3 бала, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

## 4.2 Розрахунок узагальненого коефіцієнта якості розробки

Окрім комерційного аудиту розробки доцільно також розглянути технічний рівень якості розробки, розглянувши її основні технічні показники. Ці показники по-різному впливають на загальну якість проектної розробки.

Узагальнений коефіцієнт якості ( $B_n$ ) для нового технічного рішення розраховуємо за формулою [20]:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i, \quad (4.1)$$

де  $k$  – кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

$\alpha_i$  – коефіцієнт, який враховує питому вагу  $i$ -го технічного показника в загальній якості розробки. Коефіцієнт  $\alpha_i$  визначається експертним шляхом

і при цьому має виконуватись умова  $\sum_{i=1}^k \alpha_i = 1$ ;

$\beta_i$  – відносне значення  $i$ -го технічного показника якості нової розробки.

Відносні значення  $\beta_i$  для різних випадків розраховуємо за такими формулами:

- для показників, зростання яких вказує на підвищення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ni}}{I_{ai}}, \quad (4.2)$$

де  $I_{ni}$  та  $I_{na}$  – чисельні значення конкретного  $i$ -го технічного показника якості відповідно для нової розробки та аналога;

- для показників, зростання яких вказує на погіршення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ai}}{I_{ni}}; \quad (4.3)$$

Використовуючи наведені залежності можемо проаналізувати та порівняти техніко-економічні характеристики аналогу та розробки на основі отриманих наявних та проектних показників, а результати порівняння зведемо до таблиці 4.4.

Таблиця 4.4 – Порівняння основних параметрів розробки та аналога.

Показники (параметри)	Одиниця вимірювання	Аналог	Проектований пристрій	Відношення параметрів нової розробки до аналога	Питома вага показника
Доступність (дружність) інтерфейсу	бал	5	9	1,8	0,3
Розмір на диску	Мб	50	2	25	0,1
Потреба в оперативній пам'яті	Гб	8	4	2	0,25
Швидкодія взаємозв'язку	с	0,9	0,5	1,8	0,1
Кількість підтримуваних баз даних	од	1	3	3	0,25

Узагальнений коефіцієнт якості ( $B_n$ ) для нового технічного рішення складе:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i = 1,8 \cdot 0,3 + 25 \cdot 0,1 + 2 \cdot 0,25 + 1,8 \cdot 0,1 + 3 \cdot 0,25 = 4,47.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 4,47 рази.

### 4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Інформаційна технологія гейміфікації додатку для вивчення основ математики», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Витрати на основну заробітну плату дослідників ( $Z_o$ ) розраховуємо у відповідності до посадових окладів працівників, за формулою [19]:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.4)$$

де  $k$  – кількість посад дослідників залучених до процесу досліджень;

$M_{ni}$  – місячний посадовий оклад конкретного дослідника, грн;

$t_i$  – число днів роботи конкретного дослідника, дн.;

$T_p$  – середнє число робочих днів в місяці,  $T_p=22$  дні.

$$Z_o = 15000,00 \cdot 22 / 22 = 15000,00 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 4.5.

Таблиця 4.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	15000,00	681,82	22	15000,00
Інженер-дослідник (програміст)	17000,00	772,73	22	17000,00



Консультант математики категорії)	(вчитель вищої категорії)	12000,00	545,45	5	2727,27
Технік 1-ї категорії		7800,00	354,55	22	7800,00
Всього					42527,27

Витрати на основну заробітну плату робітників ( $Z_p$ ) за відповідними найменуваннями робіт НДР на тему «Інформаційна технологія гейміфікації додатку для вивчення основ математики» розраховуємо за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.5)$$

де  $C_i$  – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

$t_i$  – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду  $C_i$  можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.6)$$

де  $M_M$  – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), прийmemo  $M_M=6700,00$  грн;

$K_i$  – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [19];

$K_c$  – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

$T_p$  – середнє число робочих днів в місяці, приблизно  $T_p = 22$  дн;

$t_{зм}$  – тривалість зміни, год.

$$C_1 = 6700,00 \cdot 1,10 \cdot 1,35 / (22 \cdot 8) = 56,53 \text{ грн.}$$

$$З_{р1} = 56,53 \cdot 3,50 = 197,86 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 4.6.

Таблиця 4.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Інсталяція програмного забезпечення розробки програмного забезпечення	3,50	2	1,10	56,53	197,86
Встановлення цифрових обчислювальних систем та мережевого доступу	2,50	3	1,35	69,38	173,45
Відлагодження програмних модулів аналітичного дослідження	2,00	5	1,70	87,37	174,73
Всього					546,04

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод} = (З_o + З_p) \cdot \frac{H_{дод}}{100\%}, \quad (4.7)$$

де  $H_{дод}$  – норма нарахування додаткової заробітної плати. Прийmemo 10%.

$$З_{дод} = (42527,27 + 546,04) \cdot 10 / 100\% = 4307,33 \text{ грн.}$$

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_n = (З_o + З_p + З_{дод}) \cdot \frac{H_{zn}}{100\%} \quad (4.8)$$

де  $H_{zn}$  – норма нарахування на заробітну плату. Приймаємо 22%.

$$Z_n = (42527,27 + 546,04 + 4307,33) \cdot 22 / 100\% = 10423,74 \text{ грн.}$$

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Інформаційна технологія гейміфікації додатку для вивчення основ математики».

Витрати на матеріали ( $M$ ), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{\sigma j}, \quad (4.9)$$

де  $H_j$  – норма витрат матеріалу  $j$ -го найменування, кг;

$n$  – кількість видів матеріалів;

$C_j$  – вартість матеріалу  $j$ -го найменування, грн/кг;

$K_j$  – коефіцієнт транспортних витрат, ( $K_j = 1,1 \dots 1,15$ );

$B_j$  – маса відходів  $j$ -го найменування, кг;

$$M_1 = 3,0 \cdot 225,00 \cdot 1,1 - 0 \cdot 0 = 742,50 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 4.7.

Таблиця 4.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Папір канцелярський офісний (A4) XEROX ULTRA PLUS	225,00	3,0	0	0	742,50
Папір для заміток (A5) Light	152,00	3,0	0	0	501,60

Начиння канцелярське Premium	256,00	3,0	0	0	844,80
Органайзер офісний	210,00	3,0	0	0	693,00
Картридж для принтера	2086,00	2,0	0	0	4589,20
Диск оптичний	25,00	5,0	0	0	137,50
USB-пам'ять Microtech 32 GB	135,00	2,0	0	0	297,00
Тека для паперів	101,00	3,0	0	0	333,30
Всього					8138,90

Витрати на комплектуючі ( $K_e$ ), які використовують при проведенні НДР на тему «Інформаційна технологія гейміфікації додатку для вивчення основ математики», розраховуємо, згідно з їхньою номенклатурою, за формулою:

$$K_e = \sum_{j=1}^n H_j \cdot C_j \cdot K_j \quad (4.10)$$

де  $H_j$  – кількість комплектуючих  $j$ -го виду, шт.;

$C_j$  – покупна ціна комплектуючих  $j$ -го виду, грн;

$K_j$  – коефіцієнт транспортних витрат, ( $K_j = 1,1 \dots 1,15$ ).

$$K_e = 1 \cdot 3680,00 \cdot 1,05 = 3864,00 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 4.8.

