

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
(підпис)

« ____ » _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему
**РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЕСКТОПНОГО
НАВЧАЛЬНОГО ЗАСТОСУНКУ З ТЕМИ «РЕКУРСИВНІ АЛГОРИТМИ»
ДИСТАНЦІЙНОГО НАВЧАЛЬНОГО КУРСУ «АНАЛІЗ АЛГОРИТМІВ»**

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

Виконавець роботи Тарасюк Назар Анатолійович
_____ « ____ » _____ 2024 р.
(підпис)

Науковий керівник к.пед.н., доцент, Кошова Оксана Петрівна
_____ « ____ » _____ 2024 р.
(підпис)

Рецензент

ПОЛТАВА 2024 р.

ЗАТВЕРДЖУЮЗавідувач кафедри _____ Олена ОЛЬХОВСЬКА
(підпис)

« ____ » _____ 202__ р.

**ЗАВДАННЯ І КАЛЕНДАРНИЙ ГРАФІК
ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

на тему «Розробка програмного забезпечення десктопного навчального застосунку з теми «Рекурсивні алгоритми» дистанційного навчального курсу «Аналіз алгоритмів»»

зі спеціальності 122 Комп'ютерні науки
освітня програма «Комп'ютерні науки»
ступеня магістр

Прізвище, ім'я, по батькові Тарасюк Назар Анатолійович

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

Термін подання студентом роботи « ____ » _____ 2023 р.

Вихідні дані до кваліфікаційної роботи: статті, наукові роботи та документації з теми розробки навчальних застосунків та сервісів для тестування.

Зміст пояснювальної записки (перелік питань, які потрібно розробити)

ВСТУП**РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ****РОЗДІЛ 2. ОГЛЯД СИСТЕМ АНАЛОГІЧНОГО ПРИЗНАЧЕННЯ****2.1. Навчальний додаток «Quizlet»****2.2. Навчальні тренажери для дистанційних курсів****2.3. Навчальний додаток «SoloLearn»****2.4. Навчальний додаток «Enki»****РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА****3.1. Опис проектних рішень, інструментів та підходів до розробки для десктопного застосунку.****3.2. Опис методології створення програмного забезпечення****РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА****4.1. Опис та побудова діаграми класів та діаграми взаємодії роботи з системою.****4.2. Інструкція з використання десктопного застосунку****ВИСНОВКИ****СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ****ДОДАТОК А**

Перелік графічного матеріалу: 1 аркуш блок-схем, 32 ілюстрації.

Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Постановка задачі	Кошова О.П.		
Інформаційний огляд	Кошова О.П.		
Теоретична частина	Кошова О.П.		
Практична частина	Кошова О.П.		

Календарний графік виконання кваліфікаційної роботи

Зміст роботи	Термін виконання	Фактичне виконання
Вступ		
Вивчення методичних рекомендацій та стандартів та звіт керівнику		
Постановка задачі		
Інформаційний огляд джерел бібліотек та інтернету		
Теоретична частина		
Практична частина		
Закінчення оформлення		
Доповідь студента на кафедрі		
Доробка (за необхідністю), рецензування		

Дата видачі завдання «__» _____ 2023 р.

Здобувач вищої освіти _____ Тарасюк Назар Анатолійович
(підпис)Науковий керівник _____ к.пед.н., доц. Кошова О.П.
(підпис) (науковий ступінь, вчене звання, ініціали та прізвище)

Результати захисту кваліфікаційної роботи

Кваліфікаційна робота оцінена на _____
(балів, оцінка за національною шкалою, оцінка за ECTS)

Протокол засідання ЕК № _____ від «__» _____ 2024 р.

Секретар ЕК _____
(підпис) (ініціали та прізвище)

Затверджую

Зав. кафедрою _____
к.ф.-м.н. Олена ОЛЬХОВСЬКА
« ____ » _____ 2023 р.

Погоджено

Науковий керівник _____
к.пед.н., Оксана КОШОВА
« ____ » _____ 2023 р.

План

дипломного проекту з фаху
спеціальності 122 Комп'ютерні науки
освітня програма 122 Комп'ютерні науки
на тему «Розробка програмного забезпечення десктопного навчального застосунку з теми «Рекурсивні алгоритми» дистанційного навчального курсу «Аналіз алгоритмів»»
Прізвище, ім'я, по батькові Тарасюк Назар Анатолійович

ВСТУП**РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ****РОЗДІЛ 2. ОГЛЯД СИСТЕМ АНАЛОГІЧНОГО ПРИЗНАЧЕННЯ****2.1. Навчальний додаток «Quizlet»****2.2. Навчальні тренажери для дистанційних курсів****2.3. Навчальний додаток «SoloLearn»****2.4. Навчальний додаток «Enki»****РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА****3.1. Опис проектних рішень, інструментів та підходів до розробки для десктопного застосунку.****3.2. Опис методології створення програмного забезпечення****РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА****4.1. Опис та побудова діаграми класів та діаграми взаємодії роботи з системою.****4.2. Інструкція з використання десктопного застосунку****ВИСНОВКИ****СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ****ДОДАТОК А**

Здобувач вищої освіти _____ Н.А. Тарасюк

(підпис)

« ____ » _____ 2023 р.

РЕФЕРАТ

Записка: 53 сторінки, 32 рисунки, 1 додаток, 7 літературних джерел.

Мета дипломної роботи розробка десктопного навчального застосунку для тестування на тему «Рекурсивні алгоритми» навчального дистанційного курсу «Аналіз алгоритмів».

Об'єкт розробки - десктопний навчальний застосунок.

Методи дослідження – використання сучасних підходів для написання якісних додатків. Графічний інструмент Figma для розроблення інтерфейсу та мова програмування C#.

Проведено аналіз тренажерів та застосунків аналогічного призначення. Визначено проектні рішення для розробки застосунку, вибрані інструменти та методики для розробки застосунку у сфері дистанційної освіти. Обґрунтовано вибір та описано підходи до розробки програмного забезпечення. Складено діаграму прецедентів, що демонструє процес взаємодії користувачів із системою. Розроблено інтерфейс додатку за допомогою мови програмування C# та підготовлено інструкцію з використання.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ.....	9
РОЗДІЛ 2. ОГЛЯД СИСТЕМ АНАЛОГІЧНОГО ПРИЗНАЧЕННЯ.....	12
2.1. Навчальний додаток «Quizlet»	12
2.2. Навчальні тренажери для дистанційних курсів.....	13
2.3. Навчальний додаток «SoloLearn».....	15
2.4. Навчальний додаток «Enki».....	17
РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА	19
3.1. Опис проектних рішень, інструментів та підходів до розробки для десктопного застосунку.....	19
3.2. Опис методології створення програмного забезпечення.....	27
РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА	30
4.1. Опис та побудова діаграми класів та діаграми взаємодії роботи з системою.	30
4.2. Інструкція з використання додатка.....	39
ВИСНОВКИ.....	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	46
ДОДАТОК А.....	47

ВСТУП

Актуальність. У сучасному суспільстві під час швидкого технологічного прогресу нові технології та засоби організації освітнього процесу створюють сприятливі умови для удосконалення рівня опанування професійними навичками майбутніми фахівцями із комп'ютерних наук. Інтернет виступає обширною платформою для студентів, яка забезпечує доступ до безлічі інформації. В цьому контексті актуальною є адаптація освітньої системи до сучасних вимог, з метою підготовки фахівців, які зможуть успішно конкурувати в сучасному світі. Одним із можливих шляхів вирішення цього завдання є створення та використання в навчальному процесі мобільних навчальних додатків. Адже, мобільні додатки відкривають нові можливості враховуючи різноманітні ресурси мережі Інтернет. Такі додатки, і десктопні у тому числі, можуть ефективно використовуватися для навчання не лише студентів, але й професіоналів для підвищення їхньої професійної компетентності.

Основним плюсом мобільних навчальних додатків, і десктопних у тому числі, є здатність надавати освітні послуги незалежно від географічного розташування користувача, часто навіть без наявності Інтернету. Це надає студентам гнучкість у виборі оптимального для них формату навчання, дозволяючи адаптувати процес освіти до власного ритму життя та специфічних потреб. Додатково, мобільне навчання може виявитися економічно вигідним, оскільки дозволяє навчатися дистанційно. Отже, мобільні навчальні додатки мають ряд переваг, таких як: можливість користувачам здобувати знання дистанційно; доступність навчання для осіб з особливими потребами; можливість зекономити на витратах, таких як проживання та проїзд; використання сучасних технологій і методик для навчання; можливість поєднання роботи та навчання.

Разом з тим десктопні навчальні застосунки можуть включати більше функцій та інструментів, що сприяє розширенню можливостей навчання. Адже більший екран комп'ютера дозволяє створювати більш складні та інтерактивні візуальні матеріали для поліпшення процесу засвоєння. За допомогою десктопних інструментів, розробка та редагування навчального контенту може бути ефективніше. При цьому десктопні застосунки можуть бути оптимізовані для великих груп користувачів, що робить їх ідеальними для використання в навчальних закладах. Не менш важливим є і можливість використовувати високоякісні графіку та звук на десктопах поліпшує візуальний та звуковий ефект у навчанні. Крім того велика клавіатура та миша забезпечують зручне введення даних, що може бути важливим для завдань, які вимагають багато тексту або складних введень. Все вище перераховане і підтвержує актуальність нашої диспломної роботи.

Мета даної дипломної роботи полягає у розробці десктопного навчального застосунку на тему «Рекурсивні алгоритми».

Об'єктом є десктопний навчальний застосунок.

Предметом розробки є програмне забезпечення для десктопного навчального застосунку та тему «Рекурсивні алгоритми» навчального дистанційного курсу «Аналіз алгоритмів»

Застосунок розроблений за допомогою мови програмування C#; мови розмітки XAML; сервісу для розробки інтерфейсів та прототипування Figma; технології створення графічного інтерфейсу користувача WPF та середовища розробки Microsoft Visual Studio.

Перелік методів включає аналіз існуючих технологій та способів проектування навчальних додатків та застосунків.

Пояснювальна записка включає чотири основних розділи: формулювання завдань, аналіз існуючих аналогів, теоретичні аспекти та практична реалізація. Структура роботи розрахована на логічне викладення матеріалу та повне розкриття теми дослідження.

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

Мета додатку

Сучасний світ технологій вимагає все більш ефективних методів навчання, особливо в сфері програмування та комп'ютерних наук. Тому навчальні додатки повинні відповідати певним вимогам для задоволення усіх потреб користувачів:

По перше це оцінка рівня розуміння студентами вивченої теми. Адже навчальний застосунок дозволяє користувачам оцінити свій рівень засвоєння конкретної теми або матеріалу шляхом вирішення тестових завдань. Це допомагає їм визначити, наскільки вони розуміють предметну область.

По друге це виявлення прогалин у знаннях. Навчальний застосунок для тестування допомагає ідентифікувати конкретні аспекти теми, в яких користувачі мають слабкості або прогалини в знаннях. Це надає можливість зосередити увагу на конкретних аспектах, які потребують додаткового вивчення.

