ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ Навчально-науковий інститут денної освіти Форма навчання денна

Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту Завідувач кафедри _____ Олена ОЛЬХОВСЬКА «___»____202_ р.

КВАЛІФІКАЦІЙНА РОБОТА НА ТЕМУ:

Розробка програмного забезпечення дидактичного інструменту з теми «Критерій максимуму середнього виграшу для кількісних шкал» дистанційного навчального курсу «Системний аналіз та теорія прийняття рішень»

Зі спеціальності 122 Комп'ютерні науки освітня програма «Комп'ютерні науки» ступеня магістра

Виконавець роботи Пилипенко Назарій Вікторович

_____ «___» ____ 202___ р.

Науковий керівник к. ф.-м. н., доцент, Черненко Оксана Олексіївна

______ «___» ____ 202___ р.

Рецензент

ЗАТВЕРДЖУЮ

Завідувач кафедри _____ Олена ОЛЬХОВСЬКА

«___» <u>січня</u> 2023 р. ЗАВДАННЯ ТА КАЛЕНДАРНИЙ ГРАФІК ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

на тему «Розробка програмного забезпечення дидактичного інструменту з теми «Критерій максимуму середнього виграшу для кількісних шкал» дистанційного навчального курсу «Системний аналіз та теорія прийняття рішень»»

зі спеціальності 122 Комп'ютерні науки

освітня програма «Комп'ютерні науки»

ступеня <u>магістра</u>

Призвіще, ім'я, по батькові Пилипенко Назарій Вікторович

Затверджена наказом ректора №8-Н від «16» січня 2023 р.

Термін подання студентом роботи «__» ____ 202_ р.

Вихідні дані до кваліфікаційної роботи роботи: <u>публікації з теми навчальні</u> тренажери в дистанційних курсах з комп'ютерних наук.

Зміст пояснювальної записки (перелік питань, які потрібно розробити)

ВСТУП

- 1. ПОСТАНОВКА ЗАДАЧІ
- 2. ІНФОРМАЦІЙНИЙ ОГЛЯД
- 2.1 Огляд схожих за тематикою тренажерів
- 2.2 Актуальність розробки тренажеру
- 3. ТЕОРЕТИЧНА ЧАСТИНА
- 3.1 Критерій максимуму середнього виграшу для кількісних шкал
- 3.2 Алгоритм роботи тренежера
- 3.3 Блок-схеми програми тренажера
- 4. ПРАКТИЧНА ЧАСТИНА
- 4.1 Опис розробки тренажера

4.2 Інструкція до роботи з тренажером ВИСНОВОК

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ДОДАТОК А Перелік графічного матеріалу: 3-4 аркуші блок-схем, інші необхідні ілюстрації.

Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище,	Підпи	с, дата
	ініціали, посада	завдання	завдання
	консультанта	видав	прийняв
Постановка задачі	Черненко О.О.		
Інформаційний огляд	Черненко О.О.		
Теоретична частина	Черненко О.О.		
Практична реалізація	Черненко О.О.		

6. Календарний графік виконання кваліфікаційної роботи

	Зміст роботи	Термін	Фактичне
		виконання	виконання
1.	Вступ		
2.	Вивчення методичних рекомендацій та стандартів та		
звіт	керівнику		
3.	Постановка задачі		
4.	Інформаційний огляд джерел бібліотек та інтернету		
5.	Теоретична частина		
6.	Практична частина		
7.	Закінчення оформлення		
8.	Доповідь студента на кафедрі		
9.	Доробка (за необхідністю), рецензування		

Дата видачі завдання «	_>>.	202_ p.
Здобувач вищої освіти		Пилипенко Назарій Вікторович

Науковий керівник	к.фм.н., доц.

Результати захисту кваліфікаційної роботи

Кваліфікаційна робота оцінена на __________(балів, оцінка за національною шкалою, оцінка за ЕСТS)

Протокол засідання ЕК № ____ від «____» ____ 202_ р.

Секретар ЕК_____

(ініціали та прізвище)

Затверджую

Зав. кафедри _____ к. ф.-м. н. Олена ОЛЬХОВСЬКА «____» _____ 202_ p.

Погоджено Науковий керівник К. ф.-м. н. _____ 202 p. «___»____

План

кваліфікайійної роботи ступеня магістра зі спеціальності 122 Комп'ютерні науки освітня програма «Комп'ютерні науки» Пилипенко Назарій Вікторович Прізвище, ім'я, по батькові

на тему «Розробка програмного забезпечення дидактичного інструменту з теми «Критерій максимуму середнього виграшу для кількісних шкал» дистанційного навчального курсу «Системний аналіз та теорія прийняття рішень»

ВСТУП

- 1. ПОСТАНОВКА ЗАДАЧІ
- 2. ІНФОРМАЦІЙНИЙ ОГЛЯД
- 2.1 Огляд схожих за тематикою тренажерів
- 2.2 Актуальність розробки тренажеру
- 3. ТЕОРЕТИЧНА ЧАСТИНА
- 3.1 Критерій максимуму середнього виграшу для кількісних шкал
- 3.2 Алгоритм роботи тренежера
- 3.3 Блок-схеми програми тренажера
- 4. ПРАКТИЧНА ЧАСТИНА
- 4.1 Опис розробки тренажера

4.2 Інструкція до роботи з тренажером

ВИСНОВОК

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ДОДАТОК А

Здо	бувач вищої о	світи		Назарій ПИЛИПЕНКО
	\ \	202	n	

«____»____202_p.

ΡΕΦΕΡΑΤ

Записка: 53 стор., в т.ч. основна частина 43 стор., джерел - 12.

Мета роботи: розробка програмного забезпечення тренажеру з теми «Критерій максимуму середнього виграшу для кількісних шкал» для дистанційного навчального курсу «Системний аналіз та теорія прийняття рішень».

Об'єкт розробки: програма-тренажер на тему «Критерій максимуму середнього виграшу для кількісних шкал».

Предмет розробки: програмне забезпечення тренажеру з теми «Критерій максимуму середнього виграшу для кількісних шкал» для дистанційного навчального курсу «Системний аналіз та теорія прийняття рішень».

Методи розробки: для програмної реалізації тренажеру використано середовище розробки MS Visual Studio Code за допомогою мови програмування Python 3.10 та використання бібліотеки PyQT5.

<u>Методи дослідження</u>: При розробці тренажера з критерій максимуму середьного виграшу для кількісних шкал для дистанційного навчального курсу "Системний аналіз та теорія прийняття рішень" можуть бути використані наступні методи дослідження:

• Літературний огляд: Дослідження та аналіз наявних джерел, книг, наукових статей, документації, інтернет-ресурсів, пов'язаних з критерій максимуму середьного виграшу для кількісних шкал, алгоритмізацією та програмуванням. Цей метод дозволяє збирати інформацію про теоретичні аспекти, приклади використання та рекомендації щодо розробки тренажера.

• Тестування та зворотній зв'язок: Випробування тренажера з критерій максимуму середьного виграшу для кількісних шкал серед цільової аудиторії, яка складається з учнів курсу " Системний аналіз та теорія прийняття рішень ". Збір поверненого зворотного зв'язку, аналіз результатів

тестування та впровадження виправлень та вдосконалень на основі отриманих даних

3MICT

ВСТУП	8
1. ПОСТАНОВКА ЗАДАЧІ	10
2. ІНФОРМАЦІЙНИЙ ОГЛЯД	12
2.1 Огляд схожих за інформаційним складом тренажерів	12
2.2 Актуальність розробки тренажеру	14
3. ТЕОРЕТИЧНА ЧАСТИНА	16
3.1 Критерій максимуму середнього виграшу для кількісних шкал	16
3.2 Алгоритм роботи тренежера	18
3.3 Блок-схеми алгоритму тренажера	26
4. ПРАКТИЧНА ЧАСТИНА	27
4.1 Опис розробки тренажера	27
4.2 Інструкція до роботи з тренажером	33
ВИСНОВОК	41
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	43
ДОДАТОК А	45

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

Умовні позначення, символи,	Пояснення умовного позначення,
скорочення, терміни	скорочень, символів
Тренажер	Програма-вчитель, призначена для
	вивчення і закріплення знань з
	різноманітних тем; створюється для
	різноманітних операційних систем за
	бажанням та необхідністю.
ДК	Дистанційний курс.
ДН	Дистанційне навчання.
Python	Мова програмування.
PyQt5	Бібліотека Python
IDE Visual Studio Code	Середовище для написання коду.