Таблиця 4.8 – Витрати на комплектуючі

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн	Сума, грн
Маршрутизатор VX2300-PH	1	3680,00	3864,00
Всього			3864,00

До статті «Специстаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання специстаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість специстаткування розраховуємо за формулою:

$$B_{спеу} = \sum_{i=1}^k C_i \cdot C_{np.i} \cdot K_i , \quad (4.11)$$

де  $C_i$  – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{np.i}$  – кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

$K_i$  – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ( $K_i = 1, 10 \dots 1, 12$ );

$k$  – кількість найменувань устаткування.

$$B_{спеу} = 7650,00 \cdot 1 \cdot 1,05 = 8032,50 \text{ грн.}$$

Отримані результати зведемо до таблиці 4.9.

Таблиця 4.9 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Мережеве обладнання передачі цифрових даних	1	7650,00	8032,50
Всього			8032,50

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{прог} = \sum_{i=1}^k C_{инрг} \cdot C_{npг.i} \cdot K_i , \quad (4.12)$$

де  $C_{инрг}$  – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{\text{прг.}i}$  – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

$K_i$  – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ( $K_i = 1, 10 \dots 1, 12$ );

$k$  – кількість найменувань програмних засобів.

$$B_{\text{прг}} = 5999,00 \cdot 1 \cdot 1,05 = 6298,95 \text{ грн.}$$

Отримані результати зведемо до таблиці 4.10.

Таблиця 4.10 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Середовище програмування Visual Studio Community	1	5999,00	6298,95
Всього			6298,95

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{\text{обл}} = \frac{Ц_{\text{б}}}{T_{\text{е}}} \cdot \frac{t_{\text{вик}}}{12}, \quad (4.13)$$

де  $Ц_{\text{б}}$  – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{\text{вик}}$  – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_{\text{е}}$  – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{\text{обл}} = (32599,00 \cdot 1) / (3 \cdot 12) = 905,53 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 4.11.

Витрати на силову електроенергію ( $B_e$ ) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{eni}}{\eta_i}, \quad (4.14)$$

де  $W_{yi}$  – встановлена потужність обладнання на визначеному етапі розробки, кВт;

$t_i$  – тривалість роботи обладнання на етапі дослідження, год;

$C_e$  – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo  $C_e = 7,50$  грн;

Таблиця 4.11 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Персональний комп'ютер HP ZBook Firefly 14 G7 Mobile Workstation	32599,00	3	1	905,53
Робоче місце інженера-розробника (дослідника програмного забезпечення)	6899,00	5	1	114,98
Графічні пристрої виводу інформації	7770,00	4	1	161,88
Офісна оргтехніка	8050,00	4	1	167,71
Приміщення лабораторії розробки та дослідження	320000,00	25	1	1066,67
ОС Windows 11	5630,00	3	1	156,39
Прикладний пакет Microsoft	5220,00	3	1	145,00

Office 2021 Professional Plus				
Всього				2718,15

$K_{eni}$  – коефіцієнт, що враховує використання потужності,  $K_{eni} < 1$ ;

$\eta_i$  – коефіцієнт корисної дії обладнання,  $\eta_i < 1$ .

$B_e = 0,06 \cdot 160,0 \cdot 7,50 \cdot 0,95 / 0,97 = 72,00$  грн.

Проведені розрахунки зведемо до таблиці 4.12.

Таблиця 4.12 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Персональний комп'ютер HP ZBook Firefly 14 G7 Mobile Workstation	0,06	160,0	72,00
Робоче місце інженера-розробника (дослідника програмного забезпечення)	0,08	160,0	96,00
Графічні пристрої виводу інформації	0,12	2,0	1,80
Офісна оргтехніка	0,32	2,0	4,80
Всього			174,60

Витрати за статтею «Службові відрядження» відсутні.

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» відсутні.

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:



$$I_e = (Z_o + Z_p) \cdot \frac{H_{ie}}{100\%}, \quad (4.15)$$

де  $H_{ie}$  – норма нарахування за статтею «Інші витрати», прийmemo  $H_{ie} = 55\%$ .

$$I_e = (42527,27 + 546,04) \cdot 55 / 100\% = 23690,32 \text{ грн.}$$

#### 4.3.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (4.16)$$

де  $H_{нзв}$  – норма нарахування за статтею «Накладні (загальновиробничі) витрати», прийmemo  $H_{нзв} = 100\%$ .

$$B_{нзв} = (42527,27 + 546,04) \cdot 100 / 100\% = 43073,31 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему «Інформаційна технологія гейміфікації додатку для вивчення основ математики» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = Z_o + Z_p + Z_{одд} + Z_n + M + K_v + B_{спец} + B_{прг} + A_{обл} + B_e + B_{св} + B_{сп} + I_v + B_{нзв}. \quad (4.17)$$

$$B_{заг} = 42527,27 + 546,04 + 4307,33 + 10423,74 + 8138,90 + 3864,00 + 8032,50 + 6298,95 + 2718,15 + 174,60 + 0,00 + 0,00 + 23690,32 + 43073,31 = 153795,12 \text{ грн.}$$

Загальні витрати  $ZB$  на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ЗВ = \frac{B_{заг}}{\eta}, \quad (4.18)$$

де  $\eta$  - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo  $\eta=0,95$ .

$$ЗВ = 153795,12 / 0,95 = 161889,60 \text{ грн.}$$

#### **4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором**

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою «Інформаційна технологія гейміфікації додатку для вивчення основ математики» передбачають комерціалізацію протягом 4-х років реалізації на ринку. Робота ідентифукується як *«Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем»*.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$\Delta N$  – збільшення кількості споживачів продукту, у періоди часу, що аналізуються, від покращення його певних характеристик;

Подібні показники продемонстровано в таблиці 4.13.

Таблиця 4.13 – Збільшення кількості споживачів у співвідношенні до показників

Показник	1-й рік	2-й рік	3-й рік	4-й рік
----------	---------	---------	---------	---------

Збільшення кількості споживачів, осіб	1000	1500	1600	960
---------------------------------------	------	------	------	-----

$N$  – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, прийmemo 10000 осіб;

$C_o$  – вартість програмного продукту у році до впровадження результатів розробки, прийmemo 320,00 грн;

$\pm\Delta C_o$  – зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, прийmemo 60,40 грн.

Можливе збільшення чистого прибутку у потенційного інвестора  $\Delta\Pi_i$  для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [20]:

$$\Delta\Pi_i = (\pm\Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\mathcal{G}}{100}\right), \quad (4.19)$$

де  $\lambda$  – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість складає 20%, а коефіцієнт  $\lambda = 0,8333$ ;

$\rho$  – коефіцієнт, який враховує рентабельність інноваційного продукту).  
Прийmemo  $\rho = 43\%$ ;

$\mathcal{G}$  – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році  $\mathcal{G} = 18\%$ ;

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (60,40 \cdot 10000,00 + 380,40 \cdot 1000) \cdot 0,83 \cdot 0,43 \cdot (1 - 0,18/100\%) = 288092,54 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (60,40 \cdot 10000,00 + 380,40 \cdot 2500) \cdot 0,83 \cdot 0,43 \cdot (1 - 0,18/100\%) = 455083,19 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (60,40 \cdot 10000,00 + 380,40 \cdot 4100) \cdot 0,83 \cdot 0,43 \cdot (1 - 0,18/100\%) = 633206,56 \text{ грн.}$$

Збільшення чистого прибутку 4-го року:

$$\Delta\Pi_4 = (60,40 \cdot 10000,00 + 380,40 \cdot 5060) \cdot 0,83 \cdot 0,43 \cdot (1 - 0,18/100\%) = 740080,57 \text{ грн.}$$

Приведена вартість збільшення всіх чистих прибутків  $ПП$ , що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (4.20)$$

де  $\Delta\Pi_i$  – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

$T$  – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

$\tau$  – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні,  $\tau = 0,22$ ;

$t$  – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$ПП = 288092,54/(1+0,22)^1 + 455083,19/(1+0,22)^2 + 633206,56/(1+0,22)^3 + 740080,57/(1+0,22)^4 = 236141,42 + 305753,29 + 348711,21 + 334071,70 = 1224677,61$$

грн.