При цьому тестові завдання з декількома варіантами відповідей заохочують користувачів аналізувати і обирати правильні відповіді, що сприяє глибшому розумінню матеріалу та розвитку критичного мислення.

Не менш важливою функцією використання навчальних застосунків є підготовка здобувачів вищої освіти до екзаменаційного (підсумкового) оцінювання. Тому, що навчальні застосунки для тестування можуть бути корисними для підготовки до іспитів, тестів чи оцінювань, де необхідно відповідати на питання з вибором правильних відповідей.

Крім того, за допомогою таких застосунків можна стежити за власним навчальним прогресом, фіксуючи результати тестувань і спостерігаючи за їхніми змінами з часом. Отже, основною метою навчального застосунку для тестування є поліпшення якості навчання, забезпечення кращого розуміння матеріалу та підготовка до оцінювання.

Основні функції додатку

1. Практичні вправи

У навчальному процесі, крім осмислення теоретичного матеріалу, важливо вміти застосовувати ці знання на практиці. З цією метою наш десктопний застосунок пропонує низку практичних завдань у формі тестів. Ці завдання були створені таким чином, щоб користувач міг перевірити своє розуміння рекурсивних алгоритмів сортування та виявити можливі пробіли у своїх знаннях.

Однією з ключових особливостей цих тестів є те, що до кожного запитання пропонується чотири варіанти відповідей. Користувачу необхідно обрати один із них, виходячи зі своєї логіки та знань на дану тему. Цей підхід спонукає до підвищення рівня опанування відповідною темою та сприяє розвитку інформаційно-аналітичних умінь.

Таким чином, практичні завдання стають невід'ємною частиною навчального процесу, допомагаючи користувачеві не лише закріпити свої знання, але й активно використовувати їх на практиці.

2. Статистика

Для мотивації та кращого розуміння своїх досягнень користувачам важливо мати можливість відстежувати свій прогрес. Саме тому після кожного проходження тестування користувачеві відображаються результати проходження тесту, які допомагають студентові оцінити свої результати як за теоретичною частиною, так і за практичними завданнями.

Сумуючи вищевказане, можна стверджувати, що статистика в додатку не лише відображає рівень досягнень користувача, але й служить важливим

інструментом самоаналізу та самовдосконалення, допомагаючи користувачу максимально ефективно використовувати свій час і ресурси.

Структура десктопного застосунку:

1. **Головна сторінка.** Доступ до основних розділів застосунку та інформація про розробників.

2. **Модуль тестів.** Перелік практичних завдань та теоретичних запитань, спрямованих на перевірку рівня опанування користувачами теми «Рекурсивні алгоритми» .

3. **Модуль статистики.** Візуалізація даних про успішність користувача в застосунку

З урахуванням зазначених вимог та особливостей, десктопний застосунок може стати ефективним інструментом для вивчення та поглиблення знань з аналізу алгоритмів та «Рекурсивних алгоритмів» зокрема.

РОЗДІЛ 2. ОГЛЯД СИСТЕМ АНАЛОГІЧНОГО ПРИЗНАЧЕННЯ

2.1. Навчальний додаток «Quizlet»

«Quizlet» – це інтерактивний онлайн-сервіс, призначений для створення освітніх карток, який допомагає користувачам освоювати новий матеріал, використовуючи віртуальні картки. Цей сервіс широко використовується студентами, вчителями та навчальними закладами по всьому світу для підготовки до тестів, контрольних робіт, іспитів або просто для закріплення навчального матеріалу (Рисунок 2.1).



Рисунок 2.1. Лого інтерактивного сервісу «Quizlet»

Quizlet - це популярна освітня платформа, яка надає інструменти для створення, вивчення та тестування навчальних матеріалів. Основні можливості Quizlet включають:

1. Створення навчальних наборів (карткових колод). Ви можете створювати набори карток з питаннями та відповідями, які допомагають вивчати новий матеріал.
2. Режими вивчення. Quizlet пропонує різні режими вивчення, включаючи повторення словникового запасу, тестування, виготовлення звуків тощо.
3. Спільництво користувачів. Платформа має активне спільноту користувачів, які створюють та діляться навчальними наборами.

4. Можливість імпорту та експорту. Ви можете імпортувати навчальний матеріал або експортувати його для обміну з іншими користувачами.

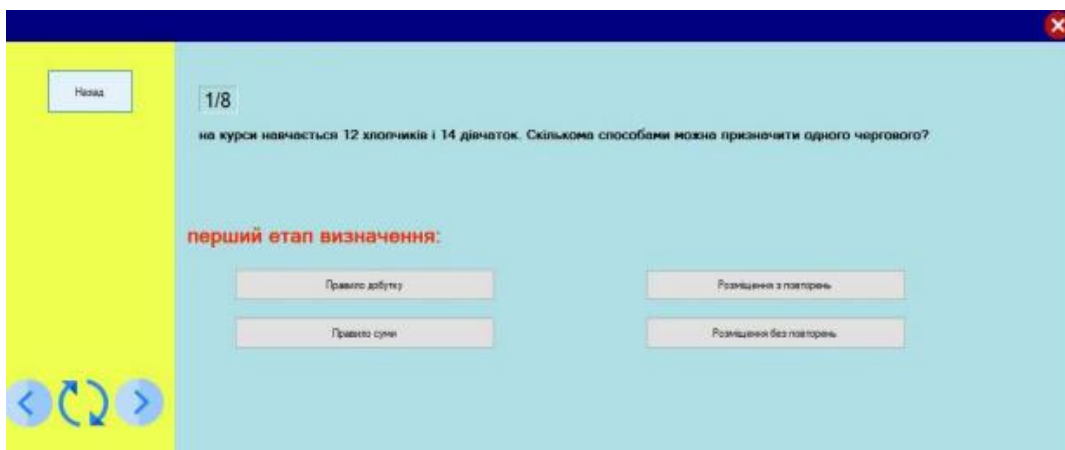
5. Додатки для мобільних пристроїв. Quizlet також доступний у вигляді додатків для смартфонів та планшетів, що дозволяє вчителям та учням вчити та перевіряти свої знання в будь-який час і в будь-якому місці.

Загалом, Quizlet є корисним інструментом для навчання та пам'яті, і він широко використовується як у шкільних класах, так і в самонавчанні.

2.2. Навчальні тренажери для дистанційних курсів

В рамках виконання кваліфікаційних робіт студентами спеціальності «Комп'ютерні науки» було створено чимало різних тренажерів.

Зокрема тренажер «Елементи комбінаторики. Формула класичної ймовірності», автор Сіам Карім Ашрафович. Фрагмент його реалізації представлено на рисунку далі. Але важливо зазначити, що в цьому тренажері акцент зроблено саме на задачах, що стосуються комбінаторики, а не її застосування під час розв'язування задач на пошук ймовірності.



Для створення тренажера застосовано об'єктно-орієнтовану мову програмування C# із застосуванням середовища візуальної розробки програм Microsoft Visual Studio 2019.

У контексті нашої роботи заслуговує на особливу увагу розроблений тренажер Фесенко Д.І. з дисципліни «Аналіз алгоритмів» на тему «Аналіз алгоритму сортування злиттям»

Для створення даного тренажеру було використано алгоритм сортування злиттям та хешування. Для розробки було застосовано об'єктно-орієнтовану мову програмування Java у вільному інтегрованому середовищі розробки Apache NetBeans.

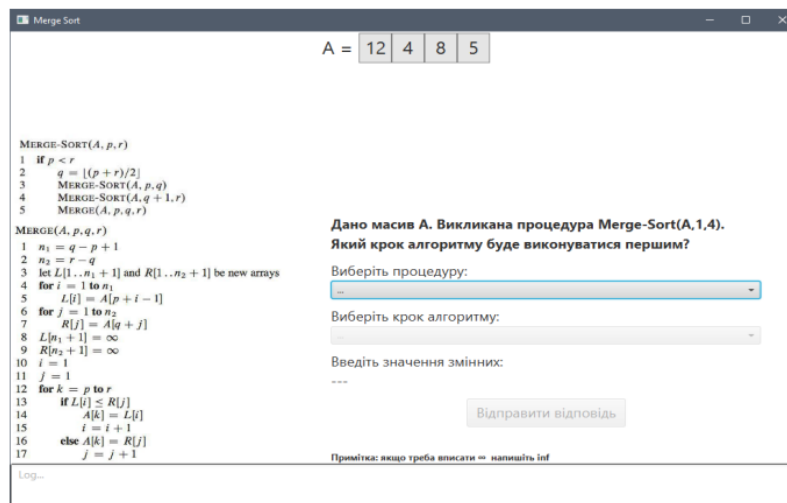


Рисунок 2.2. Фрагмент тренажеру «Аналіз алгоритму сортування вставками»

Програмні засоби для реалізації програми студента даного тренажера. JavaFX – набір інструментів для створення кросплатформенних графічних додатків на платформі Java. Особливості JavaFX: JavaFX насамперед поставляється з великим набором елементів графічного інтерфейсу, такі як: кнопки, таблиці, меню, діаграми, тощо. Що в свою чергу дає змогу зекономити досталь часу. JavaFX часто використовує стилі CSS, а також є можливість використовувати спеціальні параметри FXML з метою формування GUI, щоб не робити це в коді Java. Це полегшує швидке розташування графічного інтерфейсу користувача або зміна зовнішнього вигляду. JavaFX має готові до використання частини діаграми, тому не потрібно їх писати з самого нуля, коли потрібна базова діаграма. JavaFX додатково поставляється з підтримкою 3D графіки, яка може бути корисна при розробці ігор або схожих додатків. Програмні засоби для реалізації програми викладача. Java MessageDigest – клас

передбачає шифрувальну хеш-функцію, що може вичислювати дайджест повідомлення з бінарних даних. Коли ми отримуємо набір зашифрованих даних, то не маємо жодної гарантії, що він не був змінений. Дайджест повідомлення допомагає вирішити цю проблему. Для того, щоб встановити чи було змінено дані, відправник повинен розрахувати дайджест повідомлення з даних і відправити їх разом з даними. Одержувач після отримання зашифрованих даних і дайджест повідомлення, може перерахувати дайджест повідомлення з даних і перевірити, чи відповідає обчислений дайджест повідомлення дайджесту повідомлення, отриманому з даними. ²⁷ Якщо два дайджест повідомлення співпадають, то є велика ймовірність того, що результати були не змінені [7].

2.3. Навчальний додаток «SoloLearn»

SoloLearn - це популярна платформа для навчання програмуванню та розвитку інших навичок в області технологій. Основні риси і можливості SoloLearn включають:

1. Безкоштовний доступ - SoloLearn пропонує безкоштовний доступ до курсів з різних мов програмування, таких як Python, JavaScript, Java, C++, та інших, а також до курсів з веб-розробки, алгоритмів та інших суміжних тем.
2. Інтерактивне навчання - Платформа надає можливість навчатися шляхом розв'язування завдань та практичних вправ, що допомагає засвоювати матеріал практично.
3. Мобільні додатки - SoloLearn має мобільні додатки для iOS та Android, що дозволяє користувачам навчатися на ходу.
4. Робота в команді - Платформа має активну спільноту користувачів, де можна обговорювати питання, ділитися досвідом та навіть брати участь у викликах та змаганнях.
5. Підтвердження отриманих результатів - SoloLearn також надає можливість отримати сертифікати про успішне завершення курсів, які можуть

бути важливими для пошуку роботи або подальшого навчання.