ВСТУП

Сучасна освіта нерозривно пов'язана з використанням новітніх технологій та електронних ресурсів для підвищення якості навчання та залучення студентів до активного навчального процесу. Одним із напрямків розвитку дистанційної освіти є використання інтерактивних тренажерів, які дозволяють студентам вільно експериментувати та отримувати практичні навички у симуляційних середовищах.

Ця магістерська робота присвячена розробці програмного забезпечення тренажеру, який базується на критерії максимуму середнього виграшу для кількісних шкал у контексті системного аналізу та теорії прийняття рішень. Метою роботи є створення інтерактивного навчального інструменту, який допоможе студентам усвідомити та застосувати принципи системного аналізу та теорії прийняття рішень у вирішенні складних проблем.

Робота зосереджена на розробці тренажеру з використанням критерію максимуму середнього виграшу для кількісних шкал, який визначає оптимальні рішення на основі аналізу вхідних даних та обраних критеріїв. Цей критерій є важливим інструментом системного аналізу, оскільки дозволяє оцінити та порівняти різні альтернативи з урахуванням їхнього впливу на результати.

Дистанційний навчальний курс "Системний аналіз та теорія прийняття рішень" є основою для розробки тренажеру. Цей курс надає студентам фундаментальні знання та навички у сфері системного аналізу та теорії прийняття рішень, що є необхідними для вирішення проблем реального світу. Розроблений тренажер сприятиме практичному застосуванню отриманих знань та допоможе студентам зрозуміти і усвідомити основні концепції курсу через взаємодію з віртуальними сценаріями та завданнями.

В контексті розвитку дистанційної освіти ця робота має велике значення, оскільки вона відповідає потребам студентів у практичних навичках та інтерактивних методах навчання. Застосування розробленого тренажеру може сприяти покращенню процесу навчання та розширенню можливостей для самостійного вивчення системного аналізу та теорії прийняття рішень.

У подальшому тексті магістерської роботи будуть розглянуті докладніше цілі та завдання роботи, проведений огляд літератури, методологія розробки програмного забезпечення тренажеру, його реалізація, отримані результати та аналіз, а також висновки та рекомендації щодо подальших досліджень у цій галузі.

Мета роботи: розробка програмного забезпечення тренажеру з теми «Критерій максимуму середнього виграшу для кількісних шкал» для дистанційного навчального курсу «Системний аналіз та теорія прийняття рішень».

Об'єкт розробки: програма-тренажер на тему «Критерій максимуму середнього виграшу для кількісних шкал».

Предмет розробки: програмне забезпечення тренажеру з теми «Критерій максимуму середнього виграшу для кількісних шкал» для дистанційного навчального курсу «Системний аналіз та теорія прийняття рішень».

Методи розробки: для програмної реалізації тренажеру використано середовище розробки MS Visual Studio з платформою Unity 2021 за допомогою мови програмування С#.

Структура пояснювальної записки:

- вступ;
- постановка задачі;
- інформаційний огляд;
- теоретична частина;
- практична частина;
- висновки;
- перелік використаних інформаційних джерел,
- додатки.

Обсяг пояснювальної записки: 44 сторінки, з них основна частина – 40 сторінок, додатки – 2 сторінки, інформаційні джерела – 12 назв.

1. ПОСТАНОВКА ЗАДАЧІ

Головною задачею даної бакалаврської роботи є розробка програмного забезпечення тренажеру з теми «Критерій максимуму середнього виграшу для кількісних шкал» для дистанційного навчального курсу «Системний аналіз та теорія прийняття рішень» та впровадження розробленого тренажеру в систему MOODLE PUET.

Основні вимоги до розробки навчального тренажеру:

- 1. Розробка алгоритму та блок-схеми:
 - Скласти детальний алгоритм роботи навчального тренажеру, включаючи послідовність кроків та процеси, які відбуватимуться в системі.
 - Створити блок-схеми, що відображатимуть основні кроки алгоритму, з метою зрозумілого представлення логіки роботи тренажеру.
- 2. Вибір платформи та середовища розробки:
 - Обрати платформу, яка найкраще підходить для розробки навчального тренажеру.
 - Визначити середовище розробки, яке підтримує обрану платформу і забезпечує зручність у створенні та впровадженні тренажеру.
- 3. Вибір мови програмування:
 - Вибрати мову програмування, яка найкраще підходить для реалізації функціональності тренажеру.
 - Врахувати можливості та підтримку обраної мови програмування в обраному середовищі розробки та платформі MOODLE PUET.
- 4. Реалізація програмного тренажеру:
 - Розробити програму-тренажер на обраній платформі та мові програмування, відповідно до складеного алгоритму.
 - Забезпечити зручний та приємний інтерфейс користувача, що включатиме зрозумілі інструкції та навігацію.
- 5. Перевірка відповіді та повідомлення користувача:

- Реалізувати перевірку введеної відповіді користувачем з варіантами відповідей.
- Після вибору варіанту відповіді повідомити користувача про правильність обраного варіанту та надати необхідні коментарі або пояснення.
- 6. Валідація тренажера:
 - Перевірити програмний тренажер на валідність, забезпечивши коректну роботу всіх його функціональних можливостей.
 - Впевнитися, що тренажер відповідає поставленим вимогам і працює стабільно та ефективно.
- 7. Підтримка операційної системи Windows:
 - Забезпечити роботу тренажера на операційній системі Windows будь-якої версії.
 - Врахувати сумісність з різними версіями Windows, забезпечивши належну роботу та відповідність інтерфейсу.
- 8. Створення інструкції користувача:
 - Розробити детальну інструкцію користувача, яка пояснюватиме процес роботи з тренажером та надаватиме необхідні кроки та рекомендації для ефективного використання тренажеру.

Загальним результатом роботи буде розроблений та реалізований навчальний тренажер на базі критерію максимуму середнього виграшу для кількісних шкал, який буде інтегровано в систему MOODLE PUET та відповідатиме всім вимогам та обмеженням. Такий тренажер надасть студентам можливість отримати практичні навички та використати теоретичні знання в сфері системного аналізу та теорії прийняття рішень.

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1 Огляд схожих за інформаційним складом програмного забезпечення

Для інформаційного огляду було обрано два тренажери, що вже впроваджені в систему MOODLE PUET, а саме:

• Пояснювальна записка до дипломної роботи на тему Тренажер з теми «Знаходження вектору пріоритетів та відношення узгодження в методі аналізу ієрархій» дистанційного навчального курсу «Системний аналіз та теорія прийняття рішень» [2]

Автор: Гусар, Михайло Васильович

• Пояснювальна записка до бакалаврської роботи на тему Алгоритмізація та програмна реалізація тренажеру з теми «Обчислення коефіцієнта впевненості» дистанційного навчального курсу «Інтелектуальні інформаційні системи» [3]

Автор: Яремченко, С. І.

Під час дослідження роботи Гусара, за темою «Знаходження вектору пріоритетів та відношення узгодження в методі аналізу ієрархій» було знайдено наступні позитивні сторони розробленого тренажеру:

- На стартовій сторінці виводиться тема тренажера, розробник;
- В тренажері є як теоретична так і практична інформація;
- Існує вибір мови тренажеру українська або англійська;
- Після вибору задачі виведеться її умова;

• Якщо в будь-який момент розрахунків було допущено помилку, то виведеться повідомлення про помилку і відобразяться розрахункові формули.

Під час дослідження роботи Гусара, за темою «Знаходження вектору пріоритетів та відношення узгодження в методі аналізу ієрархій» було знайдено наступні негативні сторони розробленого тренажеру:

• Не доцільно використано робоче поле тренажеру;

• Занадто малий шрифт під час роботи на комп'ютерах з високою діагоналлю;

• Повідомлення про помилку з'являється не завжди;

• Відсутня кнопка повтору роботи;

• Відсутні завдання з теорії.