Величина початкових інвестицій  $PV$ , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{инв} \cdot 3B, \quad (4.21)$$

де  $k_{инв}$  – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо  $k_{инв} = 2$ ;

$3B$  – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 161889,60 грн.

$$PV = k_{инв} \cdot 3B = 2 \cdot 161889,60 = 323779,20 \text{ грн.}$$

Абсолютний економічний ефект  $E_{абс}$  для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV \quad (4.22)$$

де  $ПП$  – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 1224677,61 грн;

$PV$  – теперішня вартість початкових інвестицій, 323779,20 грн.

$$E_{абс} = ПП - PV = 1224677,61 - 323779,20 = 900898,41 \text{ грн.}$$

Внутрішня економічна дохідність інвестицій  $E_e$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_e = T_{жс} \sqrt[4]{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.23)$$

де  $E_{абс}$  – абсолютний економічний ефект вкладених інвестицій, 900898,41 грн;

$PV$  – теперішня вартість початкових інвестицій, 323779,20 грн;

$T_{жс}$  – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, 4 роки.

$$E_e = T_{жс} \sqrt[4]{1 + \frac{E_{абс}}{PV}} - 1 = (1 + 900898,41/323779,20)^{1/4} - 1 = 0,39.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій  $\tau_{min}$ :

$$\tau_{\text{мін}} = d + f, \quad (4.24)$$

де  $d$  – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні  $d = 0,1$ ;

$f$  – показник, що характеризує ризикованість вкладення інвестицій, прийmemo 0,25.

$\tau_{\text{мін}} = 0,1 + 0,25 = 0,35 < 0,39$  свідчить про те, що внутрішня економічна дохідність інвестицій  $E_g$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Інформаційна технологія гейміфікації додатку для вивчення основ математики» доцільно.

Період окупності інвестицій  $T_{\text{ок}}$  які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{\text{ок}} = \frac{1}{E_g}, \quad (4.25)$$

де  $E_g$  – внутрішня економічна дохідність вкладених інвестицій.

$$T_{\text{ок}} = 1 / 0,39 = 2,53 \text{ р.}$$

$T_{\text{ок}} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

#### 4.5 Висновок до розділу 4

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія гейміфікації додатку для вивчення основ математики» становить 42,3 бала, що, свідчить про комерційну важливість

проведення даних досліджень (рівень комерційного потенціалу розробки високий).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 4,47 рази.

Також термін окупності становить 2,53 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Інформаційна технологія гейміфікації додатку для вивчення основ математики».

## ВИСНОВКИ

В даній магістерській роботі було проведено детальний аналіз предметної області гейміфікації. В результаті цього аналізу було встановлено, що гейміфікація є підходом, що застосовує елементи гри в неігровому контексті з метою стимулювання активності та залучення користувачів до різних видів діяльності. Також був проведений аналітичний огляд відомих реалізацій систем гейміфікації, що дозволило ознайомитись з основними принципами та методиками, які використовуються у таких системах. Цей огляд був корисним для розуміння різноманітності підходів та можливостей, що існують у гейміфікації.

В роботі було розроблено алгоритм роботи модуля гейміфікації додатку для вивчення основ математики. Також було проведено проектування UML-діаграми додатку, що сприяло кращому розумінню структури та функціональності модулю, що розробляється. Проведений аналіз демонструє доцільність застосування гейміфікації в додатках навчального типу, зокрема для вивчення основ математики. Гейміфікація може зробити навчання більш захоплюючим та мотивуючим для користувачів, сприяючи активному залученню до процесу навчання та досягненню кращих результатів. Було обґрунтовано вибір середовища і мови програмування, що сприяло успішній реалізації розроблюваного програмного модуля гейміфікації. Також було проведено тестування розробленого модуля та порівняльна характеристика його ефективності. Результати тестування підтвердили правильність реалізації модуля та показали, що його ефективність на 2.05% більша у порівнянні з аналогами.

Отже, всі задачі магістерського дослідження виконано, мету роботи досягнуто.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Чверкун-Капуш М.О., Петришин С.І. Огляд гейміфікованих застосунків для вивчення математики в Матеріали конференції «ЛІІ Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2023)», Вінниця, 2023. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2023/paper/view/18843/15585>
2. Детердінг С., Діксон Д., Халед Р. та Наке Л. (2011). Від елементів ігрового дизайну до ігровості: визначення гейміфікації. У матеріалах 15-ї Міжнародної академічної конференції MindTrek: Уявлення про медіасередовище майбутнього (стор. 9-15). ACM.
3. Zichermann, G., & Cunningham, C. (2011). Гейміфікація за проектом: впровадження ігрових механізмів у веб- та мобільних додатках. "O'Reilly Media, Inc."
4. Капп, К. М. (2012). Гейміфікація навчання та викладання: ігрові методи та стратегії для навчання та освіти. Джон Вайлі та сини.
5. Вербак К. та Хантер Д. (2012). Задля перемоги: як ігрове мислення може революціонізувати ваш бізнес. Wharton Digital Press.
6. Huotari, K., & Hamari, J. (2012). Визначення гейміфікації: перспектива маркетингу послуг. У матеріалах 16-ї Міжнародної академічної конференції MindTrek (стор. 17-22). ACM.
7. Дічева, Д., Дічев, К., Агре, Г., і Ангелова, Г. (2015). Гейміфікація в освіті: систематичне картографічне дослідження. *Journal of Educational Technology & Society*, 18(3), 75-88.
8. Хамарі, Дж., Койвісто, Дж., і Сарса, Х. (2014). Чи працює гейміфікація? — Огляд літератури емпіричних досліджень гейміфікації. У матеріалах 47-ї Гавайської міжнародної конференції з системних наук (стор. 3025-3034). IEEE.
9. Ніколсон, С. (2012). Орієнтована на користувача теоретична основа для змістовної гейміфікації. У *Games+ Learning+ Society Conference* (том 8,

стор. 223- 230).

10. Ландерс Р. Н., Бауер К. Н., Каллан Р. К. та Армстронг М. Б. (2017). Психологічна теорія та гейміфікація: перспектива теорії самовизначення. Симуляція та ігри, 48 (4), 507-535.

11. Morschheuser, V., Hamari, J., & Koivisto, J. (2016). Гейміфікація в краудсорсингу: огляд. У матеріалах 49-ї Гавайської міжнародної конференції з системних наук (стор. 2368-2377). IEEE.

12. Csikszentmihalyi, M. (1990). Потік: психологія оптимального досвіду. Harper & Row.

13. Деці, Е. Л. та Райан, Р. М. (1985). Внутрішня мотивація та самодетермінація в поведінці людини. Springer Science & Business Media.

14. Чоу, Ю. К. (2015). Дійсна гейміфікація: крім балів, значків і таблиць лідерів. Octalysis Media.