6. Різнорівневі курси - Курси на SoloLearn доступні для різних рівнів навчання, від початківців до досвідчених розробників.

7. Виклики та змагання - Платформа пропонує різноманітні виклики та змагання, що допомагають користувачам вдосконалювати свої навички та порівнювати їх з іншими учасниками.

SoloLearn є корисним ресурсом для тих, хто бажає навчитися програмувати та розвивати свої навички в сфері інформаційних технологій (Рисунок 2.3).



Рисунок 2.3. Лого навчального додатку «SoloLearn»

Разом з тим SoloLearn також має свої недоліки. Ось деякі з них:

- Недостатньо високий рівень навчальних матеріалів. SoloLearn добре підходить для початківців та тих, хто шукає основні знання. Проте платформа може бути обмеженою для досвідчених розробників, які бажають вивчати більш складні аспекти програмування.
- Втрата актуальності матеріалів. Швидке розвиток технологій означає, що матеріали на SoloLearn можуть застарівати. Важливо самостійно перевіряти актуальність інформації та додатково досліджувати нові технології.
- Відсутність інтерактивних проєктів. Платформа фокусується на розв'язанні завдань і вправ, але вона може не надавати достатньо можливостей для роботи над реальними проєктами.
- Неможливість отримати ступінь. Ви не отримаєте формального

академічного ступеня або диплому, використовуючи цю платформу, що може бути важливим для деяких робочих місць або навчальних програм.

Загалом, SoloLearn - це корисний інструмент для початківців і тих, хто шукає основні знання у програмуванні, але він має свої обмеження і не підходить для всіх потреб і рівнів навчання.

2.4. Навчальний додаток «Enki»

«Enki» – це освітній додаток, розроблений для тих, хто бажає покращити свої навички у програмуванні, дизайні, основах даних та інших галузях інформаційних технологій. За допомогою коротких щоденних уроків та завдань, «Enki» допомагає користувачам поглибити свої знання та залишатися в курсі найсвіжіших трендів і технологій. (Рисунок 2.4)



Рисунок 2.4. Навчальний додаток «Enki»

Переваги навчального додатку «Enki»:

1. Короткі щоденні уроки. «Enki» пропонує короткі інтерактивні уроки, які можна виконувати щодня. Це робить навчання більш доступним і сприяє постійному зростанню навичок.
2. Різноманітність тем. Додаток пропонує уроки та виклики в різних сферах ІТ, включаючи програмування, дизайн і основи даних. Це дозволяє користувачам вибирати те, що їх цікавить найбільше.
3. Адаптивність: «Enki» враховує рівень навчання користувача і надає завдання, які відповідають його потребам та вмінням.
4. Мобільність додатку. Додаток доступний на мобільних пристроях (iOS та Android), що дозволяє навчатися в будь-якому місці та в будь-який час.

Разом з тим у цього навчального додатку є і недоліки, а саме:

1. Платність. Деякі розділи та функції в «Enki» вимагають підписки або плати, що може відлякувати користувачів.
2. Обмеженість в контенті. Хоча «Enki» пропонує багато корисних уроків, він може бути обмеженим в порівнянні з більшими освітніми платформами або курсами.
3. Не для глибокого поглибленого навчання: Додаток надає загальні знання та вправи, але не завжди підходить для глибокого вивчення конкретних тем або проектів.

Загалом, «Enki» - це корисний інструмент для швидкого та доступного навчання в галузі інформаційних технологій, але він може не підходити для всіх потреб і рівнів навчання, особливо для тих, хто шукає більше глибокого та розширеного навчання.

Отже, проаналізувавши існуючі системи аналогічного призначення сформулюємо проектні вимоги та інструменти для розробки власного навчального десктопного застосунку, про що буде описано більш докладно в наступних розділах пояснювальної записки.

РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Опис проектних рішень, інструментів та підходів до розробки для десктопного застосунку.

Для розробки тренажера було обрано:

- сервіс для розробки інтерфейсів та прототипування Figma;
- мову програмування C#;
- мову розмітки XAML;
- технологію створення графічного інтерфейсу користувача WPF.
- середовище розробки Microsoft Visual Studio.

Figma – це онлайн-інструмент для дизайну і прототипування інтерфейсів користувача. Figma дозволяє дизайнерам створювати векторні зображення, інтерактивні прототипи, макети веб-сторінок, мобільних додатків та інші проекти. (Рисунок 4.1). Оскільки Figma – це хмарний сервіс, він дозволяє користувачам працювати над своїми проектами з будь-якого пристрою, підключеного до Інтернету [2].



Рисунок 4.1 – Сервіс для розробки інтерфейсів «Figma»

У цьому редакторі можна створювати:

- векторні ілюстрації;
- іконки, меню, вікна, кнопки та інші елементи інтерфейсу;
- інтерактивні прототипи мобільних додатків та сайтів.

Можливості Figma:

Реалістичний дизайн: Figma дозволяє створювати дизайн інтерфейсу з великою деталізацією, включаючи векторні графічні об'єкти, тексти, фотографії та іконки.

Прототипування: Ви можете створювати інтерактивні прототипи, додавати переходи між сторінками, анімацію та інші ефекти, щоб детально відобразити функціональність вашого дизайну.

Спільна робота: Figma розроблено з урахуванням можливостей спільної роботи. Ви можете одночасно працювати над проектом з іншими користувачами, бачити зміни в реальному часі та залишати коментарі.

Версіонування: Figma автоматично зберігає історію змін проекту, дозволяючи вам легко переглядати і відновлювати попередні версії дизайну.

Векторні графіка і іконки: У Figma вбудовані інструменти для створення векторної графіки, а також можливість імпортувати власні іконки та ресурси.

Спеціальні інструменти для UI/UX дизайну: Figma має інструменти для швидкого створення макетів, мобільних екранів, веб-сторінок та інтерфейсів.

Збірка відгуків: Ви можете легко ділитися проектом з іншими користувачами, отримувати від них відгуки та коментарі, а також встановлювати права доступу.

Інтеграція з іншими інструментами: Figma підтримує інтеграцію з іншими популярними інструментами для розробки та дизайну, такими як Sketch, Adobe XD, Zeplin, та інші.

Робота з командами: Для великих проектів Figma пропонує можливості керування правами доступу та обміну ресурсами між командами.

Функціонал Figma:

- створення дизайну інтерфейсу з великою деталізацією, включаючи векторні графічні об'єкти, тексти, фотографії та іконки.

- створення інтерактивних прототипів, додавання переходів між сторінками, анімацію та інші ефекти, щоб детально відобразити функціональність вашого дизайну.
- можливість спільної роботи. Ви можете одночасно працювати над проектом з іншими користувачами, бачити зміни в реальному часі та залишати коментарі.
- можливість автоматично зберігати історію змін проекту, дозволяючи вам легко переглядати і відновлювати попередні версії дизайну.
- вбудовані інструменти для створення векторної графіки, а також можливість імпортувати власні іконки та ресурси.
- Спеціальні інструменти для UI/UX дизайну. Figma має інструменти для швидкого створення макетів, мобільних екранів, веб-сторінок та інтерфейсів.
- підтримка інтеграції з іншими популярними інструментами для розробки та дизайну, такими як Sketch, Adobe XD, Zeplin, та інші.
- можливість керування правами доступу та обміну ресурсами між командами [2].

C# – це об'єктно-орієнтована мова програмування, розроблена компанією Microsoft для розробки програм на платформі NET (Рисунок 4.2). Мова була розроблена у 2000 році Андерсом Хейлсбергом, Скотом Вілтанутом та Пітером Гольде під егідою Microsoft Research (при фірмі Microsoft) [3].

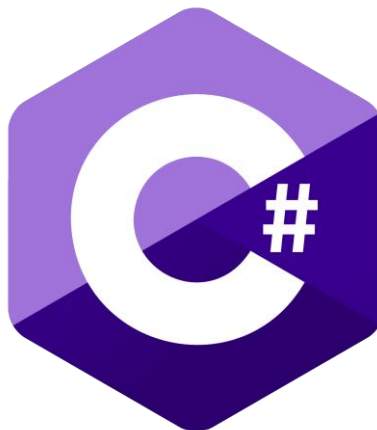


Рисунок 4.2 – Логотип мови програмування «C#»

C# має чимало бібліотек, що допомагають спростити процес створення застосунку й забезпечують повторне використання стандартного коду, застосунку, а також стандартизують архітектуру та стиль написання коду, що важливо для можливого розширення функціональності іншими розробниками.

Основні особливості мови C#:

- Об'єктно-орієнтоване програмування. C# включає в себе всі ключові принципи ООП, такі як успадкування, інкапсуляція та поліморфізм. Це сприяє створенню програм з чіткою структурою та послідовним дизайном, що полегшує розуміння і зміну коду.
- Статична типізація. C# - це мова з високим рівнем строгого контролю за типами, що допомагає виявити помилки на етапі компіляції, що спрощує виявлення та виправлення них.
- Автоматичне керування пам'яттю. Мова має вбудовану систему автоматичного управління пам'яттю, що спрощує процес розробки програм, оскільки розробники можуть уникнути проблем, пов'язаних з витоками пам'яті та іншими подібними проблемами.
- Підтримка делегатів та подій. Мова надає можливість використовувати делегати та події, що дозволяє розробникам створювати функції вищого порядку та ефективно реалізовувати механізми зворотного зв'язку в програмах.
- Використання інтерфейсів. Це дає можливість розробникам відокремлювати інтерфейси користувача від реалізації програмного забезпечення.
- Підтримка паралельного програмування. Можливість використовувати багатопотоковість і паралельні обчислення дозволяє створювати більш продуктивні програми.
- Підтримка Language Integrated Query (LINQ). LINQ спрощує виконання складних операцій над об'єктами швидко та легко.

– Платформона незалежність. C# розроблена для платформи .NET, що дозволяє створювати програми, які працюють на різних операційних системах, таких як Windows, Linux, macOS і інші [3].

XAML – мова розмітки для декларативного програмування додатків, розроблена Microsoft (Рисунок 4.3) [4].



Рисунок 4.3 – Логотип мови розмітки «XAML»

Використання в додатках, розроблених для платформи Microsoft .NET, зокрема для створення додатків Windows Presentation Foundation (WPF) і Universal Windows Platform (UWP), має мова опису інтерфейсу користувача на основі XML, відома як XAML. XML дозволяє розробникам візуалізувати інтерфейс користувача без необхідності кодування його вручну. Він надає зручний і декларативний спосіб створення ієрархії елементів та налаштування їх властивостей, таких як розмір, положення, кольори, шрифти, зображення, анімації і т. д.