Під час дослідження роботи Яремченка, за темою «Обчислення коефіцієнта впевненості» було знайдено наступні позитивні сторони розробленого тренажеру:

• На стартовій сторінці виводиться тема тренажера, розробника, що дає змогу користувачу одразу зрозуміти який тренажер він проходить та з якої теми;

• В тренажері є як теоретична так і практична інформація що позитивно впливає на студентів, так як якщо користувач відповів не вірно, за допомогою підказки може дізнатись саме чому це є не вірна відповідь;

• Під час роботи в будь-який момент можна повернутися до меню через відповідну кнопку та повторити роботу з тренажером так як користувач не завжди має час на проходження усього тесту;

• Неможливо продовжити роботу з тренажером не обравши правильний варіант відповіді.

Під час дослідження роботи Яремченка, за темою «Обчислення коефіцієнта впевненості» було знайдено наступні негативні сторони розробленого тренажеру:

• Одноманітні практичні завдання які негативно впливають на користувача;

• Кнопка закриття повідомлення про помилку іноді не працює;

• Неприємний візуальний дизайн тренажеру, що відволікає від роботи користувача.

• Кнопки для вибору варіанту відповіді під час роботи з тестами натискаються декілька разів.

2.2 Актуальність розробки програмного забезпечення

Проаналізувавши всі помилки та зауваження, виявлені в оглядах тренажерів, розроблених Гусаром та Яремченком, можна зробити важливі кроки для покращення розробки тренажеру для магістерської роботи. З метою створення цікавого та привабливого тренажеру, який би відповідав вимогам та надавав якісний користувацький досвід, додаткові аспекти, що потрібно врахувати в процесі розробки тренажеру, включають:

• Приділити особливу увагу розробці інтуїтивно зрозумілого інтерфейсу, що дозволить користувачам швидко орієнтуватись у тренажері та без зусиль почати виконання завдань;

• Підвищити привабливість тренажеру шляхом включення різноманітних практичних завдань, де користувачі зможуть самостійно формулювати відповіді та перевіряти їх правильність, надаючи більш інтерактивний та залучаючий досвід;

• Гарантувати, що всі кнопки та функціонал працюють коректно та без зайвих зусиль з боку користувача.

• Врахувати всі недоліки попередніх тренажерів, забезпечуючи повну функціональність та надійність всіх елементів тренажеру.

• Забезпечити, щоб повідомлення, що відображатимуться користувачу, були чіткими, зрозумілими та надавали інформацію, яка допоможе користувачеві розуміти, для чого воно призначене та як виконати необхідні дії.

• Розробити адаптивний дизайн, який забезпечить зручне використання тренажеру на різних розмірах екранів, включаючи як великі монітори, так і пристрої з малими діагоналями. • Додати кнопку або функціонал, який дозволить користувачам повторити роботу з тренажером без необхідності виходити з програми та повторно запускати її, забезпечуючи більш зручний та ефективний процес навчання.

Ці нові аспекти допоможуть створити магістерський тренажер, який не тільки буде демонструвати покращення в порівнянні з попередніми роботами, а й надасть користувачам захоплюючий та змістовний досвід навчання.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1 Критерій максимуму середнього виграшу для кількісних шкал

Критерій максимуму середнього виграшу є одним із ключових критеріїв, який використовується в раціональній стратегії при прийнятті рішень. Він дозволяє визначити оптимальне рішення, враховуючи середній виграш, пов'язаний з кожною альтернативою.

Цей критерій дозволяє враховувати не лише окремі виграші альтернатив, а й їх середні значення, що забезпечує більш об'єктивну оцінку і порівняння рішень. Використання коефіцієнтів важливості рішень допомагає визначити оптимальне рішення, яке забезпечить найбільший середній виграш у контексті розглянутої проблеми.

Застосування критерію максимуму середнього виграшу може мати широкий спектр застосувань, включаючи сфери бізнесу, економіки, наукових досліджень та соціального вибору. Цей критерій надає можливість систематизувати та порівнювати різні альтернативи з урахуванням їх середнього виграшу, що сприяє більш об'єктивному та збалансованому процесу прийняття рішень.

Якщо переваги рішень на множині альтернатив вимірюються в *кількісній* шкалі, то середній виграш обчислюється як математичне сподівання виграшу:

$$\beta_i = \sum_{k=\varphi}^{\pi} p_k \cdot f_{ik}$$
, $i = 1, 2, ..., m$

де pk – ймовірність k-ї альтернативи, fik – значення функції переваг, яка оцінює i-е рішення для k-ї альтернативи.

Відповідно до загального правила вибору рішень (10.2) правило вибору рішення, яке відповідає критерію максимуму середнього виграшу, має вигляд

$$\max \beta_i \ (\beta_1, \beta_2, \dots, \beta_m) \Rightarrow Y^*$$

Розглянемо частинні випадки. Нехай вірогідність появи всіх альтернатив однакова (всі ймовірності рівні між собою). Оскільки сума ймовірностей усіх альтернатив дорівнює одиниці, то для рівних ймовірностей одержуємо, що всі вони дорівнюють pk=1/n, де n – кількість альтернатив. У цьому випадку середні виграші рішень обчислюються за формулою

$$\beta_i = \frac{1}{n} \sum_{k=1}^n f_{ik}$$

Множник 1/п не впливає на визначення максимуму, тому середні виграші рішень можна обчислити за формулою

$$\beta_i = \sum_{k=1}^n f_{ik}$$

Якщо має місце тільки одна альтернатива, наприклад Sj, то її поява є достовірною подією і, отже, pj=1. Решта альтернатив мають нульові ймовірності появи: pk=0, j≠k. У цьому випадку середні виграші просто дорівнюють значенням функції переваг для j-ї альтернативи

$$\beta_i = f_{ij}$$
 , $i = 1, 2, ..., m$

3.2 Алгоритм роботи програмного забезпечення

Робота з програмним забезпеченням виконується за наступним алгоритмом:

Крок 1. Запуск програмного забезпечення.

Крок 2. Користувачу надається доступ до початкового меню тренажеру, в якому видно тему та розробника роботи.

Крок 3. На початковому меню тренажера за допомогою кнопки «Почати» користувач переходить до практичних завдань.

Крок 4. Користувач отримує практичне завдання у вигляді тесту:

Завдання 1. Що таке критерій максимуму середнього виграшу в системному аналізі?

а) Метод максиміну

b) Метод максимаксу

с) Метод мінімаксу

d) Метод середнього виграшу

Правильна відповідь - d)

Після вибору правильної відповіді користувач отримує доступ до кнопки «Далі», що перенаправляє його до наступного практичного завдання.

Крок 5. Користувач отримує практичне завдання у вигляді тесту:

Завдання 2. Яка основна мета використання критерію максимуму середнього виграшу?

а) Мінімізація ризику

b) Максимізація можливості

с) Мінімізація витрат

d) Максимізація середнього виграшу

Правильна відповідь - b)

Після вибору правильної відповіді користувач отримує доступ до кнопки «Далі», що перенаправляє його до наступного практичного завдання.

Крок 6. Користувач отримує практичне завдання у вигляді тесту:

Завдання 3. Яка умова повинна бути виконана для застосування критерію максимуму середнього виграшу?

а) Відомий розподіл імовірності

b) Незалежність змінних

с) Наявність альтернативних варіантів

d) Максимальна вигода від прийнятого рішення

Правильна відповідь - а)

Після вибору правильної відповіді користувач отримує доступ до кнопки «Далі», що перенаправляє його до наступного практичного завдання.

Крок 7. Користувач отримує практичне завдання у вигляді тесту:

Завдання 4. Яку інформацію надає матриця виграшів при застосуванні критерію максимуму середнього виграшу?

а) Імовірності кожного стану природи

b) Середні виграші для кожного рішення

с) Максимальні виграші для кожного рішення

d) Ризики для кожного рішення

Правильна відповідь - а)

Після вибору правильної відповіді користувач отримує доступ до кнопки «Далі», що перенаправляє його до наступного практичного завдання.

Крок 8. Користувач отримує практичне завдання у вигляді тесту:

Завдання 5. Який підхід використовується для визначення оптимального рішення за критерієм максимуму середнього виграшу?

а) Метод максиміну

b) Метод максимаксу

с) Метод мінімаксу

d) Метод середнього виграшу

Правильна відповідь - d)

Після вибору правильної відповіді користувач отримує доступ до кнопки «Далі», що перенаправляє його до наступного практичного завдання.