15. Джонсон, С. (2005). Все погане добре для вас: як сучасна популярна культура насправді робить нас розумнішими. Ріверхед Букс.

16. Рігбі, С., і Райан, Р. М. (2011). Як гейміфікація може мотивувати учнів до навчання? У дослідженні та практиці технологічного вдосконаленого навчання (с. 1-13). Шпрінгер Берлін Гейдельберг.

17. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.

18. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа – Вінниця : ВНТУ, 2016. – 113 с.

19. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.

20. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа – Вінниця : ВНТУ, 2016. – 113 с.

## ДОДАТКИ

**Додаток А (обов'язковий) Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень**

**ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ  
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ**

Назва роботи: Інформаційна технологія гейміфікації додатку для вивчення основ математики

Тип роботи: магістерська кваліфікаційна робота  
(БДР, МКР)

Підрозділ кафедра комп'ютерних наук, ФІПА  
(кафедра, факультет)

**Показники звіту подібності Unicheck**

Оригінальність 85,6% Схожість 14,4%

**Аналіз звіту подібності (відмітити потрібне):**

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи  Чверкун-Капуш М.О.

Керівник роботи \_\_\_\_\_ Петришин С.І.

**Опис прийнятого рішення**

Магістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку  Озеранський В.С.

## Додаток Б (обов'язковий) Лістинг програми

```

using System;
using System.Threading;

namespace WpfApplication2
{
    class Expression
    {
        private int a;
        private int b;
        private int c;
        private int payman1; private
        int payman2; private int
        payman3; private double
        x1; private double x2;
        private double x3; private
        double p1; private double
        p2; private double p3;
        private string a1 = "";
        private string a2 = "";
        private string b1 = "";
        private string b2 = "";
        private string operation1;
        private string operation2;
        private int result;
        private string expr = "";
        public void Generate()
        {
            Random rand = new Random();

            //Generiruyem Skobki #region
            int tempBr = rand.Next(0, 2); if
            (tempBr == 0)
                a1 = "";
            else a1 = "(";
            if (a1 == "(")
                a2 = ")";
            else a2 = "";

```

```

if (a1 == "(")
{
    b1 = "";
    b2 = "";
}
else
{
    tempBr = rand.Next(0, 2); if
(tempBr == 0)
    b1 = "";
else b1 = "(";
    if (b1 == "(")
        b2 = ")";
    else b2 = "";
}
#endregion

//Generiruyem chisla #region
//Pervoe chislo
payman1 = rand.Next(0, 2); if
(payman1 == 0)
    a = rand.Next(0, 50);
else //Esli est' stepen'
{
    double k1;
    x1 = rand.Next(1, 10); p1
= rand.Next(1, 4); k1 =
Math.Pow(x1, p1); a =
(int)k1;
}

//Vtoroe chislo
payman2 = rand.Next(0, 2); if
(payman2 == 0)
    b = rand.Next(1, 50);

else //Esli est' stepen'
{

```

```

double k2;
x2 = rand.Next(2, 10); p2
= rand.Next(1, 4); k2 =
Math.Pow(x2, p2); b =
(int)k2;
}
//Tret'e chislo
payman3 = rand.Next(0, 2); if
(payman3 == 0)
    c = rand.Next(1, 50);

else                //Esli est' stepen'
{
    double k3;
    x3 = rand.Next(2, 10); p3
    = rand.Next(1, 4); k3 =
    Math.Pow(x3, p3); c =
    (int)k3;
}
#endregion

//Generiruyem operacii
#region
int temp1 = rand.Next(0, 4); // '+' -> 0, '-' -> 1, 'x' -> 2, '/' -> 3 if (temp1 != 3)
{
    switch (temp1)
    {
        case 0:
            operation1 = "+";
            break;

        case 1:
            operation1 = "-"; break;

        case 2:
            operation1 = "x"; break;

    }
}

else                //Proverka delenia

```

```

{
    operation1 = "/"; if
    ((a % b) == 0)
    {
        operation1 = "/";
    }
    else
    {
        Thread.Sleep(10);
        Generate();
    }
}

//Generiruem vtoruyu operaciyu
int temp2 = rand.Next(0, 3); // '+' -> 0, '-' -> 1, 'x' -> 2 switch (temp2)
{
    case 0:
        operation2 = "+"; break;

    case 1:
        operation2 = "-";
        break;

    case 2:
        operation2 = "x"; break;
}
#endregion

//Generiruyem resultat dlya proverki otveta
if (a1 == "" && b1 == "") //Sluchay bez skobok
#region
{
    if (operation1 == "+" && operation2 == "+") result = a + b +
        c;
    else if (operation1 == "+" && operation2 == "-") result = a - b;
    else if (operation1 == "+" && operation2 == "x") result = a +
        b * c;
    else if (operation1 == "-" && operation2 == "+") result = a - b

```



```

    + c;
else if (operation1 == "-" && operation2 == "-") result = a - b
    - c;
else if (operation1 == "-" && operation2 == "x") result = a - b
    * c;
else if (operation1 == "x" && operation2 == "+") result = a *
    b + c;
else if (operation1 == "x" && operation2 == "-") result = a * b - c;
else if (operation1 == "x" && operation2 == "x")

    result = a * b * c;
else if (operation1 == "/" && operation2 == "+") result = a / b
    + c;
else if (operation1 == "/" && operation2 == "-") result = a / b - c;
else if (operation1 == "/" && operation2 == "x") result = a / b
    * c;
}
#endregion

else if (a1 == "(" && b1 == "") //Sluchay (a+b)*c
#region
{
    if (operation1 == "+" && operation2 == "+") result = (a + b) +
        c;
    else if (operation1 == "+" && operation2 == "-") result = (a + b) -
        c;
    else if (operation1 == "+" && operation2 == "x") result = (a +
        b) * c;
    else if (operation1 == "-" && operation2 == "+") result = (a -
        b) + c;
    else if (operation1 == "-" && operation2 == "-") result = (a -
        b) - c;
    else if (operation1 == "-" && operation2 == "x") result = (a -
        b) * c;
    else if (operation1 == "x" && operation2 == "+") result = (a *
        b) + c;
    else if (operation1 == "x" && operation2 == "-") result = (a * b) -
        c;
    else if (operation1 == "x" && operation2 == "x")

        result = (a * b) * c;
}

```

```

else if (operation1 == "/" && operation2 == "+") result = (a /
    b) + c;
else if (operation1 == "/" && operation2 == "-") result = (a / b) -
    c;
else if (operation1 == "/" && operation2 == "x") result = (a /
    b) * c;
}
#endregion

else if (a1 == "" && b1 == "(") //Sluchay a*(b+c)
#region
{
    if (operation1 == "+" && operation2 == "+") result = a + (b +
        c);
    else if (operation1 == "+" && operation2 == "-") result = a + (b -
        c);
    else if (operation1 == "+" && operation2 == "x") result = a +
        (b * c);
    else if (operation1 == "-" && operation2 == "+") result = a -
        (b + c);
    else if (operation1 == "-" && operation2 == "-") result = a - (b
        - c);
    else if (operation1 == "-" && operation2 == "x") result = a -
        (b * c);
    else if (operation1 == "x" && operation2 == "+") result = a *
        (b + c);
    else if (operation1 == "x" && operation2 == "-") result = a * (b -
        c);
    else if (operation1 == "x" && operation2 == "x")
        result = a * (b * c);
    else if (operation1 == "/" && operation2 == "+")
    {
        if (a % (b + c) == 0)
            result = a / (b + c);
        else a = a - a % (b + c);
        result = a / (b + c);
    }
    else if (operation1 == "/" && operation2 == "-")
    {
        if (a % (b - c) == 0)