XAML використовує формат XML для визначення елементів та їх атрибутів. Кожен елемент у XAML відповідає об'єкту, який є екземпляром певного типу. Простір імен типу (класу, перерахування тощо) визначається як простір імен, що фізично знаходиться у збірці (DLL) бібліотеки .NET Framework.

Зазвичай XAML інтегрується з кодом програми, написаним на C# або інших мовах програмування, які використовують .NET Framework або .NET Core.

XAML надає можливість створювати різноманітні інтерфейси користувача, включаючи як прості форми, так і складні макети з анімацією та змінними станами. Важливо відзначити, що він дозволяє відокремлювати розмітку від бізнес-логіки додатку, що сприяє зручній розробці, тестуванню та підтримці коду.

Крім того, XAML підтримує декілька мов програмування, таких як C#, VB.NET, F# та інші, що дає розробникам можливість використовувати мову, яка найкраще відповідає їхнім потребам [4].

WPF (Windows Presentation Foundation) - це технологія, яка використовується для створення інтерфейсу користувача в додатках для операційної системи Windows. Вона була випущена компанією Microsoft у 2006 році, разом з .NET Framework 3.0, і надає потужні засоби для створення сучасних та ефективних інтерфейсів для Windows додатків (Рисунок 4.4) [5].



Рисунок 4.4 – Логотип технології «Wpf»

WPF (Windows Presentation Foundation) - це рушій, відповідальний за створення, відображення та маніпулювання користувацькими інтерфейсами, документами, зображеннями, фільмами та мультимедійними даними в операційних системах Windows і їх більш пізніх версіях. WPF включає в себе

набір бібліотек, які надають всю необхідну функціональність для створення, запуску, виконання та управління клієнтськими програмами для операційних систем Windows. (Рисунок 4.5).

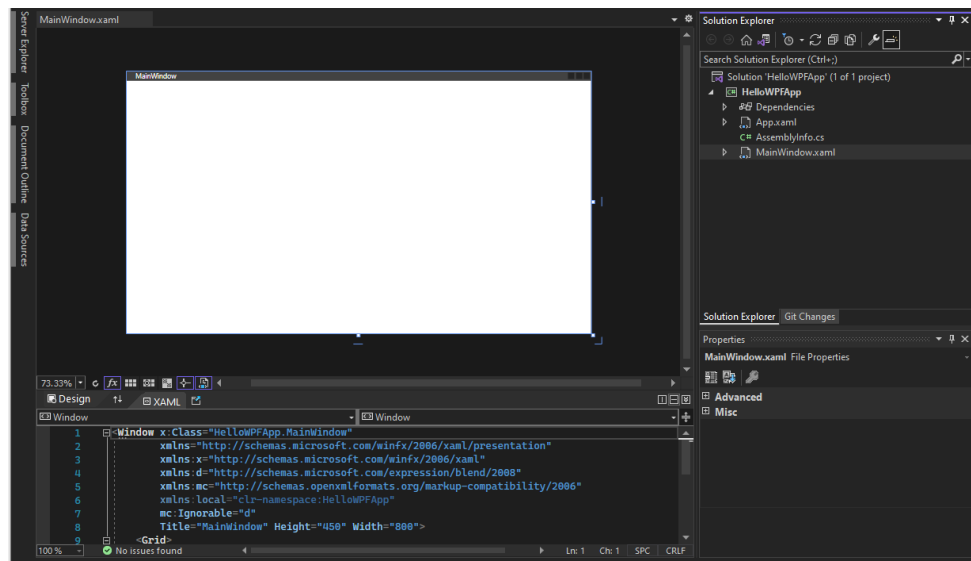


Рисунок 4.5 – Створення користувацького інтерфейсу за допомогою «WPF»

- WPF (Windows Presentation Foundation) використовує мову розмітки XAML (Extensible Application Markup Language) для візуального опису інтерфейсу користувача, що дає розробникам можливість створювати інтерфейс безпосередньо з коду.
- WPF надає розширену підтримку для анімації, 3D-графіки та можливість створювати додатки як для робочого столу, так і для Windows Store.
- Переваги WPF включають:
- Можливість створювати власні елементи управління інтерфейсом з нескінченними можливостями і гнучкістю.
- Масштабованість, що дозволяє створювати додатки для різних розмірів екранів і пристроїв.
- Можливість багаторазового використання коду та елементів управління.
- Високоєфективні елементи управління з потужними можливостями прив'язки даних, які спрощують роботу з даними.

- Можливість додавати анімацію та інші спеціальні ефекти до елементів управління та компонентів інтерфейсу користувача.
- Використання Direct2D для відображення 2D/3D графічних даних, що використовує графічний процесор системи у WPF [5].

Microsoft Visual Studio – інтегроване середовище розробки програмного забезпечення від фірми Microsoft. Дане середовище дозволяє створювати різноманітні програми, веб-сайти, мобільні додатки, ігри тощо (Рисунок 4.6).

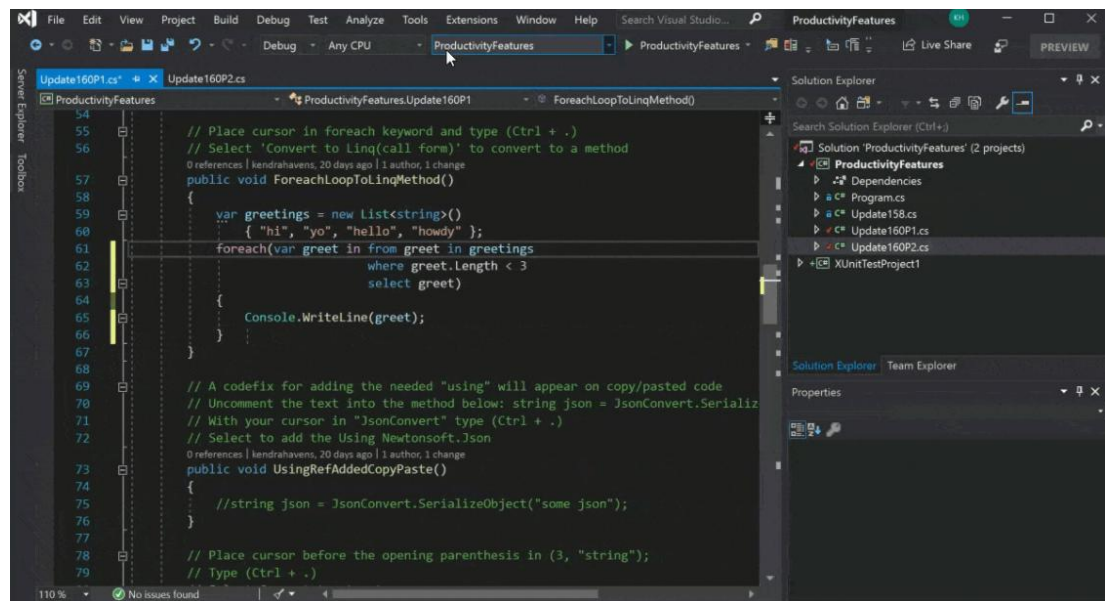


Рисунок 4.6 – Середовище розробки програмного забезпечення «Microsoft Visual Studio»

Visual Studio - це інтегроване середовище розробки (IDE), яке надає розробникам набір інструментів для написання коду, створення інтерфейсів користувача, відлагодження програм, збирання та розгортання додатків на різних платформах. Важливо відзначити, що Visual Studio підтримує різні мови програмування, такі як C++, C#, Visual Basic, F#, JavaScript та інші.

Можливості та переваги Visual Studio включають:

- Створення та розробка додатків для різних платформ.
- Наявність широкого набору інструментів розробки, включаючи емулятори, відладники та симулятори.
- Розширений редактор коду з IntelliSense, автозавершенням коду, підсвічуванням синтаксису та перевіркою коду.

- Підтримка популярних мов програмування, таких як C#, Visual Basic, F#, C++, HTML, JavaScript, TypeScript, Python.
- Інтеграція з пакетом SDK для Azure, що дозволяє завантажувати проекти в Azure та аналізувати їх продуктивність та ефективність за допомогою служби Application Insights.
- Підтримка розширень, які дають доступ до тисяч розширень у галереї Visual Studio.
- Інтеграція з системою контролю версій Git.
- Безкоштовна версія Visual Studio Community.
- Спрощене модульне встановлення та конфігурацію.

Visual Studio доступний в різних редакціях, включаючи безкоштовну Visual Studio Community, а також платні редакції Visual Studio Professional та Visual Studio Enterprise.

Загалом, Visual Studio є потужним інструментом для розробки програмного забезпечення, який надає розробникам широкі можливості для створення різноманітних додатків, ігор, а також для аналізу даних та інших завдань [6].

3.2. Опис методології створення програмного забезпечення

Для розробки мобільного додатку нами було обрано методологію Waterfall (каскадну модель). Каскадна модель є послідовною методологією розробки, де кожен наступний етап починається після завершення попереднього. (Рисунок 4.7).



Рисунок 4.7. Модель розробки за каскадною методологією

Каскадна модель ділить процес розробки на послідовні фази:

1. Затвердження вимог. Цей етап включає в себе аналіз потреб користувачів і визначення усіх функціональних і нефункціональних вимог до програмного продукту. Це важливий крок, оскільки невірно зрозумілі або пропущені вимоги можуть призвести до проблем пізніше в процесі розробки.

2. Проектування. На цьому етапі розробляється загальна архітектура програмного продукту. Визначаються ключові компоненти, їх взаємодія, інтерфейс користувача, технічні деталі, та інше. Можуть бути створені схеми, діаграми та інші документи для опису архітектури.

3. Розробка. На цьому етапі програмісти розробляють код програмного продукту відповідно до специфікацій, які були розроблені на попередніх етапах. Розробка може проводитися з використанням різних технологій та мов програмування.

4. Тестування. Після розробки коду проводиться тестування, включаючи функціональне тестування, тестування на відповідність вимогам, виявлення помилок та їх виправлення. Також можуть бути проведені тести на продуктивність та безпеку.

5. Впровадження та підтримка. Готовий продукт встановлюється на цільових пристроях або серверах. Цей етап також включає в себе підготовку до

запуску, документацію для користувачів та навчання їх використовувати програмний продукт. Після випуску продукту в експлуатацію може знадобитися підтримка, оновлення для виправлення помилок та впровадження нових функцій. Цей етап може продовжуватися протягом тривалого часу.

Основною перевагою каскадної моделі є її структурованість і простота. Вона відмінно підходить для невеликих проєктів або проєктів з чітко визначеними вимогами, які не змінюються протягом часу. Відсутність команди та необхідності швидко реагувати на зміни ринку робить цю модель відповідною для нашого проєкту.

Модель розробки за каскадною методологією (етапи можуть відрізнятися через специфіку проєкту, але суть одна у всіх) [*] .

РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

4.1. Опис та побудова діаграми класів та діаграми взаємодії роботи з системою.

Наведемо спочатку схему розробленого десктопного застосунку (рис.4.8.)