Крок 9. Користувач отримує практичне завдання у вигляді тесту:

Завдання 6. Яка змінна у системному аналізі відповідає за рішення, яке приймається?

а) Випадкова змінна

b) Змінна стану природи

с) Змінна прибутку

d) Контрольована змінна

Правильна відповідь - d)

Після вибору правильної відповіді користувач отримує доступ до кнопки «Далі», що перенаправляє його до наступного практичного завдання.

Крок 10. Користувач отримує практичне завдання у вигляді тесту:

Завдання 7. Що представляє собою функція сприйняття ризику?

а) Відображення ставлення до ризику при прийнятті рішень

b) Кількісне визначення величини ризику

с) Оцінка імовірності втрат

d) Визначення оптимального рішення

Правильна відповідь - а)

Після вибору правильної відповіді користувач отримує доступ до кнопки «Далі», що перенаправляє його до наступного практичного завдання.

Крок 11. Користувач отримує практичне завдання у вигляді тесту:

Завдання 8. Оберіть вираз знаходження критерію максимуму середнього виграшу?

1) 1 + 1 + n, де n, m – ймовірність альтернатив 1 та 2;

2) 1*n + 1 m, де n, m – ймовірність альтернатив 1 та 2;

3) 1*n + 1*m, де n,m – ймовірність альтернатив 1 та 2;

4) Немає правильної відповіді

Правильна відповідь - 3)

Після вибору правильної відповіді користувач отримує доступ до кнопки «Далі», що перенаправляє його до наступного практичного завдання.

Крок 12. Користувач отримує практичне завдання у вигляді тесту:

Завдання 9. Якість рішення за критерієм максимуму середнього виграшу залежить від:

а) Імовірності прийняття кожного рішення

b) Кількості станів природи

с) Значень виграшів та ймовірностей

d) Попередніх знань

Правильна відповідь - с)

Після вибору правильної відповіді користувач отримує доступ до кнопки «Далі», що перенаправляє його до наступного практичного завдання.

Крок 13. Користувач отримує практичне завдання у вигляді тесту:

Завдання 10. Яку інформацію надає матриця ймовірностей при застосуванні критерію максимуму середнього виграшу?

а) Середній виграш для кожного рішення

b) Ймовірність виникнення кожного стану природи

с) Максимальний виграш для кожного рішення

d) Ймовірність виникнення кожного ризикового інтервалу

Правильна відповідь - b)

Після вибору правильної відповіді користувач отримує доступ до кнопки «Далі», що перенаправляє його до наступного практичного завдання.

Крок 14. Користувач отримує практичне завдання у вигляді тесту:

Завдання 11. Якість прийнятого рішення за критерієм максимуму середнього виграшу перевіряється за допомогою:

а) Графіка Парето

b) Принципу максимуму середнього

с) Методу симплекс-планування

d) Методу сенситивних аналізів

Правильна відповідь - b)

Після вибору правильної відповіді користувач отримує доступ до кнопки «Далі», що перенаправляє його до наступного практичного завдання.

Крок 15. Користувач отримує практичне завдання у вигляді тесту:

Завдання 12. Який метод допомагає визначити, яке рішення краще, коли змінні є випадковими і мають ймовірнісний характер?

а) Метод максиміну

b) Метод максимаксу

с) Метод мінімаксу

d) Метод головного пропорційного відношення

Правильна відповідь - с)

Після вибору правильної відповіді користувач отримує доступ до кнопки «Далі», що перенаправляє його до наступного практичного завдання.

Крок 16. Користувач отримує практичне завдання у вигляді тесту:

Завдання 13. Якість рішення за критерієм максимуму середнього виграшу залежить від:

а) Кількості альтернативних варіантів

b) Рівня ризику прийняття рішення

с) Величини середнього виграшу для кожного рішення

d) Повного відсутності ризику

Правильна відповідь - с)

Після вибору правильної відповіді користувач отримує доступ до кнопки «Далі», що перенаправляє його до наступного практичного завдання.

Крок 17. Користувач отримує практичне завдання у вигляді тесту:

Завдання 14. Яку змінну у системному аналізі вважають стохастичною?

а) Випадкова змінна

b) Змінна стану природи

с) Змінна прибутку

d) Контрольована змінна

Правильна відповідь - а)

Після вибору правильної відповіді користувач отримує доступ до кнопки «Далі», що перенаправляє його до наступного практичного завдання.

Крок 18. Користувач отримує практичне завдання у вигляді тесту:

Завдання 15. Якість рішення за критерієм максимуму середнього виграшу може знизитися при:

а) Збільшенні кількості альтернативних варіантів

b) Збільшенні ризику

с) Зменшенні величини середнього виграшу для кожного рішення

d) Впровадженні інновацій

Правильна відповідь - с)

Після вибору правильної відповіді користувач отримує доступ до кнопки «Далі», що перенаправляє його до наступного практичного завдання.

Крок 19. Користувач отримує практичне завдання у вигляді тесту:

Завдання 16. Який з методів допомагає знайти оптимальне рішення в умовах невизначеності та обмежень ресурсів?

а) Метод максиміну

b) Метод максимаксу

с) Метод мінімаксу

d) Метод головного пропорційного відношення

Правильна відповідь - d)

Після вибору правильної відповіді користувач отримує доступ до кнопки «Далі», що перенаправляє його до наступного практичного завдання.

Крок 20. Користувач отримує практичне завдання у вигляді тесту:

Завдання 17. Якість прийнятого рішення за критерієм максимуму середнього виграшу оцінюється за допомогою:

а) Графіка Парето

b) Принципу максимуму середнього

с) Методу симплекс-планування

d) Методу сенситивних аналізів

Правильна відповідь - d)

Після вибору правильної відповіді користувач отримує доступ до кнопки «Далі», що перенаправляє його до наступного практичного завдання.

Крок 21. Користувач отримує практичне завдання у вигляді тесту:

Завдання 18. Який метод допомагає визначити, яке рішення краще, коли змінні мають випадковий характер?

а) Метод максиміну

b) Метод максимаксу

с) Метод мінімаксу

d) Метод головного пропорційного відношення

Правильна відповідь - с)

Після вибору правильної відповіді користувач отримує доступ до кнопки «Далі», що перенаправляє його до наступного практичного завдання.

Крок 22. Користувач отримує практичне завдання у вигляді тесту:

Завдання 19. Якість рішення за критерієм максимуму середнього виграшу залежить від:

а) Кількості альтернативних варіантів

b) Рівня ризику прийняття рішення

с) Величини середнього виграшу для кожного рішення

d) Повного відсутності ризику

Правильна відповідь - с)

Після вибору правильної відповіді користувач отримує доступ до кнопки «Далі», що перенаправляє його до наступного практичного завдання

Крок 23. Користувач отримує практичне завдання у вигляді тесту:

Завдання 20. Яку змінну у системному аналізі вважають стохастичною?

а) Випадкова змінна

b) Змінна стану природи

с) Змінна прибутку

d) Контрольована змінна

Правильна відповідь - b)

Після вибору правильної відповіді користувач отримує доступ до кнопки «Далі», що перенаправляє його до наступного практичного завдання

Крок 24. Після завершення роботи над практичними матеріалами тренажеру користувач переходить до фінального вікна тренажеру у якому може завершити або повторити роботу за допомогою відповідних кнопок.

3.3 Блок-схеми алгоритму програмного забезпечення



Рисунок 3.1 – Блок-схеми алгоритму роботи тренажера

4. ПРАКТИЧНА ЧАСТИНА

4.1 Опис розробки програмного забезпечення

Для створення тренажеру була обрана мова програмування Python з використанням бібліотеки PyQT5.

Мова прогамування Python ϵ одна з найпопулярніших мов програмування. Вона має простий та зрозумілий синтаксис, що дозволяє легко вивчтити дану мову. Такоє вона є найрозповсюдженішою, що дає змогу більш очне розуміння того, на що здатна дана мова програмування. Завдяки її популярності, дана мова регулярно оновлюється, приноситься багато позитивних змін до неї. Також дана мова має широкий спектр бібліотек та інструментів, що дають можливість спростити розробку, або навпаки, розширити програму. Тому вибір саме Python забезпечує продуктивність та гнучкість для розробки тренажеру.