```

```

        result = a / (b - c);
    else a = a - a % (b - c);
    result = a / (b - c);
}
else if (operation1 == "/" && operation2 == "x")
{
    if (a % (b * c) == 0)
        result = a / (b * c);
    else a = a - a % (b * c);
    result = a / (b * c);
}
}
}
#endregion
}

//Proverka otveta
public bool isCorrect(int result)
{
    return this.result == result;}

//Pokazivaem novoe virajenie public
string Show()
{
    for (int i = 0; i < 1; i++)
    {
        expr = "";
        string s = " ";
        expr = expr + a1; if
        (payman1 == 0)
            expr += Convert.ToString(a) + s + operation1 + s + b1; else
            expr += Convert.ToString(x1) + "^" + Convert.ToString(p1) + s + operation1 + s +
b1;

        if (payman2 == 0)
            expr += Convert.ToString(b) + a2 + s + operation2 + s; else
            expr += Convert.ToString(x2) + "^" + Convert.ToString(p2) + a2 +
s + operation2 + s;

        if (payman3 == 0)
            expr += Convert.ToString(c) + b2 + s + "=" + " "; else
            expr += Convert.ToString(x3) + "^" + Convert.ToString(p3) + b2 +

```

```
s + "=" + " ";
```

```
    }
    //Console.Write(expr); return
    expr; // CHANGED
  }
}
}
using System;
```

```
namespace WpfApplication2
{
    public class Functions
    {
        public string answer;

        public string Generate()
        {
            String[] functions = { "f(x) = x", "f(x) = 1/x", "f(x) = x^3", "f(x) = -x^3",
"f(x) = sinx", "f(x) = tgx", "f(x) = ctgx", "f(x) = arcsinx", "f(x) = arctgx",
        "f(x) = x^2", "f(x) = x^2 + x", "f(x) = |x|", "f(x) = cosx", "f(x) = -x^2",
        "f(x) = x^3 + 1", "f(x) = (x+3)^2", "f(x) = 12x + 1", "f(x) = Sqrt(x)"};
            Random rand = new Random();
            int index = rand.Next(0, functions.Length);
            //Console.WriteLine("Is          this          functon          Odd          or          Even:
                {0}", functions[index]);

            if (index >= 0 && index <= 8) answer =
                "Odd";
            else if (index >= 9 && index <= 14) answer
                = "Even";
            else
                answer = "Neither Odd nor Even";

            return functions[index];
        }
    }
}
```



```
case 2:
    x = 100 * rand.Next(1, 10);
    y = 100 * rand.Next(1, 10); break;
}

//Generating a, b, c, d
a = rand.Next(2, 10);

b = rand.Next(2, 10); if
(b == a)
{
    while (b == a)
    {
        Thread.Sleep(10);
        b = rand.Next(2, 10);
    }
}
c = rand.Next(1, 10);
d = rand.Next(1, 10); if
(d == c)
{
    while (d == c)
    {
        Thread.Sleep(10);
        d = rand.Next(1, 10);
    }
}
```

## Додаток В (обов'язковий)

## ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ГЕЙМИФІКАЦІЇ ДОДАТКУ ДЛЯ  
ВИВЧЕННЯ ОСНОВ МАТЕМАТИКИВиконав: студент 2-го курсу, групи ЗКН-22мспеціальності 122 «Комп'ютерні науки»

(шифр і назва напрямку підготовки, спеціальності)

Чверкун-Капуш М.О.

(прізвище та ініціали)

Керівник: ж.т.н., ст. викл. кафедри КНПетришин С.І.

(прізвище та ініціали)

« 07 » 12 2023 р.