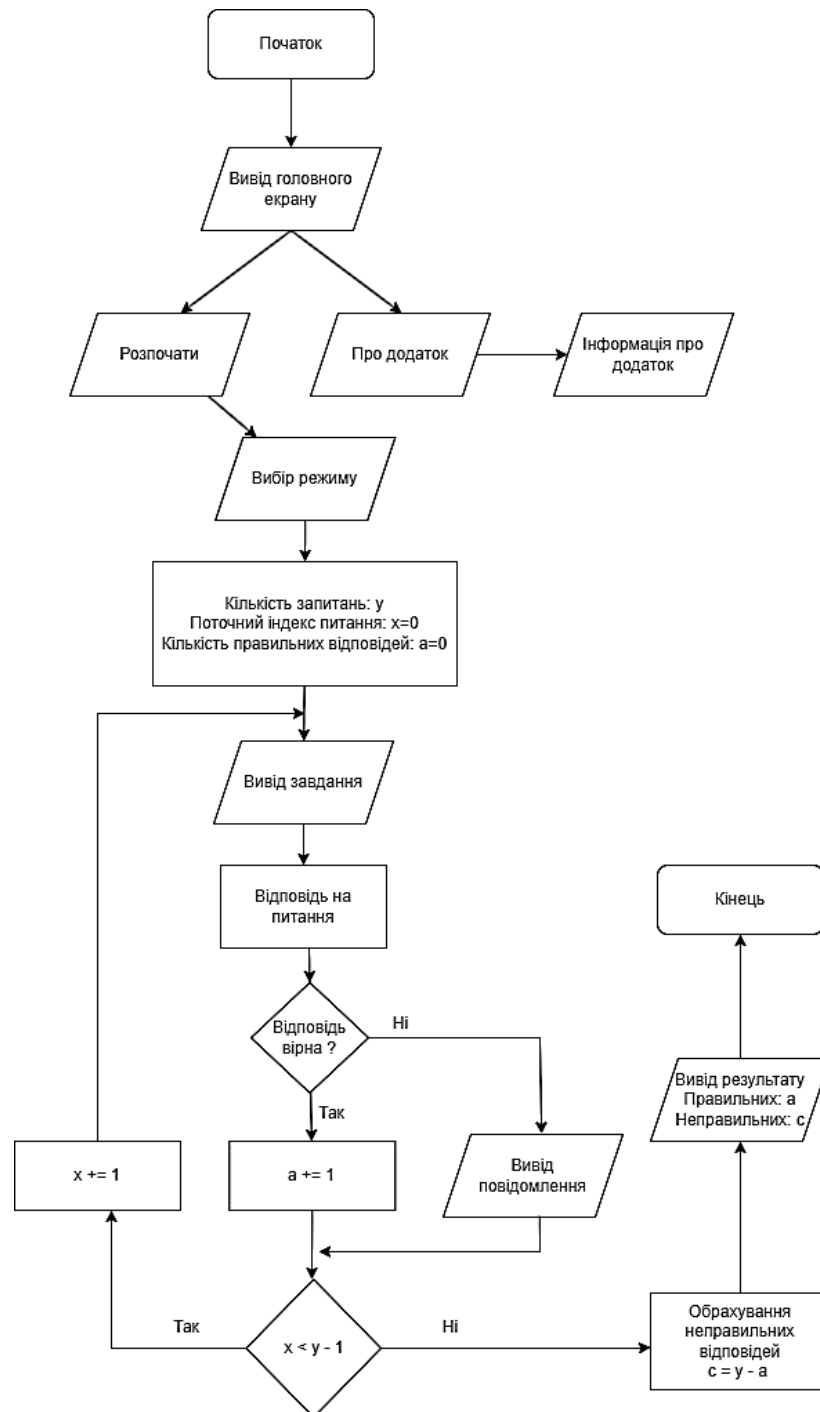


Рисунок 4.8. Блок схема застосунку

Відповідно до схеми нашого застосунку, побудуємо діаграму Взаємодії (Рисунок 4.9), демонструє ключові можливості та функціональність нашого застосунку. Ролі у цьому контексті є відображенням того, які дії учасники освітнього процесу можуть здійснювати за допомогою розробленого застосунку. Така діаграма допомагає краще зрозуміти та відобразити взаємодію між користувачами та додатком.

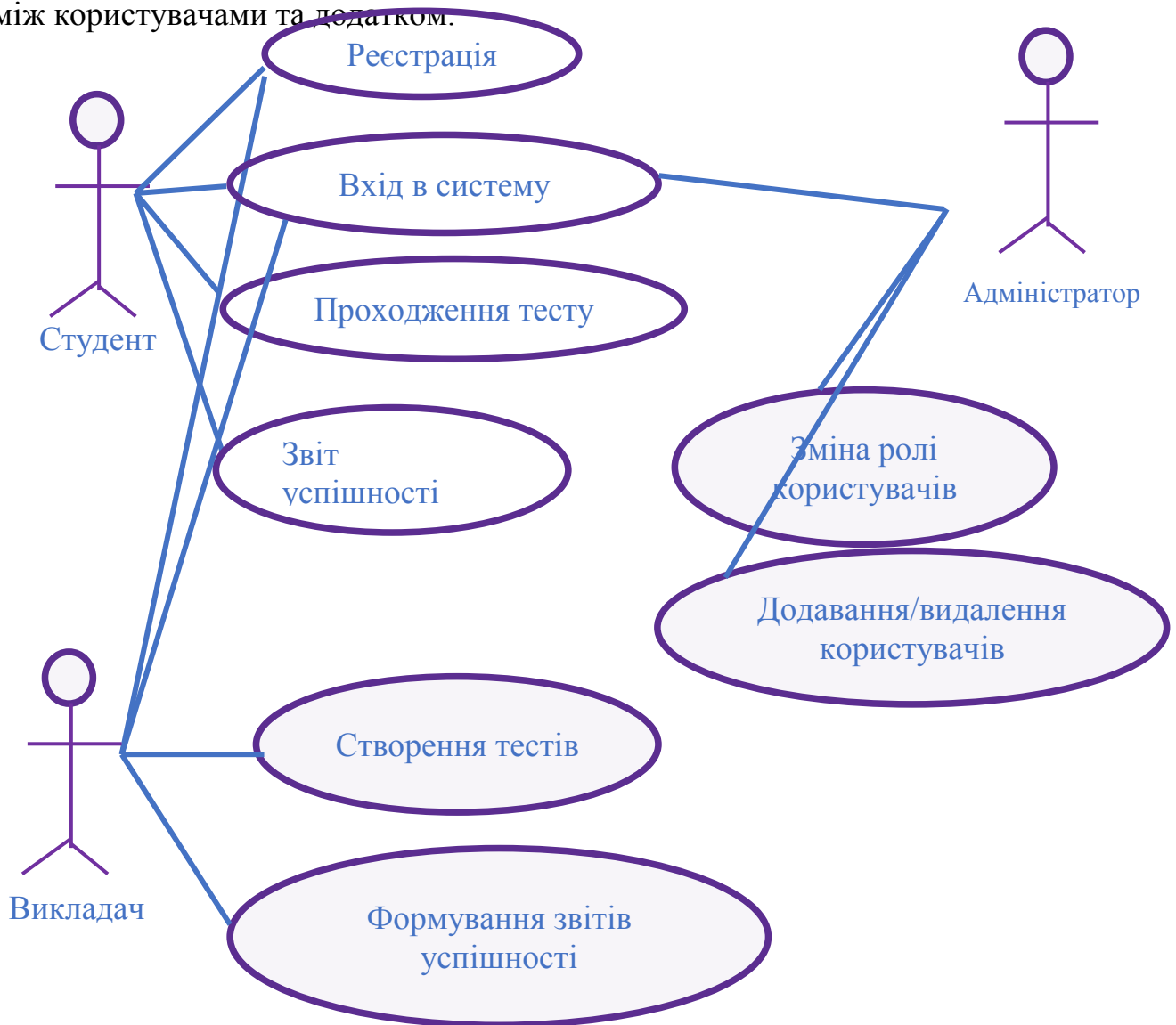


Рисунок 4.9. Діаграма взаємодії розробленого застосунку

Додаток створено на технології Windows Presentation Foundation (WPF) - аналог WinForms, система для побудови клієнтських застосунків Windows з візуально привабливими можливостями взаємодії з користувачем, графічна (презентаційна) підсистема у складі .NET Framework (починаючи з версії 3.0), що використовує мову XAML.



Рисунок 4.10.

Технологія WPF використовує багаторівневу архітектуру. На вершині ваш додаток взаємодіє з високорівневим набором служб, які повністю написані керованим кодом C#. Дійсна робота з трансляції об'єктів .NET у текстури і трикутники Direct3D відбувається "за лаштунками", з використанням низькорівневого некерованого компонента на ім'я `milcore.dll`. Бібліотеку `milcore.dll` реалізовано в некерованому коді тому, що вона потребує тісної інтеграції з Direct3D, і на додачу до того ж для неї надзвичайно важлива продуктивність.

На рисунку 4.11. показано рівні, на яких побудовано роботу програми WPF:

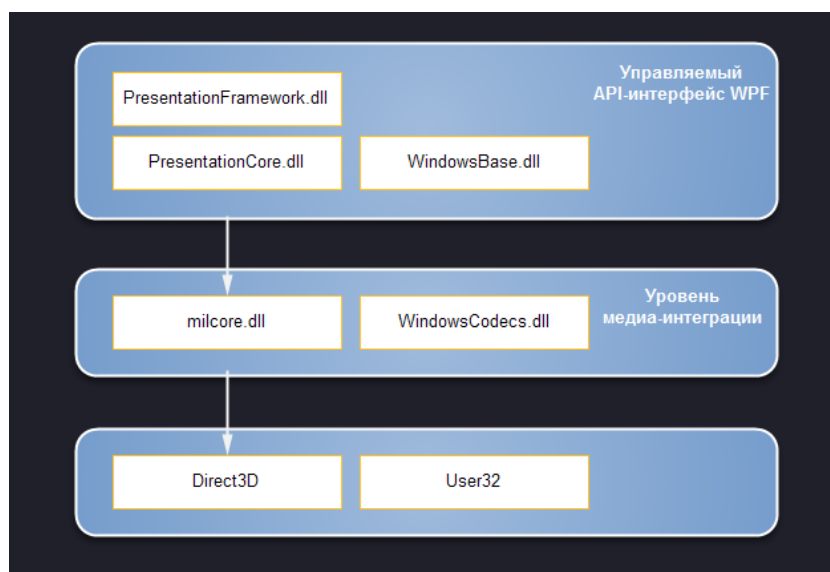


Рисунок 4.11. Рівні роботи програми WPF

Додаток складається з набору візуальних компонентів Windows(вікон), кожне з яких виконує своє завдання. Далі на діаграмі класів вікна застосунку:

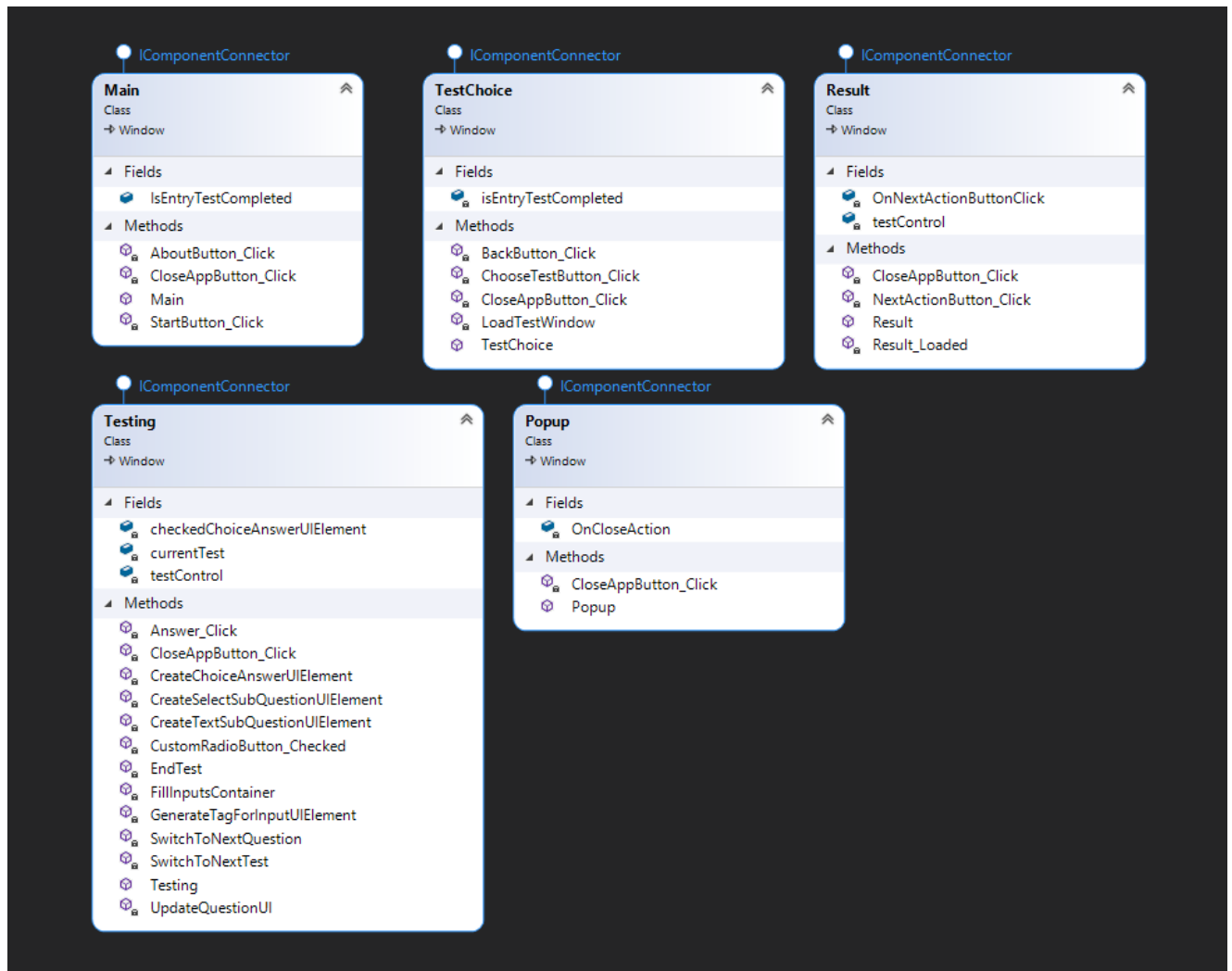


Рисунок 4.12. Діаграма класів розробленого десктопного застосунку

Відображено всі вікна, що використовуються.

Робота десктопного застосунку починається із ініціалізації основного вікна

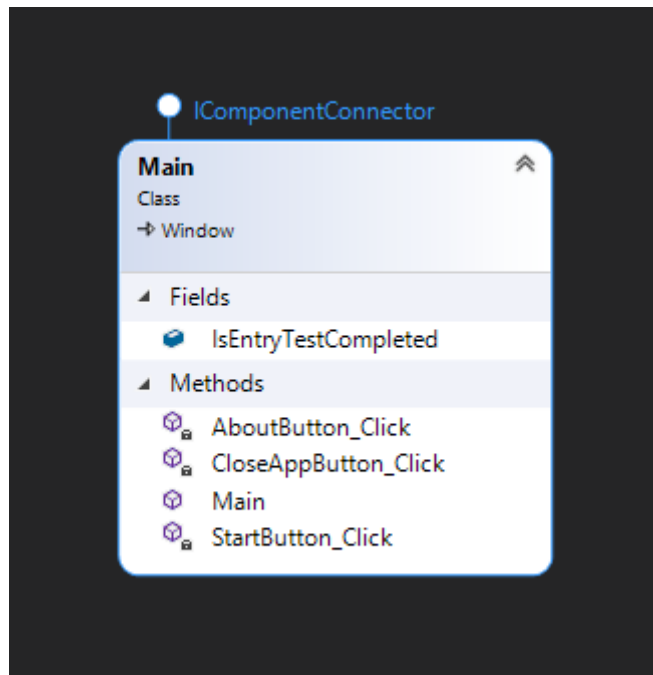


Рисунок 4.13. Ініціалізація основного вікна

Його основне завдання запропонувати користувачу на вибір :

- ознайомитися із інформацією про застосунок (Popup)

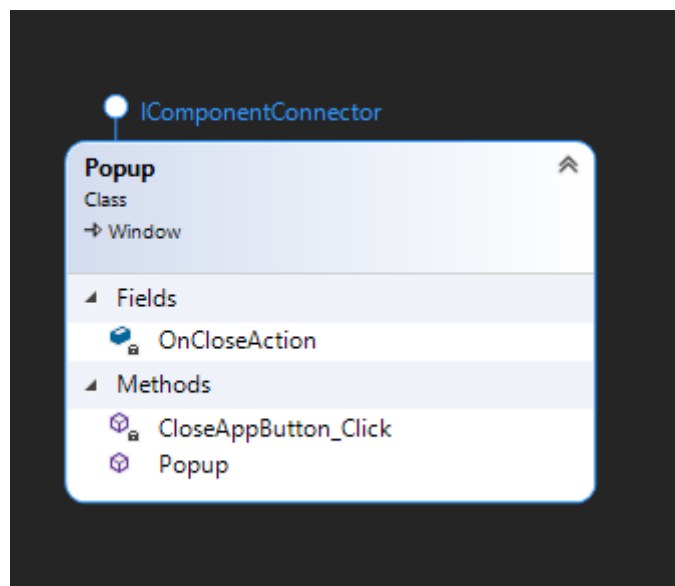


Рисунок 4.14. Інформація про застосунок

- перейти до вибору тесту:

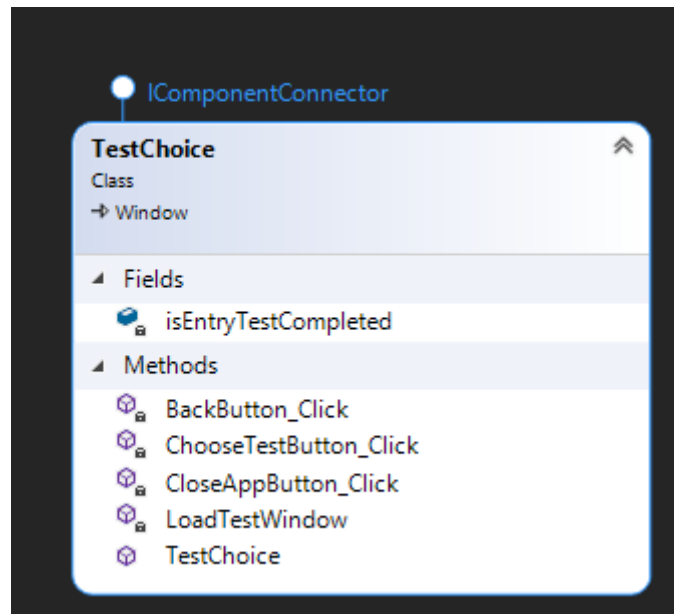


Рисунок 4.15. Вибір тесту

Після вибору тесту відбувається перехід у вікно тестування:

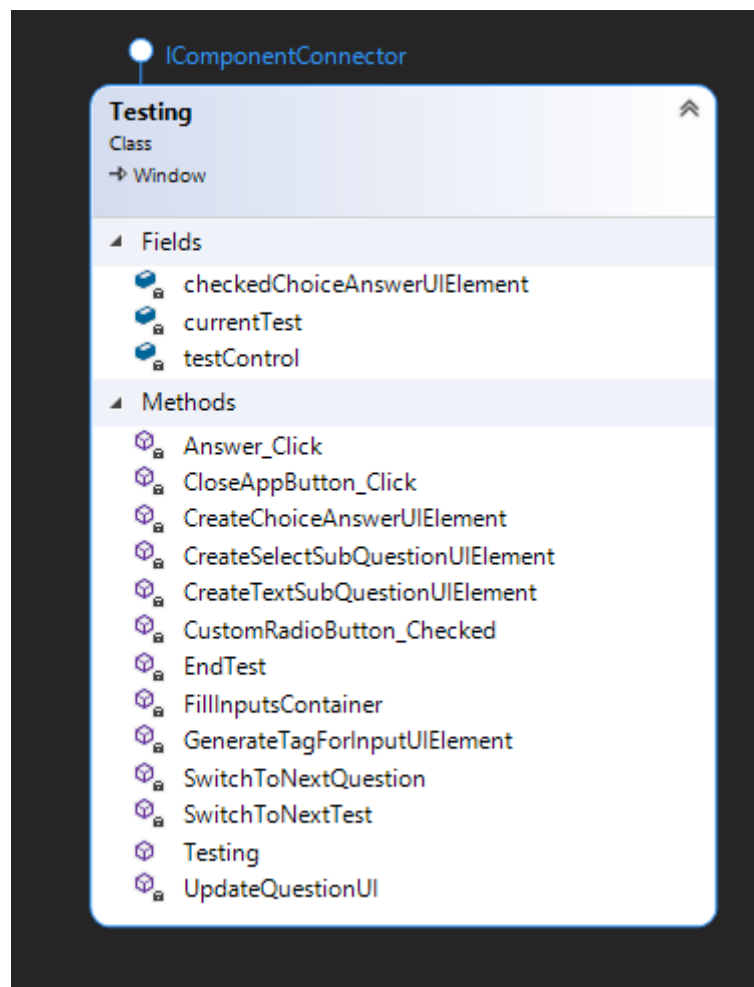


Рисунок 4.16. Перехід у вікно тестування

Після проходження тестів автоматично завантажиться вікно відображення результатів:

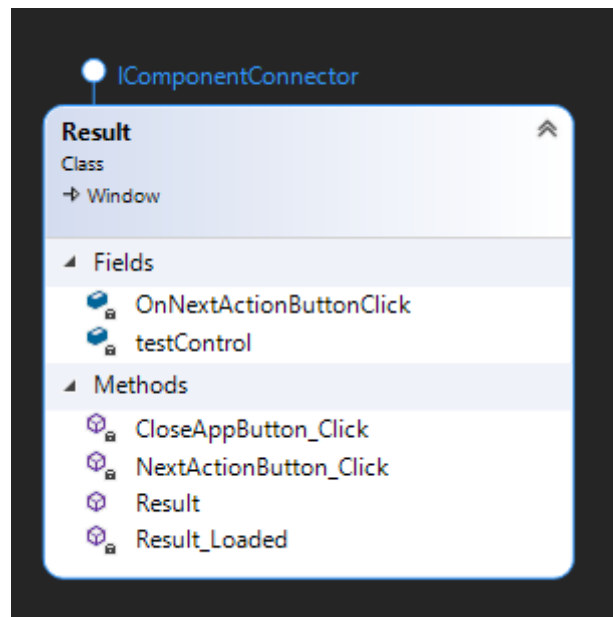


Рисунок 4.17. Результати тестування

Сама система тестування реалізована на основі використання декількох класів сутностей основне завдання яких є опис структури задачі.