Бібліотека PyQt є однією з популярних бібліотек для розробки графічних інтерфейсів (GUI) на мові програмування Python. Вона надає зручний доступ до функцій та можливостей фреймворку Qt, що є потужним інструментом для створення кросплатформенних програм з графічним інтерфейсом.

РуQt була розроблена Ривером Банксом і вперше випущена у 1998 році. Вона є обгорткою навколо низькорівневої бібліотеки Qt, яка була розроблена фінською компанією Trolltech (зараз Digia, a Qt тепер належить компанії The Qt Company). РуQt забезпечує доступ до Qt для мови Python і дозволяє легко створювати зручні та потужні графічні інтерфейси.

PyQt може використовуватися для розробки різноманітних десктопних програм, включаючи редактори тексту, графічні редактори, інструменти аналізу даних, ігри, програми для наукової роботи, програми для управління базами даних та багато іншого. Оскільки Qt є кросплатформеним фреймворком, програми, розроблені з використанням PyQt, можуть

працювати на різних операційних системах, таких як Windows, macOS та Linux, з мінімальними змінами коду.

Обрання PyQt для розробки програми з кількох причин. По-перше, Python є однією з найкращих мов програмування і PyQt надає зручний і простий спосіб створення графічних інтерфейсів з використанням цієї мови. Він має багатий набір документації та активну спільноту, що робить процес розробки ефективним та приємним.

По-друге, важливою перевагою PyQt є те, що вона надає повний доступ до всіх можливостей Qt, що дозволяє створювати дуже гнучкі та високоефективні програми. Також, кросплатформенні можливості Qt забезпечують, що моя програма буде працювати на різних платформах, що є важливим фактором для користувачів з різних операційних систем.

PyQt - потужна та зручна бібліотека, яка дозволяє швидко створювати професійні графічні інтерфейси для своїх програм на мові Python. За допомогою програми Qt Designer можливо з легкістью створити будь-який інтерфейс який буде мати різноманітний інтерфейс та налаштувати будьякий зовнішній вигляд.

File Edit Form View Settings Window Help				
D 💋 🕞 🕞 📑 📑 💀 🖳 💷 😑 H 🕱 🚟 🗄	5 🔍			
Widget Box 8 ×		Object li	spector	6 ×
Filter		Object	Class	
✓ Layouts		Y Mair	Window QMainWindow	
Vertical Layout			💩 centralwidget QWidget	
Horizontal Lavout			nenubar QMenuBar	
Grid Lawart		1	tatusbar QStatusBar	
233 Gran La sut				
Server				
Bell Harinantal Sancer				
and Honzontal space	00 MainWindow - untitled			
2 Vertical Spacer	Type Here			
Buttons				
Push Button				
C Tool Button				
Radio Button				
Check Box				v
Ommand Link Button		Property	Editor	ff ×
Valog Button Box		Eller		- ×
 Item Views (Model-Based) 		Monthled	tur i OktoioMindaur	
List View		Paninto	ow r gearminow	(alua
™® Tree View		Ргоренту		alue
Table View		- Que	rtName I	AsiaWindow .
Columo View		× 000	daet	
V Item Widents (Item-Based)		wind	owModality	IonModal
List Widget		enab	led	2
The Wildow		🔆 🔆 🔆 🔆 🕹 🗸 🗸 🗸 🗸 🗸 🗸 🗸 🗸	netry	(0, 0), 640 x 480]
Table Widget		· · · · · · · · · · · · · · · · · · ·		
Containers			Vidth (40
Group Box		Resource	e Browser	6° X
Scroll Area			•	Filter
Tool Box		<re></re>	surce root>	
Tab Widget		21 A		
Stacked Widget				
50 Frame				
22 Widget				
HDI Area				
Dock Widget				
V Input Widgets				
Combo Box				
Font Combo Box				
Edit Line Edit				
AII Text Edit		Action 8	ditor Resource Browser	

Рисунок 4.1 – Вікно редактору QT Designer

QT Designer - це візуальний інструмент для створення графічних інтерфейсів користувача для програм, що базуються на бібліотеці PyQt (або PySide - ще один альтернативний фреймворк для Python). Він дозволяє розробникам створювати GUI за допомогою миші та графічного інтерфейсу, уникаючи величезної кількості вручну написаного коду.

Створення форм та вікон за допомогою QT Designer:

- 1. Відкриття QT Designer: Після встановлення PyQt (або PySide) ви можете відкрити QT Designer, запустивши команду designer з командного рядка або виконавши відповідну дію через вашу інтегровану середовище розробки (IDE).
- 2. Вікно QT Designer: Після запуску ви побачите вікно QT Designer, яке надає вам ряд інструментів, таких як кнопки, поля вводу, таблиці та багато іншого.
- 3. Створення форми: Для створення форми, просто перетягніть бажані елементи з панелі інструментів на вікно QT Designer. Ви можете розташовувати їх на формі і змінювати їх властивості за допомогою правого кліку та контекстного меню.
- 4. Підключення сигналів і слотів: QT Designer дозволяє підключати сигнали (наприклад, подія "клік" на кнопці) до слотів (функції, яку ви хочете викликати при виникненні сигналу). Це можна зробити за допомогою функції "Сигнали та слоти" у QT Designer.

QSS (Qt Style Sheets) - це мова стилів, яка дозволяє налаштовувати зовнішній вигляд елементів інтерфейсу в QT. Вона дуже схожа на CSS (Cascading Style Sheets) і дозволяє змінювати кольори, розміри, шрифти, фонові зображення та багато іншого.

QT Designer надає можливість застосувати QSS до вашої форми, щоб налаштувати зовнішній вигляд елементів GUI. Ви можете редагувати стилі прямо в QT Designer або зберегти їх в окремому файлі і підключити до вашої програми під час її виконання.

Блоки в QT Designer - це контейнери, які дозволяють групувати кілька елементів інтерфейсу разом. Вони допомагають організовувати та керувати елементами в більш структурованому способі. Блоки можуть містити інші блоки або окремі елементи, і ви можете змінювати їх властивості як єдину одиницю.

Загалом, QT Designer спрощує процес створення графічних інтерфейсів для ваших програм, а QSS дозволяє налаштувати зовнішній вигляд інтерфейсу за допомогою стилів. Використовуючи QT Designer та QSS разом з бібліотекою PyQt, ви можете швидко створити привабливий і зручний GUI для своїх програм.

ut Designer				- 0 ×
File Edit Form View Settings Window Help				
0 🖉 🕞 👘 🖓 🖷				
Widget Box & X		Object Inspector		6 ×
Filter		Object	Class	^
 Layouts 		 MainWindow 	QMainWindow	
Vertical Layout		✓	QWidget	
Horizontal Layout		verticalLayout	QVBoxLayout	
Grid Layout		answer_Button1	QPushButton	
223 Child Layout	💽 ТРЕНАЖЕРА «РЕГУЛЯРНІ ВИРАЗИ» - Дипломная.ui*	answer_Button2	QPushButton	
55 Form Layout	Three was No.	answer Button4	OPushButton	
Spacers	питання ле	horizontalLayout	dHBoxLavout	
Horizontal Spacer		horizontalSpacer	Spacer	
Vertical Spacer		next_Button	QPushButton	
Buttons		numberQuestion	QLabel	
Push Button		questionEdit	QTextEdit	
Tool Button				
Radio Button				
Check Box				v
Command Link Button		Property Editor		6 ×
🔨 Dialog Button Box		Filter		/
 Item Views (Model-Based) 		MainWindow : OMainWindow		
List View		Property	Value	
Tree View		X 00birst	toruc.	I
Table View		objectName	MainWindow	
iable view		× OWidget	mannindow	
Column View		windowModality	NonMedal	
Item Widgets (Item-Based)	Bignosia» 1	enabled		
List Widget		✓ geometry	[(0, 0), 672 × 664]	
*8 Tree Widget		X	0	
Table Widget		Y	0	
Containers	Banoedo 2	Middle	672	
Group Box		Resource Browser		6 ×
Scroll Area		/ C	Filter	
Tool Box	Banosia» 3	<resource root=""></resource>		
Tab Widget				
Stacked Widget				
Frame	Banosa» 4			
Widget				
MDI Area	Ass 1			
Dock Widget				
V Input Widgets				
Combo Bax				
Font Combo Box				
well Line Edit				
AT Too Edia		Action Editor Resource Brows	er	
LEAST TEXT CUT				