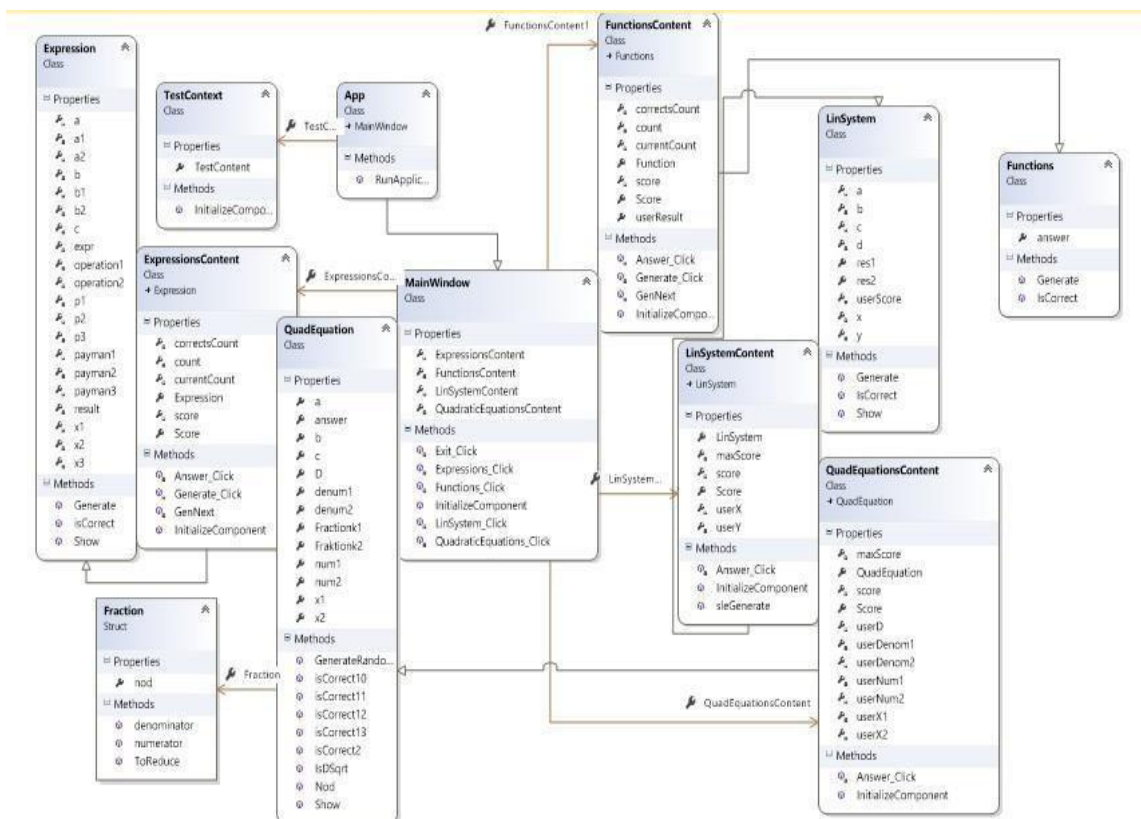


Рисунок В.1 – UML-діаграма класів додатку

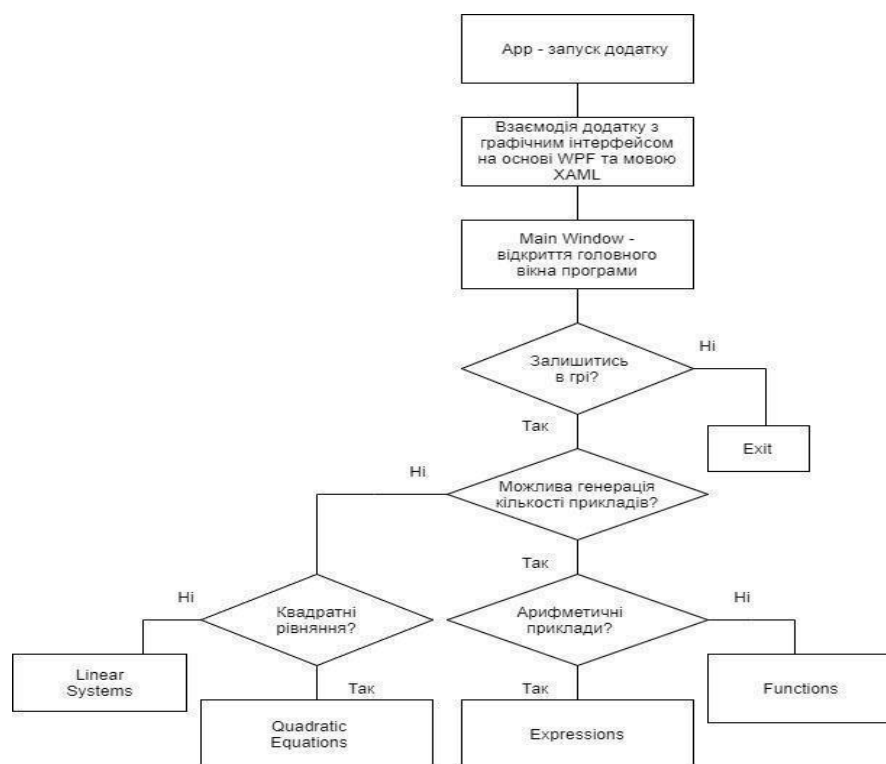


Рисунок В.2 – Алгоритм функціонування додатку





Рисунок В.3 – Алгоритм функціонування основних модулів системи



Рисунок В.4 – Початкове вікно



Рисунок В.5 – Режим гри «Linear Systems»