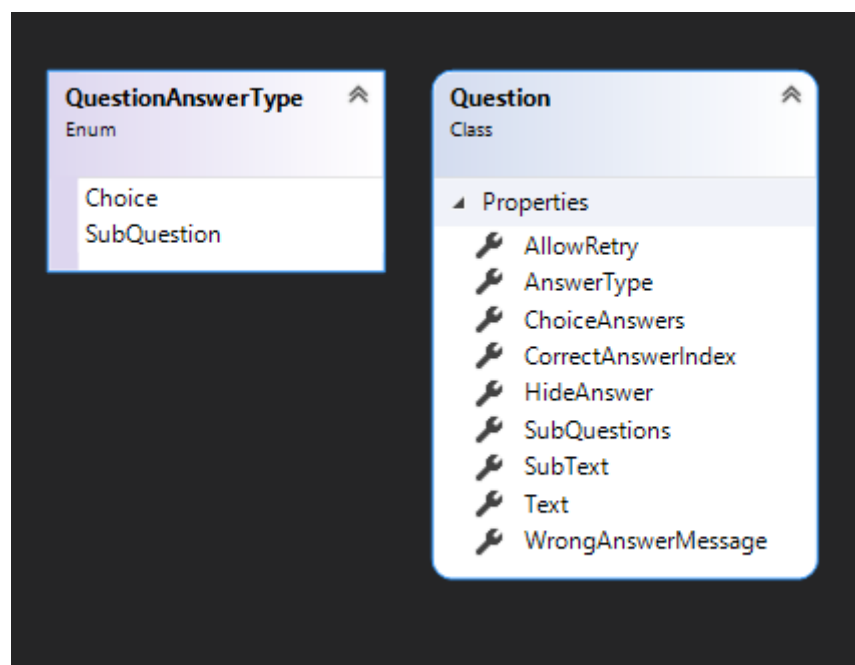


Рисунок 4.18. Класи сутності

Тести, які задаються користувачам зберігаються окремо в json форматі в окремому файлі

Приклад зберігання нашого квесту в json форматі в окремому файлі:

```
{
  "text": "Що таке рекурсивний алгоритм?",
  "answerType": "Choice",
  "choiceAnswers": [
    "Алгоритм, який завжди завершує свою роботу",
    "Алгоритм, який використовує цикли",
    "Алгоритм, який викликає сам себе",
    "Алгоритм, який не потребує вхідних даних"
  ],
  "correctAnswerIndex": "2",
  "wrongAnswerMessage": "Відповідь неправильна.",
  "hideAnswer": true,
  "allowRetry": false
}
```

Механізм, що відповідає за управління квестами:

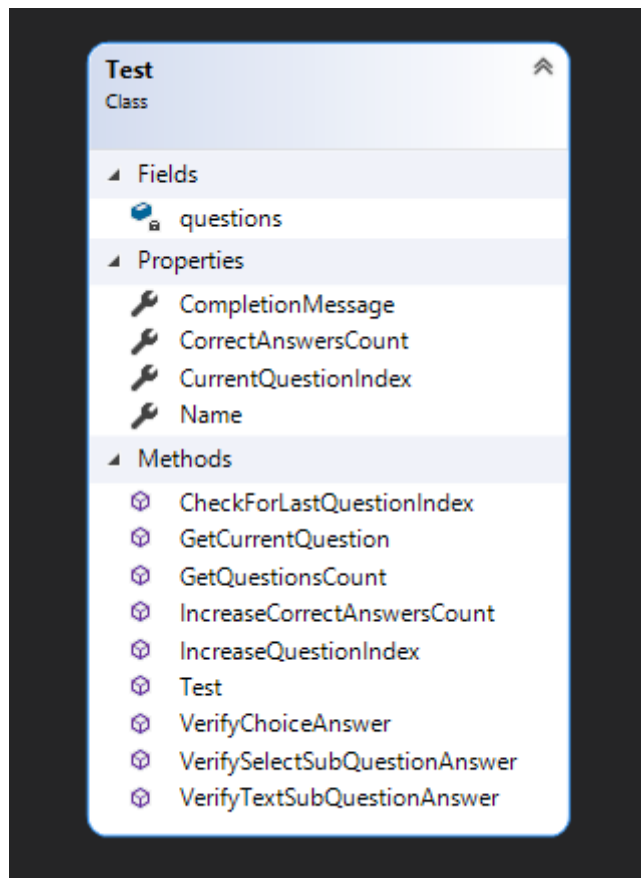


Рисунок 4.19. Механізм управління квестами

За проходження квестів відповідає механізм:

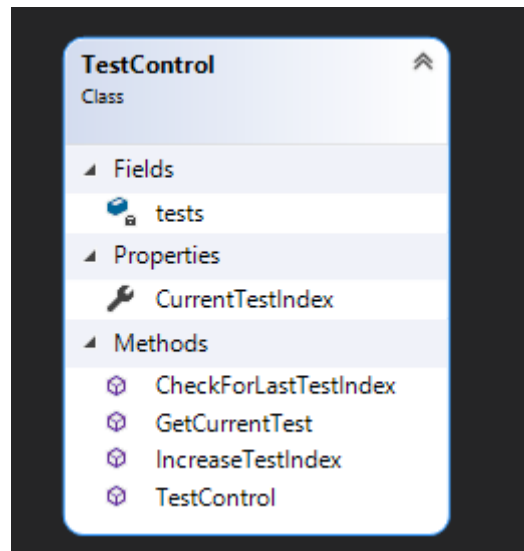


Рисунок 4.19. Механізм проходження квестів

4.2. Інструкція з використання десктопного застосунку

1. Перегляд головного екрану. На початку роботи користувач потрапляє до головного екрану. На головному екрані зазначена назва десктопного застосунку дистанційного навчального курсу «Аналіз алгоритмів» з теми «Рекурсивні алгоритми» та кнопки «Розпочати» та «Про додаток» (Рисунок 4.20).

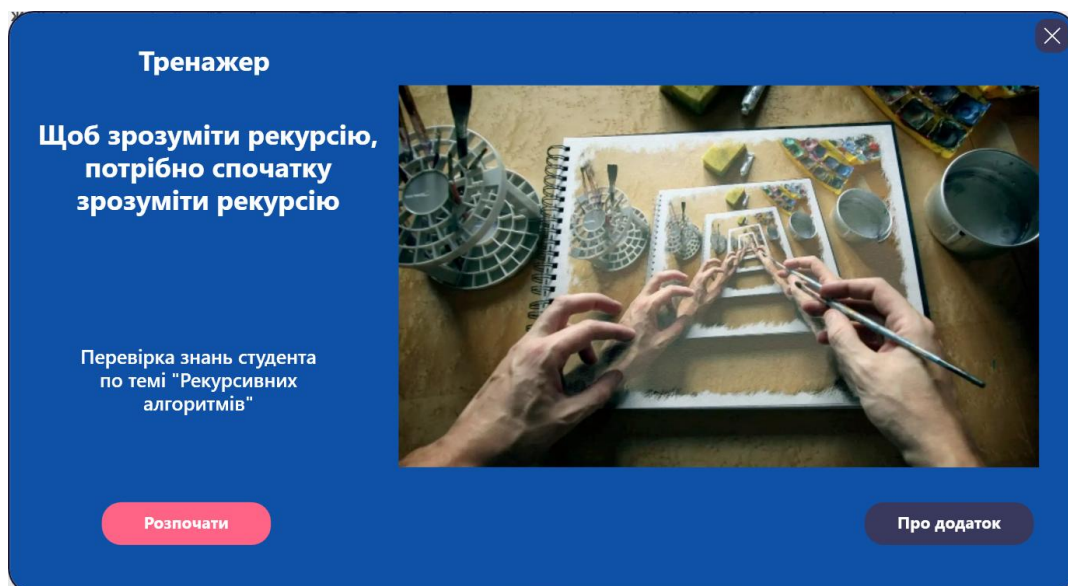


Рисунок 4.20 - Головна сторінка тренажера

2. Розпочати тестування. Щоб розпочати тестування натиснути кнопку «Розпочати» (Рисунок 4.21).

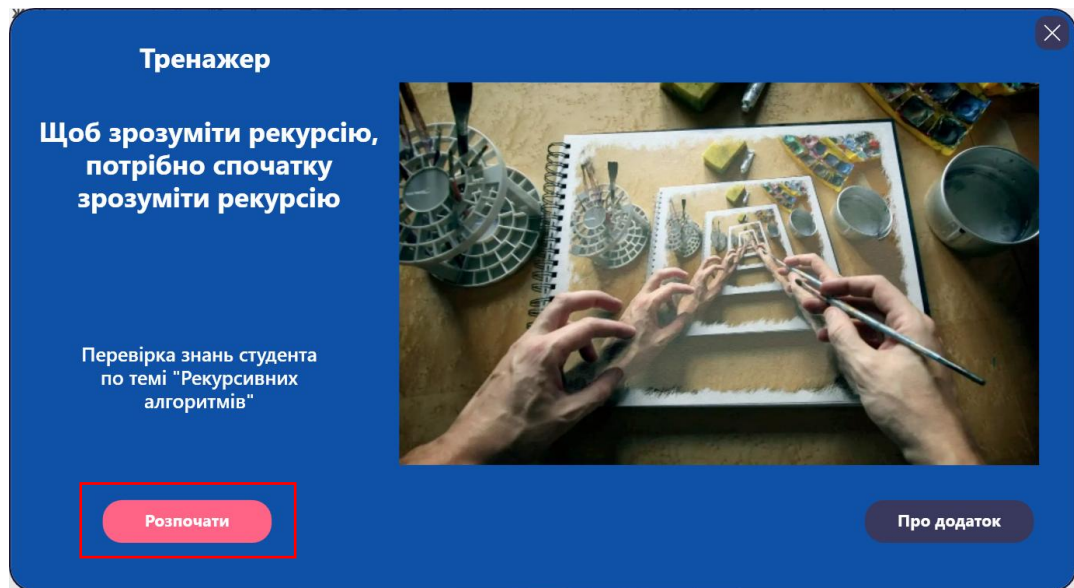


Рисунок 4.21 - Початок роботи з тренажером

3. Вибір тесту. В додатку доступно тести з теми «Рекурсивні алгоритми» (Рисунок 4.22).