Рисунок 4.2 – Вікно питань



Рисунок 4.3 – Початкове вікно тренажеру

🧧 Qt Designer			-	0 X
File Edit Form View Settings Window Help				
D 💋 🕞 🕒 📑 🏪 🖳 🖏 🛄 😑 H X 🗱 🗄	õ 🔍			
Widget Box & X		Object Inspector		ē×
Filter		Object Class		
✓ Lavouts		Y Bialon ODialon		
Vertical Laurut		closeButton OPushButton		
Maintal Lucit		label_2 QLabel		
999 Pointointai Layout		refreshButton QPushButton		
Grid Layout				
Form Layout				
✓ Spacers				
Horizontal Spacer	01 TPEHAXEPA «Kinakicthi uakanu» - finalWindow.ui*			
2 Vertical Spacer				
✓ Buttons				
Push Button				
Tool Button				
Ratio Button				
Charle Date				-
Check Box		D		
Command Link Button	Pizzue	Property Editor		D A
X Dialog Button Box	DIIdiO	Filter		/
Item Views (Model-Based)		Dialog : QDialog		
List View	December 2010 and a second sec	Property	Value	
No Tree View	Ви успішно закінчили тест	Y QObject		
Table View	Бажаете повторити?	objectName	Dialog	
Column View	Важаете повторити:	✓ QWidget		
V Item Widnets (Item Based)		enabled	2	
List Midnest		✓ geometry	[(0, 0), 640 × 480]	
		X	0	
The Widget		Y	0	
Table Widget		Width	640	
 Containers 		Height	480	
Group Box	-	 sizePolicy bissingstal Ballaci 	(Preferred, Preferred, 0, 0)	
Scroll Area	гювторити	Vertical Policy	Preferred	
Tool Box	Закінати	Herizontal Stretch	0	
Tab Widget		Vertical Stretch	0	
E Charles Midant		✓ minimumSize	0 x 0	
Jacked widger		Width	0	
Frame		Height	0	
Widget		V manimum Cine	16777315 16777315	-
MDI Area		Resource Browser		6 ×
Dock Widget		/ C	Filter	
✓ Input Widgets		<resource root=""></resource>		
Combo Box				
Font Combo Box				
Contraction for the second sec				
		Artist Editor Descurse Drouges		
1/311 Text Edit		Resource browser		

Рисунок 4.4 – Фінальне вікно тренажеру

Даний тренажер складається з 3 основних вікон. На даних вікнах в Qt Designer розтавляємо всі кнопки, текст та допоміжні плоки для того, щоб вже за допомогою мови програмування Python налаштувати коректне роботу ціх елементів керування. На вікні питань, налаштовуємо так, щоб після кожної відповіді користувачу з'являлось нове питання. Всі питання ми повинні зберігати та зробити так, щоб була можливість швидко їх редагувати. Для цього ми використовуємо JSON файл, в якому буде саме питання, варіанти



відповідей та саме правильний варінт.

Рисунок 4.5 – Структура Json файлу

На мальнку 4.5 бачимо стурктуру файлу Json де записані питання.

• Поле question – це поле де написано саме питання яке буде відображатись користувачу.

• Поле answer - це є масив відповідей, де знаходяться всі варіанти відповідей для цього питання.

• Поле correct_answer – це поле відповідає за правильну відповідь, саме там ми записуємо вірний варіант відповіді для даного питання, щоб

програма могла коректно працювати.

Завдяки всім цім технологіям, і було створено тренажер на мові програмуванні Python з використанням бібліотек PyQT5 та JSON.

4.2 Інструкція до роботи з програмним забезпеченням

Коли користувач запускає програму, він бачить стартове вікно

КРИТЕРІЙ МАКСИМУМУ СЕРЕДНЬОГО ВИГРАШУ ДЛЯ КІЛЬКІСНИХ ШКАЛ» ДИСТАНЦІЙНОГО НАВЧАЛЬНОГО КУРСУ «СИСТЕМНИЙ АНАЛІЗ ТА ТЕОРІЯ ПРИЙНЯТТЯ РІШЕНЬ»

Виконавець роботи: Пилипенко Назар Вікторович Науковий керівник: к.ф.-м.н., доц. Черненко Оксана Олексіївна

Розпочати

тренажеру де написана тема тренажеру, та того, хто створив даний тренажер. Рисунок 4.6 – Початкове вікно тренажеру

Після цього у користувача є тількі один варіант, це нажати кнопку розпочати, після чого його направить на сам тренажер, де користувачу будуть з'являтися питання на які користувач буде відповідати.



Рисунок 4.7 – Вікно питань

Програма дістає з Json файлу питання, та вставляє у відповідні елементи саме питання та варіанти відповідей. Варіанти відповідей це є кнопки. Користувач не зможе далі пройтись по тренажеру, якщо він не вірно відповість. Прогамма буде давати йому знати коли відповідь вірна, а коли ні.



Рисунок 4.8 – Демонстрація при нажаті кнопки «Далі»

Як бачимо що при натискіні кнопки «Далі», питання не змінилося і питання залишилось.

Коли користувач відповідає не вірно то його варіант відповіді підсвічується червноим, та з'являється ще діалогове вікно про те, що даний варіант відповіді не вірний.

ТРЕНАЖЕР «Кількістні шкали» —	- 🗆	\times
Питання № 1		
Що таке критерій максимуму середнього виграшу в системни	ому аналізі?	
Відповідь невірна		
ОК		
Метод максиміну		
Метод максимаксу		
Метод мінімаксу		
Метод середнього виграшу		
	Д	алі

Рисунок 4.9 – Демонстрація невірної відповіді

Після цього коли користувач тасне «ОК», діалогові вікно зникне, але відповідь і надалі буде підсвічуватись червоним, що дає користувачу змогу побачити який варіант він обрав. Якщо користувач відповідає вірно, то кнопка підсвічується зеленим, що дає змогу користувачу зрозуміти, що саме цей варіант відповіді вірний і він може натиснути кнопку «Далі», щоб відповідати на інші питання.



Рисунок 4.10 – Вірна відповідь



Рисунок 4.11 – Інше питання після натиску кнопки «Далі»

На малюнку 4.11 відображено вже інше питання, користувач має все тіж можливості, що і коли відповідав на питання №1, невірна відповідь буде підсвічуватись червоним з діалоговим вікном, а вірна буде підсвічуватись зеленим. Даний інтерфейс є зрозумілим для користувача та інтуїтивним зрозумілим.

Також програма має адаптивність, що дає змогу користувачу редагувати розмір вікна залежно від його монітору.

TPEHAXEP «Kinademi usean» — O	×
Питання № 2	
Ra soosa wts sexportawe rpmgio worwy ropsawor serpeu?	-
Metajárpoky	
Macarbuja nozvecni	
Medical a mptr	
	-
Maconsaut optigano's in ryany	
fan 1	

Рисунок 4.11 – Максимальний розмір програми

ТРЕНАЖЕР «Кількістні шкали» —		×		
Питання № 2				
Яка основна мета використання критерію максимуму виграшу?	/ середнь	ого		
Мінімізація ризику				
Максимізація можливості				
Мінімізація витрат				
Максимізація середнього виграшу				
	Да	лі		

Рисунок 4.12 – Мінімальний розмір вікна

Даний функціонал дуже зручний, тому що у кожного користувача різний розмір монітору і програма може підлаштовуватись під різні монітори.

Коли користувач відповів на всі питання, в кінці він бачить фінальне вікно тренажеру яке має лише 2 кнопки

- Повторити
- Закінчити

Кнопка «Повторити» запускає тренажер заново з першого питання, а



кнопка «Закінчити» завершує тренажер та виключає програму

Рисунок 4.13 – Кінцеве вікно з варіантом вибору

📰 ТРЕНАЖЕР «Кількістні шкали» —		\times	
Питання № 1			
Що таке критерій максимуму середнього виграшу в системному а	аналізі?		
Метод максиміну			
Метод максимаксу			
Метод мінімаксу			
Метод середнього виграшу			
	Дал	ni	

Рисунок 4.14 – Тренажер після натиску кнопки «Повтор» на фінальному вікні

ВИСНОВОК

В магістерській роботі було розроблено програмне забезпечення тренажеру з теми "Критерій максимуму середнього виграшу для кількісних шкал" для дистанційного навчального курсу "Системний аналіз та теорія прийняття рішень". Для розробки була використана мова програмування Python та бібліотека PyQT5.