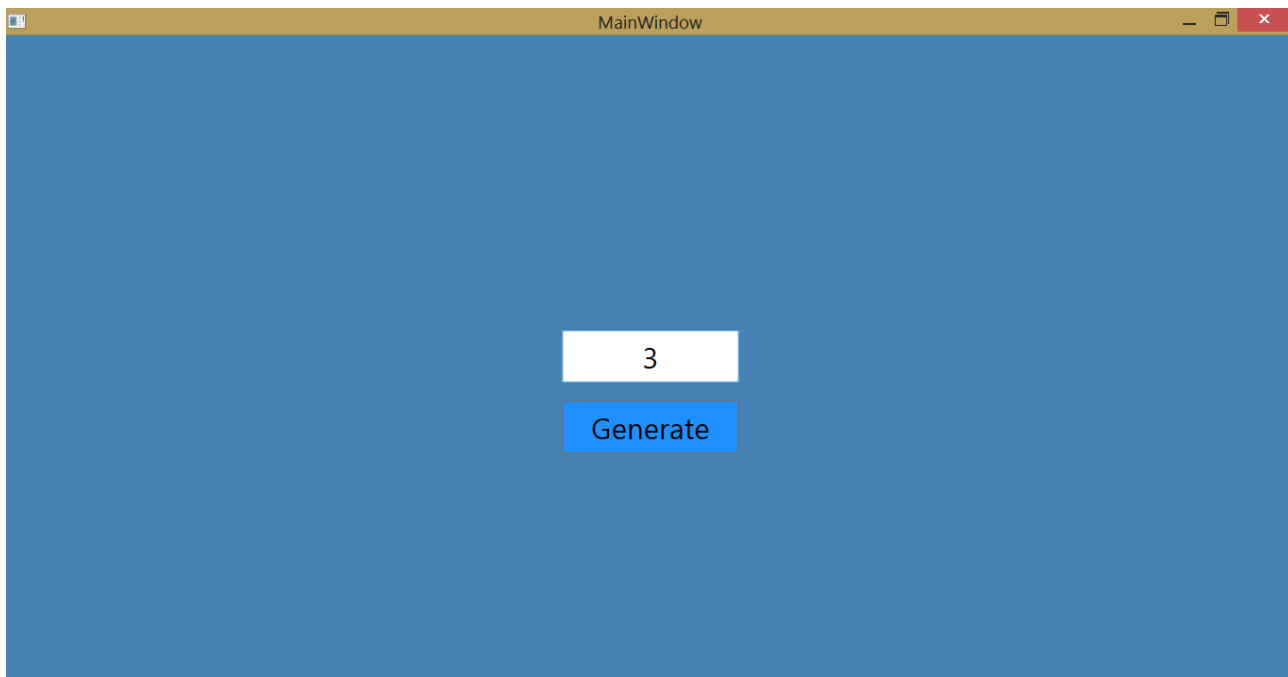


Рисунок В.6 – Генерація кількості прикладів

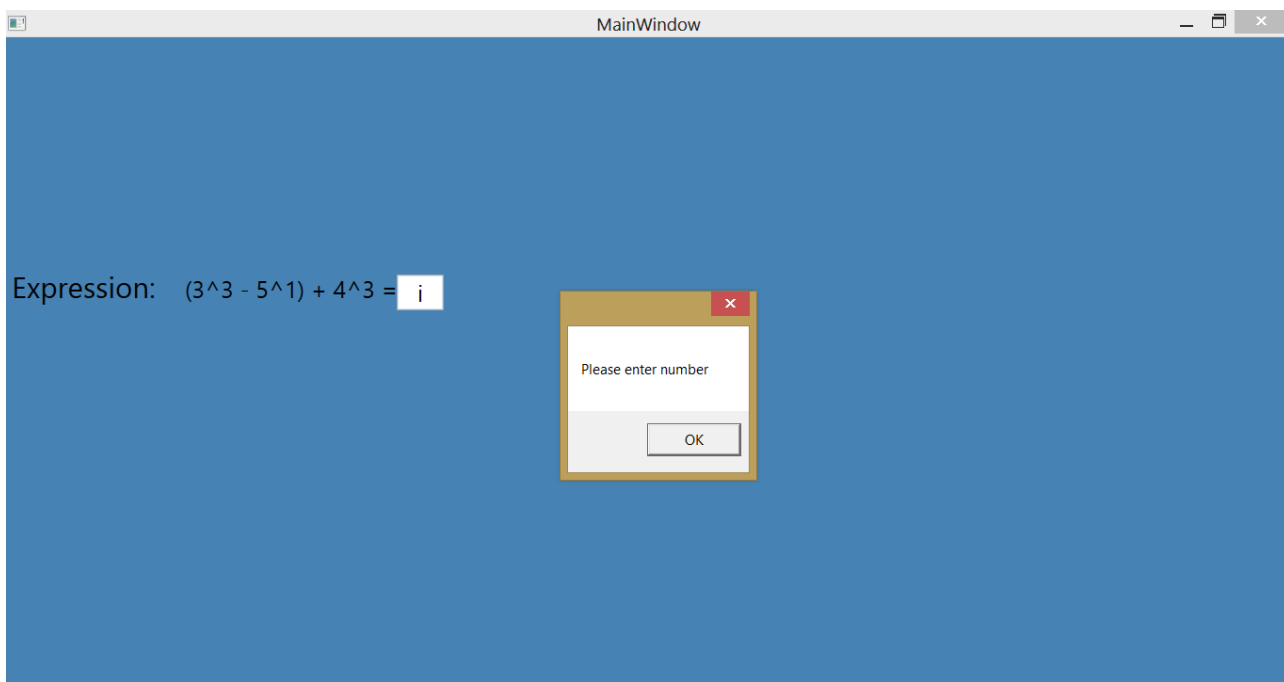


Рисунок В.7 – Обробка виключень при некоректній відповіді

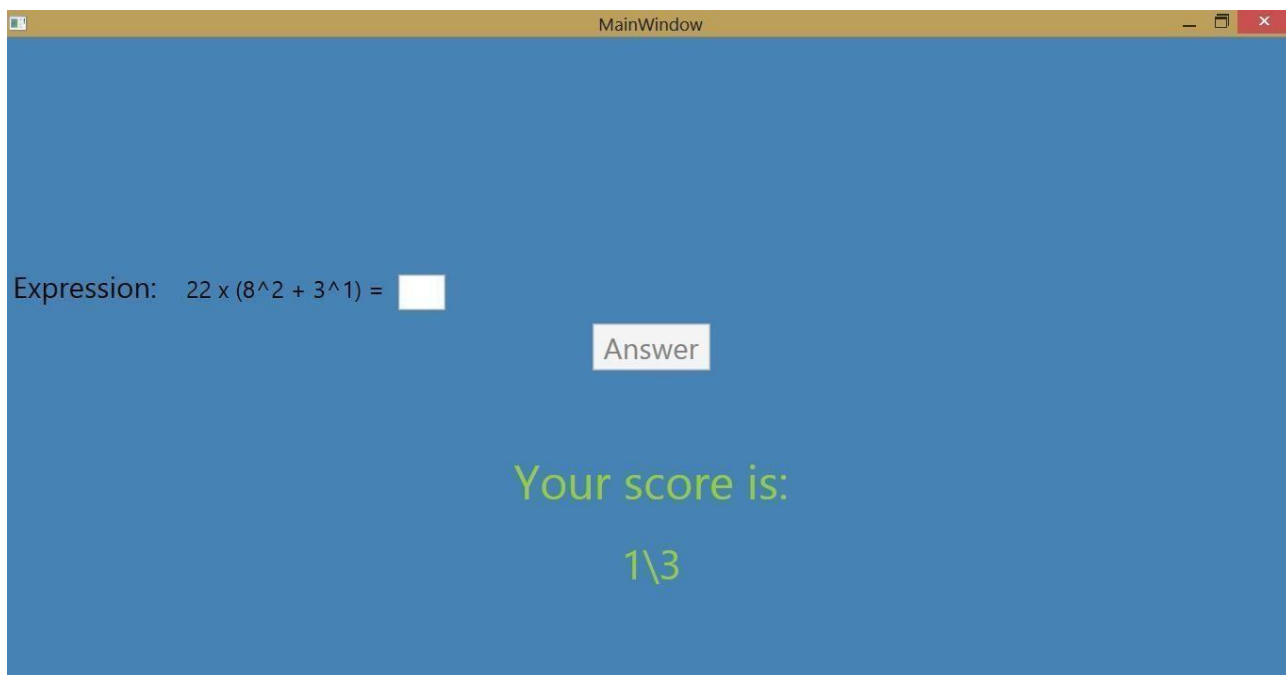


Рисунок В.8 – Виведення кількості правильних відповідей



Рисунок В.9 – Структурна схема роботи додатку

## Додаток Г (довідниковий) Інструкція користувача

Спершу запусимо файл MathQuiz.exe (рис. Г.1).

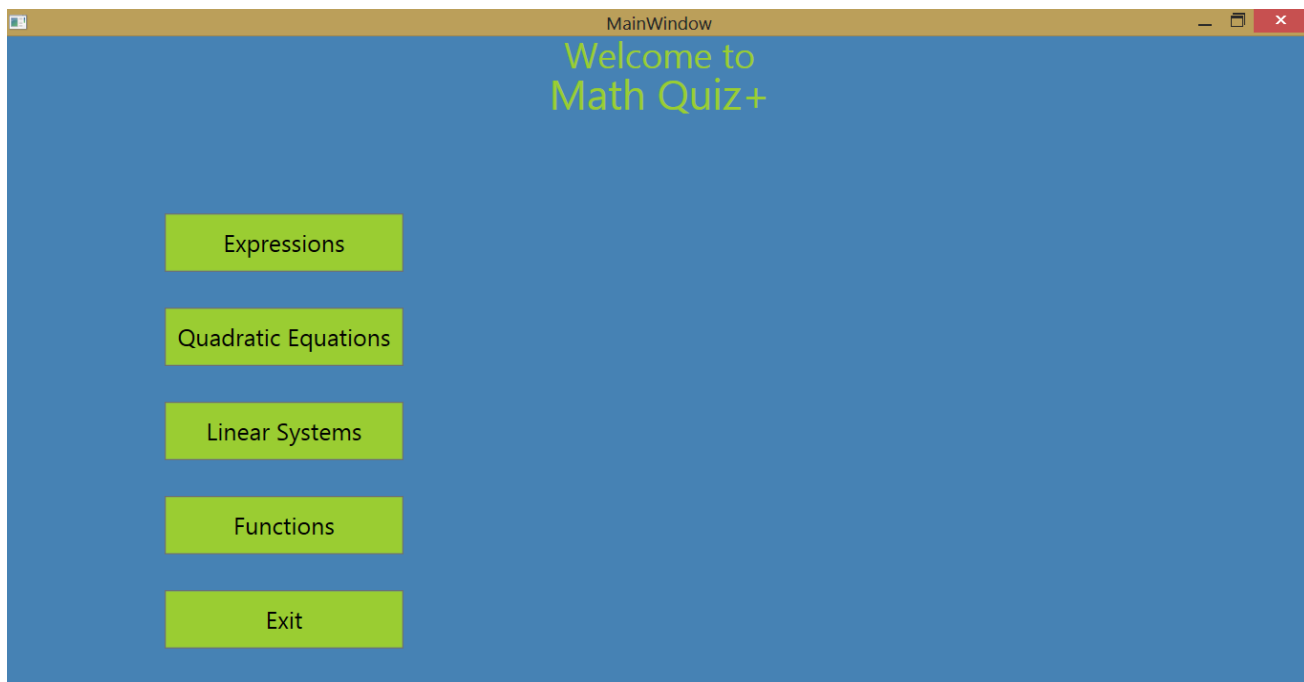


Рисунок Г.1 – Початкове вікно програми

Оберемо один з чотирьох можливих режимів гри і оберемо кількість прикладів для генерації системою (рис. Г.2).

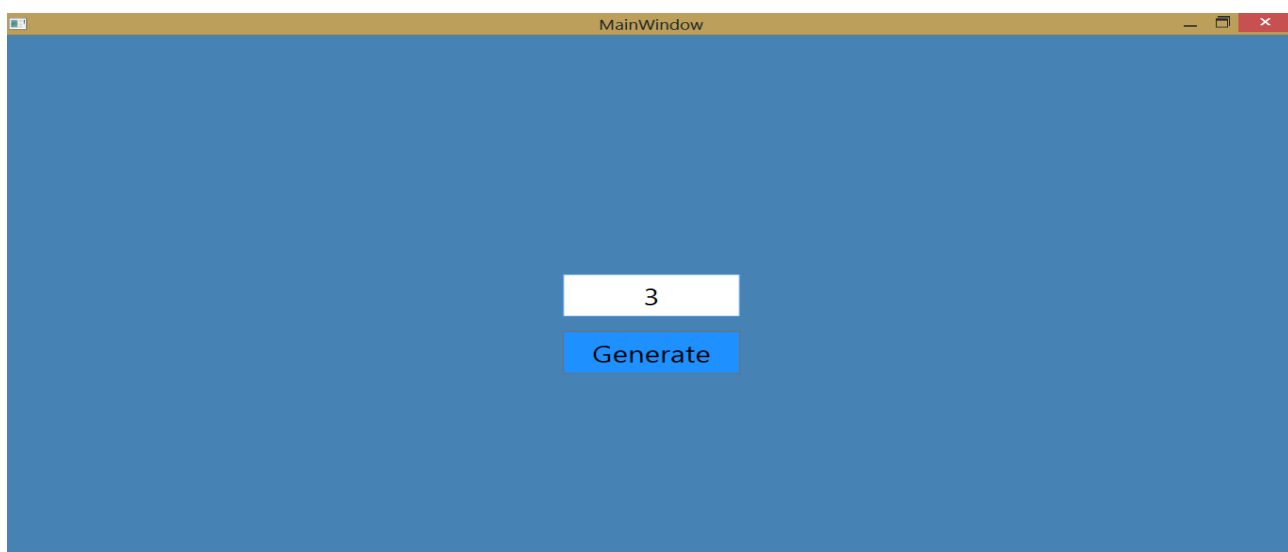


Рисунок Г.2 – Генерація потрібної кількості прикладів

Далі починаємо проходити безпосередньо математичну вікторину (рис. Г.3).

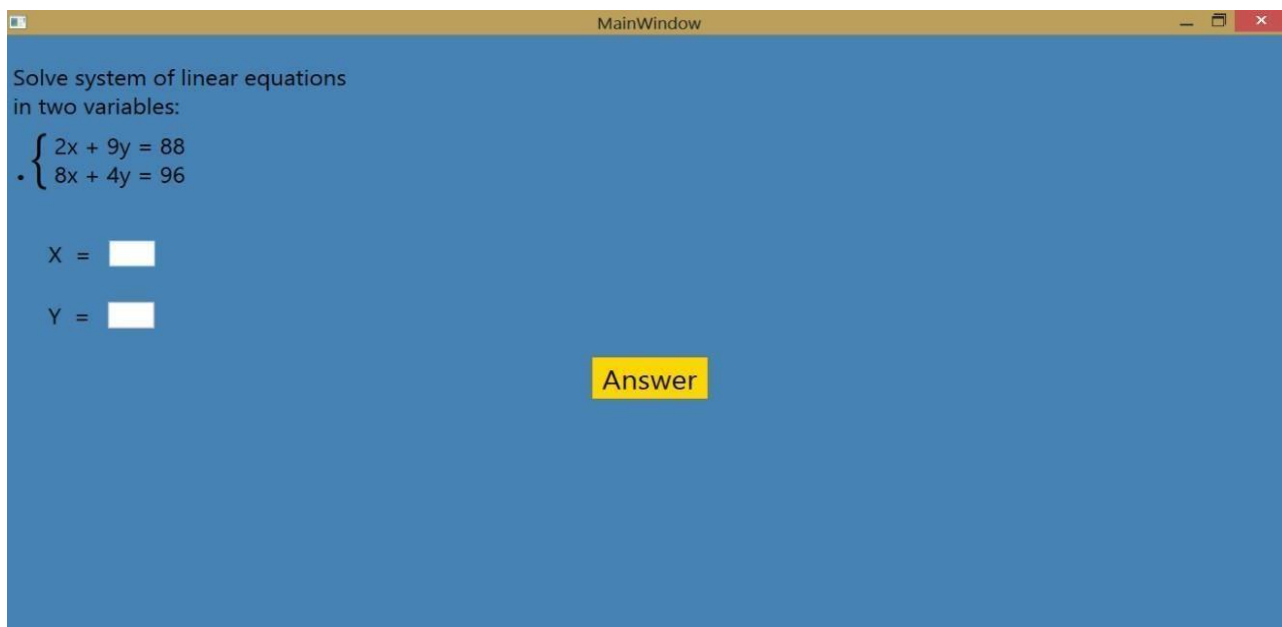


Рисунок Г.3 – Проходження вікторини

В кінці буде виведено результат проходження вікторини (рис. Г.4).

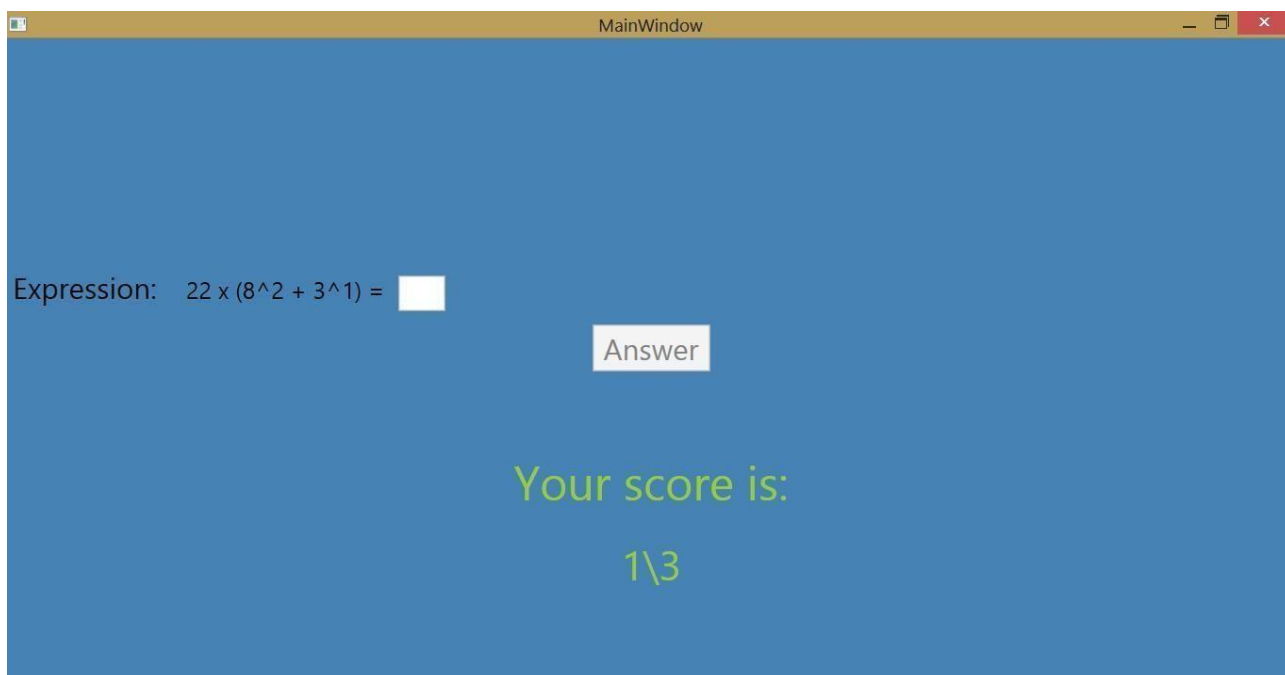


Рисунок Г.4 – Результат проходження вікторини