Метою роботи було створення ефективного та зручного тренажеру, який допомагав би студентам зрозуміти та застосовувати критерій максимуму середнього виграшу у системному аналізі та прийнятті рішень. Під час розробки було враховано різноманітні вимоги та особливості, що забезпечили зручний та логічний інтерфейс для користувача.

В результаті проведеного дослідження було детально проаналізовано існуючі тренажери з подібних навчальних курсів, виявлені їх переваги та недоліки, що надало змогу уникнути повторення вже відомих проблем при розробці даного тренажеру.

Під час розробки програмного забезпечення була використана бібліотека РуQT5, що дозволила створити інтерактивний та зручний інтерфейс для користувачів. Мова програмування Python, завдяки своїй простоті та потужності, дозволила ефективно реалізувати алгоритми та функціонал тренажеру.

Отримане програмне забезпечення відповідає всім поставленим вимогам та функціональним вимогам до тренажеру. Воно забезпечує користувачам зручне та легке оволодіння критерієм максимуму середнього виграшу, допомагає краще зрозуміти його суть та застосувати на практиці.

Завдяки використанню Python та PyQT5 було досягнуто високої ефективності та функціональності програмного забезпечення, що дозволяє швидко та зручно виконувати навчальні завдання з критерієм максимуму середнього виграшу.

Загальний висновок полягає в тому, що розроблене програмне забезпечення є корисним та цінним інструментом для навчання та розвитку навичок у галузі системного аналізу та теорії прийняття рішень. Його впровадження у систему MOODLE PUET дозволить підвищити якість та ефективність навчання студентів у даній дисципліні. Результати дослідження можуть бути використані для подальшого вдосконалення та розширення функціоналу тренажеру, що сприятиме подальшому розвитку освітньої сфери.

Основними позитивними сторонами виконаної роботи та розробленого тренажеру являються:

- Складено алгоритм роботи навчального тренажеру;
- Створено блок-схеми в відповідності до розробленого алгоритму;

• Обрано платформу та середовище для розробки, а також мову для програмування навчального тренажеру згідно розробленого алгоритму;

• Реалізовано програму-тренажер відповідно на обраній платформі за допомогою обраної мови програмування,

• Створено зручний та приємний в використанні інтерфейс тренажеру,

• Забезпечено перевірку введеної відповіді, після вибору варіанту відповіді повідомити користувача про правильність обраного варіанту,

• Здійснено перевірку тренажера на валідність,

• Забезпечено роботу тренажера на операційній платформі Windows будь-якої версії,

• Створено інструкцію користувача для роботи з тренажером.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1. Черненко 0.0. Методичні рекомендації оформлення щодо пояснювальних записок ДО курсового проєкту для студентів спеціальності 122 Комп'ютерні науки освітня програма «Комп'ютерні науки»ступеня бакалавра, магістра / О. О. Черненко – Полтава : РВВ ПУЕТ, 2022. – 59 с.
- Гусар, Михайло Васильович, Пояснювальна записка до дипломної роботи на тему Тренажер з теми «Знаходження вектору пріоритетів та відношення узгодження в методі аналізу ієрархій» дистанційного навчального курсу «Системний аналіз та теорія прийняття рішень» /Гусар, Михайло Васильович [Електронний ресурс].– Режим доступу до ресурсу: http://dspace.puet.edu.ua/handle/123456789/8993
- 3. Яремченко, С. І., Пояснювальна записка до бакалаврської роботи на тему Алгоритмізація та програмна реалізація тренажеру з теми «Обчислення коефіцієнта впевненості» дистанційного навчального курсу «Інтелектуальні інформаційні системи»/Яремченко, С. І. [Електронний ресурс].– Режим доступу до ресурсу: http://dspace.puet.edu.ua/handle/123456789/11386
- 4. Рогоза М.Є. Системи підтримки прийняття рішень: навч. пос. / М.Є. Рогоза, О.О. Ємець, Є.М. Ємець. Полтава: РВВ ПУЕТ, 2013. 328 с.
- Шиян А. А. Економічна кібернетика: вступ до моделювання соціальних і економічних систем: Навчальний посібник: МОН України / А. А. Шиян. – Львів : «Магнолія 2006», 2007. – 228 с.
- Н. В. Караєва, С. В. Войтко, Л. В. Сорокіна. Ризик-менеджмент сталого розвитку енергетики: інформаційна підтримка прийняття рішень : навчальний посібник / Н. В. Караєва, С. В. Войтко, Л. В. Сорокіна. — К. : Альфа Реклама, 2013. — 308 с.
- Балджи М.Д. Економічний ризик та методи його вимірювання. Навчальний посібник. – Харьків: Промарт, 2015. – 300 с.

- Економічний ризик: методи оцінки та управління [Текст] : навч. посібник / [Т. А. Васильєва, С. В. Лєонов, Я. М. Кривич та ін.]; під заг. ред. д-ра екон. наук, проф. Т. А. Васильєвої, канд. екон. Наук Я. М. Кривич. – Суми : ДВНЗ "УАБС НБУ", 2015. – 208 с.
- 9. Кучеренко В.Р., Карпов В.А., Карпов А.В. Економічний ризик та методи його вимірювання: Навчальний посібник. Одеса, 2011.

ДОДАТОК А

Main.py

import sys import json

from PyQt5 import QtCore, QtGui, QtWidgets from PyQt5.QtWidgets import QMessageBox from PyQt5.QtWidgets import QSizePolicy

from quest import Ui_MainWindow from startWindow import Ui_StartWindow from finalWindow import Ui_FinalWindow

class MyWindow(QtWidgets.QMainWindow):

def __init__(self, perent=None): QtWidgets.QWidget.__init__(self, perent) self.ui = Ui_MainWindow() self.startWindow() self.__numberQuest = 0 self.currentQuestion = [] self.correctAnswer = "" self.__nextQuest = True self.questionLoaded() self.ui.setupUi(self) self.newQuestion()

self.ui.answer_Button1.clicked.connect(self.clickedButton)
self.ui.answer_Button2.clicked.connect(self.clickedButton)
self.ui.answer_Button3.clicked.connect(self.clickedButton)

self.ui.answer_Button4.clicked.connect(self.clickedButton)
self.ui.next_Button.clicked.connect(self.newQuestion)

```
def questionLoaded(self):
```

with open('quest.json', 'r', encoding='utf8') as file: self.jsonQuestion = json.load(file)

def newQuestion(self):

if not self.__nextQuest: return

if self.__numberQuest == len(self.jsonQuestion):
 # self.ui.MainWindow.hide()
 self.finishWindow()
 return

self.__nextQuest = False
self.__clearStyleButton()

```
self.currentQuestion = self.jsonQuestion[self.__numberQuest]
buttonOne = self.currentQuestion["answer"][0]
buttonTwo = self.currentQuestion["answer"][1]
buttonThree = self.currentQuestion["answer"][2]
buttonFour = self.currentQuestion["answer"][3]
self.correctAnswer = self.currentQuestion["correct_answer"]
```

```
self.ui.numberQuestion.setText(f"Питання № {self.__numberQuest+1}")
self.ui.questionEdit.setPlainText(self.currentQuestion["question"])
self.ui.answer_Button1.setText(buttonOne)
self.ui.answer_Button2.setText(buttonTwo)
```

self.ui.answer_Button3.setText(buttonThree)
self.ui.answer_Button4.setText(buttonFour)

self.__numberQuest += 1

def clickedButton(self):

button = self.sender()

self.__clearStyleButton()

```
if button.text() == self.correctAnswer:
```

button.setStyleSheet("background-color: green;")

self.__nextQuest = True

else:

self.__nextQuest = False button.setStyleSheet("background-color: red;") QMessageBox.information(self, 'Увага', 'Відповідь невірна')

def __clearStyleButton(self):
 self.ui.answer_Button1.setStyleSheet("")
 self.ui.answer_Button2.setStyleSheet("")
 self.ui.answer_Button3.setStyleSheet("")
 self.ui.answer_Button4.setStyleSheet("")

def finishWindow(self):

finishWin = Ui_FinalWindow()
finishWin.setupUi(finishWin)
choise = finishWin.exec()

if choise:

```
self.ui.MainWindow.setVisible(True)
self.__numberQuest = 0
self.newQuestion()
else:
return
```

```
def startWindow(self):
    startWin = Ui_StartWindow()
    startWin.setupUi(startWin)
    startWin.exec()
```

```
if __name__=='__main__':
    app = QtWidgets.QApplication(sys.argv)
    myapp = MyWindow()
    myapp.show()
    app.exec_()
    sys.exit()
```

Quest.py

-*- coding: utf-8 -*-

Form implementation generated from reading ui file '.\ui\Дипломная.ui'

#

Created by: PyQt5 UI code generator 5.15.9

#

WARNING: Any manual changes made to this file will be lost when pyuic5 is

run again. Do not edit this file unless you know what you are doing.

from PyQt5 import QtCore, QtGui, QtWidgets

class Ui_MainWindow(object):

def setupUi(self, MainWindow):

MainWindow.setObjectName("MainWindow")

MainWindow.resize(434, 454)

MainWindow.setMinimumSize(QtCore.QSize(375, 0))

MainWindow.setStyleSheet("QMainWindow#MainWindow {\n"

"background-color: qlineargradient(spread:pad, x1:0.478, y1:0.00568182, x2:0.5, y2:1, stop:0 rgba(0, 0, 0, 255), stop:1 rgba(255, 255, 255, 255));\n"

self.MainWindow = MainWindow

self.centralwidget = QtWidgets.QWidget(MainWindow)

self.centralwidget.setStyleSheet("QWidget#centralwidget{\n"

"background-color: qlineargradient(spread:pad, x1:0.478, y1:0.00568182, x2:0.5, y2:1, stop:0 rgba(213, 237, 255, 1), stop:1 rgba(137, 183, 217, 1));\n"

self.centralwidget.setObjectName("centralwidget")

self.verticalLayout_2 = QtWidgets.QVBoxLayout(self.centralwidget)

self.verticalLayout_2.setObjectName("verticalLayout_2")

self.numberQuestion = QtWidgets.QLabel(self.centralwidget)

font = QtGui.QFont()

font.setFamily("Times New Roman")

font.setPointSize(18)

font.setBold(False)

font.setWeight(50)

self.numberQuestion.setFont(font)

self.numberQuestion.setAlignment(QtCore.Qt.AlignCenter)
self.numberQuestion.setWordWrap(False)
self.numberQuestion.setObjectName("numberQuestion")
self.verticalLayout_2.addWidget(self.numberQuestion)
self.questionEdit = QtWidgets.QTextEdit(self.centralwidget)
self.questionEdit.setStyleSheet("")
self.questionEdit.setReadOnly(True)
self.questionEdit.setObjectName("questionEdit")
self.verticalLayout_2.addWidget(self.questionEdit)
self.verticalLayout_2.addWidget(self.questionEdit)
self.verticalLayout = QtWidgets.QVBoxLayout()
self.verticalLayout = QtWidgets.QPushButton(self.centralwidget)
sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Minimum,
QtWidgets.QSizePolicy.MinimumExpanding)
sizePolicy.setHorizontalStretch(0)

```
sizePolicy.setVerticalStretch(0)
```

```
sizePolicy.setHeightForWidth(self.answer_Button1.sizePolicy().hasHeightFor
Width())
```

```
self.answer_Button1.setSizePolicy(sizePolicy)
```

 $self.answer_Button1.setStyleSheet("QPushButton\{\n"$

```
"border-radius: 5px;\n"
```

```
"background-color: white;\n"
```

 $" \} n "$

```
"\n"
```

```
"QPushButton:hover {\n"
```

```
" background-color: rgba(114, 85, 85, 0.33);\n"
```

"}")

```
self.answer_Button1.setCheckable(False)
self.answer_Button1.setObjectName("answer_Button1")
```

```
self.verticalLayout.addWidget(self.answer_Button1)
self.answer_Button2 = QtWidgets.QPushButton(self.centralwidget)
sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Minimum,
QtWidgets.QSizePolicy.MinimumExpanding)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
```

```
sizePolicy.setHeightForWidth(self.answer_Button2.sizePolicy().hasHeightFor
Width())
```

```
self.answer_Button2.setSizePolicy(sizePolicy)
    self.answer_Button2.setTabletTracking(False)
    self.answer_Button2.setStyleSheet("QPushButton{\n"
"border-radius: 5px;\n"
"background-color: white;\n"
"}\n"
"\n"
"OPushButton:hover {\n"
   background-color: rgba(114, 85, 85, 0.33);\n"
"}")
    self.answer_Button2.setAutoRepeat(False)
    self.answer_Button2.setAutoExclusive(False)
    self.answer_Button2.setAutoDefault(False)
    self.answer_Button2.setDefault(False)
    self.answer_Button2.setFlat(False)
    self.answer_Button2.setObjectName("answer_Button2")
    self.verticalLayout.addWidget(self.answer_Button2)
    self.answer_Button3 = QtWidgets.QPushButton(self.centralwidget)
    sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Minimum,
QtWidgets.QSizePolicy.MinimumExpanding)
    sizePolicy.setHorizontalStretch(0)
```

```
sizePolicy.setVerticalStretch(0)
```

sizePolicy.setHeightForWidth(self.answer_Button3.sizePolicy().hasHeightFor
Width())

```
self.answer_Button3.setSizePolicy(sizePolicy)
    self.answer_Button3.setStyleSheet("QPushButton{\n"
"border-radius: 5px;\n"
"background-color: white;\n"
"}\n"
"\n"
"OPushButton:hover {\n"
  background-color: rgba(114, 85, 85, 0.33);\n"
"
"}")
    self.answer_Button3.setObjectName("answer_Button3")
    self.verticalLayout.addWidget(self.answer_Button3)
    self.answer_Button4 = QtWidgets.QPushButton(self.centralwidget)
    sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Minimum,
QtWidgets.QSizePolicy.MinimumExpanding)
    sizePolicy.setHorizontalStretch(0)
    sizePolicy.setVerticalStretch(0)
```

```
sizePolicy.setHeightForWidth(self.answer_Button4.sizePolicy().hasHeightFor
Width())
```

```
self.answer_Button4.setSizePolicy(sizePolicy)
```

```
self.answer\_Button4.setStyleSheet("QPushButton{\n"}
```

```
"border-radius: 5px;\n"
```

```
"background-color: white;\n"
```

```
" \} n"
```

```
" n"
```

```
"QPushButton:hover {\n"
```

```
"
   background-color: rgba(114, 85, 85, 0.33);\n"
"}")
    self.answer_Button4.setObjectName("answer_Button4")
    self.verticalLayout.addWidget(self.answer_Button4)
    self.verticalLayout_2.addLayout(self.verticalLayout)
    self.horizontalLayout = QtWidgets.QHBoxLayout()
    self.horizontalLayout.setObjectName("horizontalLayout")
    spacerItem
                                   QtWidgets.QSpacerItem(40,
                                                                        20,
QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Minimum)
    self.horizontalLayout.addItem(spacerItem)
    self.next_Button = QtWidgets.QPushButton(self.centralwidget)
    self.next_Button.setObjectName("next_Button")
    self.horizontalLayout.addWidget(self.next_Button)
    self.verticalLayout_2.addLayout(self.horizontalLayout)
    MainWindow.setCentralWidget(self.centralwidget)
```

self.retranslateUi(MainWindow) QtCore.QMetaObject.connectSlotsByName(MainWindow)

def retranslateUi(self, MainWindow):

_translate = QtCore.QCoreApplication.translate

MainWindow.setWindowTitle(_translate("MainWindow", "TPEHAЖЕР «Кількістні шкали»"))

self.numberQuestion.setText(_translate("MainWindow", "Питання № ")) self.answer_Button1.setText(_translate("MainWindow", "Відповідь 1")) self.answer_Button2.setText(_translate("MainWindow", "Відповідь 2")) self.answer_Button3.setText(_translate("MainWindow", "Відповідь 3")) self.answer_Button4.setText(_translate("MainWindow", "Відповідь 4")) self.next_Button.setText(_translate("MainWindow", "Далі"))