Рисунок 4.22 - Вибір тесту

4. Перегляд вікна з тестом. Після того як користувач обрав тест він має можливість переглянути інформацію про: запитання, варіанти відповідей до нього, номер та кількість запитань. (Рисунок 4.23).

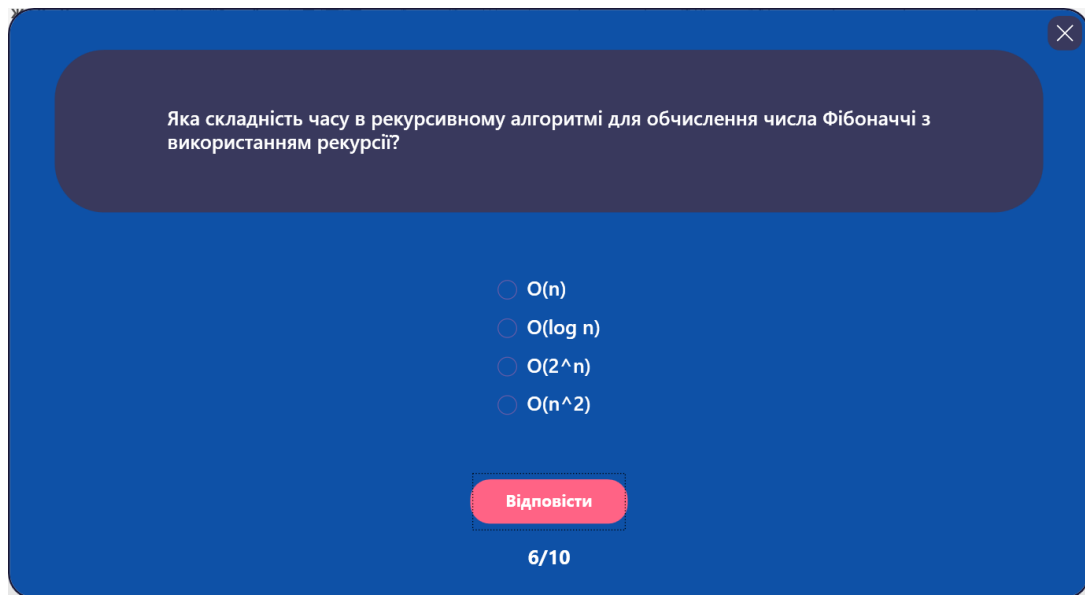


Рисунок 4.23 - Проходження тестування

5. Проходження тестування. Користувачу необхідно вибрати варіант відповіді та натиснути кнопку «Відповісти». В тренажері наявні запитання із однією відповіддю (Рисунок 4.24).

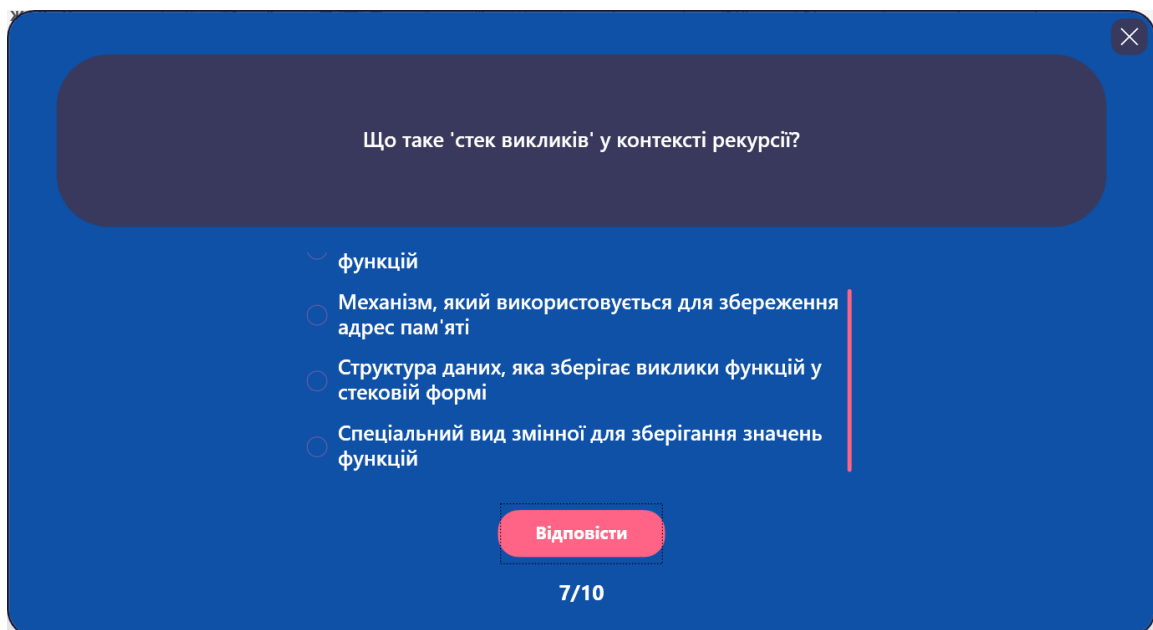


Рисунок 4.24 - Звичайний вибір

5. Перегляд неправильної відповіді. Після того, як користувач обрав неправильну відповідь та натиснув кнопку «Відповісти» на екрані з'являється модальне вікно з повідомленням «Відповідь неправильна» (Рисунок 4.25).

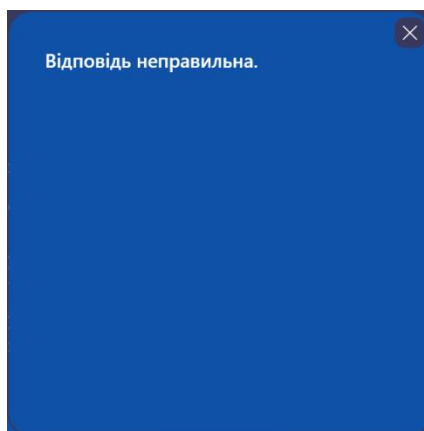


Рисунок 4.25 - Перегляд неправильної відповіді

6. Кінець проходження тестування. Коли користувач вибирає останню відповідь та натискає кнопку «Далі» він потрапляє на екран, де є можливість переглянути статистику проходження тестування (кількість правильних та неправильних відповідей) (Рисунок 4.26).

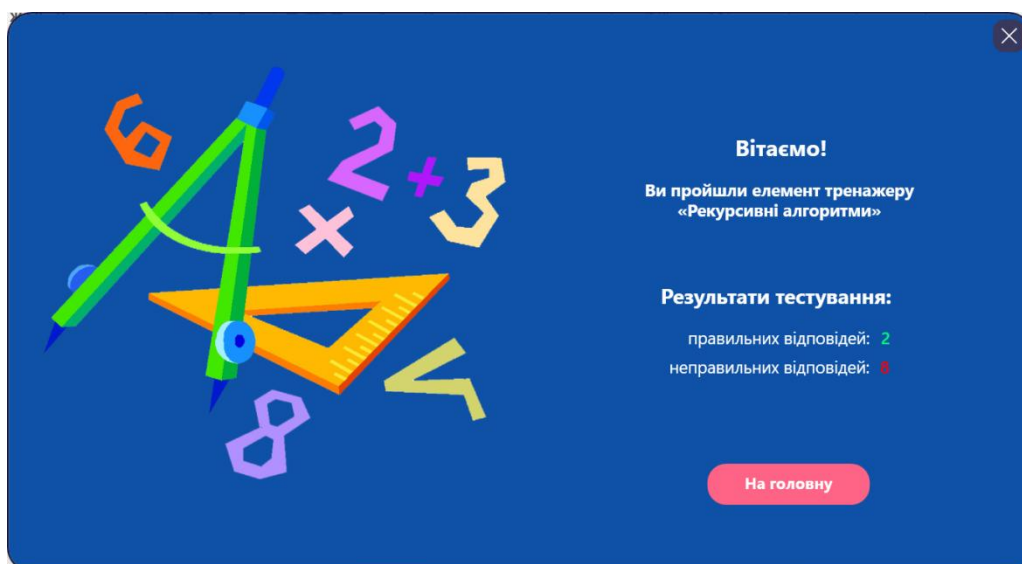


Рисунок 4.26 - Статистика проходження тестування

8. Перегляд інформації про додаток. Щоб переглянути інформацію про додаток потрібно натиснути кнопку «Про додаток» (Рисунок 4.27).

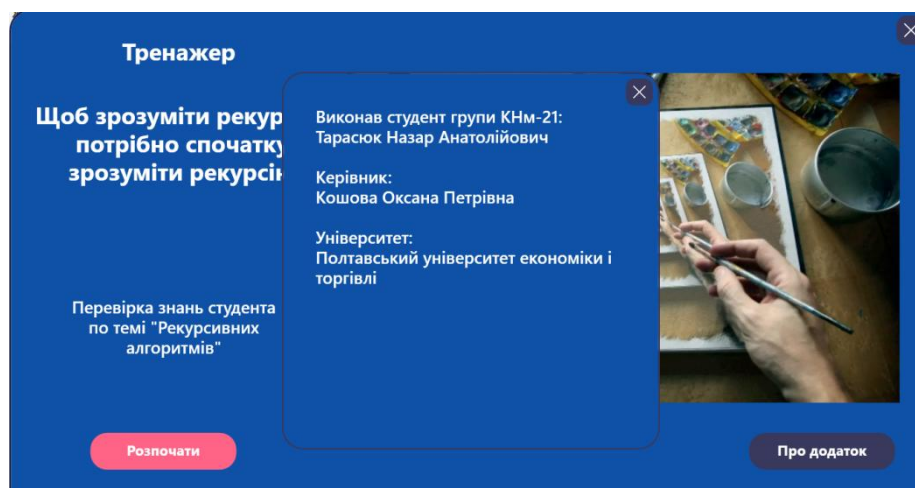


Рисунок 4.27 - Про додаток

ВИСНОВКИ

В сучасному суспільстві дистанційне навчання стає все більш популярним через його зручність і доступність. Використання різноманітних навчальних додатків та застосунків для опанування різними темами, і рекурсивними алгоритмами сортування у тому числі, демонструє, як інноваційні технології можуть удосконалити та спростити навчальний процес. Тому основне завдання нашої дипломної роботи полягало в розробці програмного забезпечення для навчання через десктопний застосунок.

Отже, у результаті виконання дипломного проєкту нами було повністю виконано поставлені завдання дипломної роботи, а саме:

- 1) Визначено ключові вимоги до навчального додатку для вивчення рекурсивних алгоритмів;
- 2) Проведено аналіз переваг та недоліків аналогічних додатків для навчання;
- 3) Проаналізовано та описано підходи та інструменти, використані при створенні навчального десктопного застосунку;
- 4) Розроблено інтуїтивний мобільний інтерфейс для взаємодії користувача з матеріалом;
- 5) Створено докладну інструкцію користувача;
- 6) Представлено блок схему застосунку, діаграму класів та діаграму взаємодії, що ілюструє взаємодію користувача із навчальним застосунком
- 7) Розроблено та апробовано навчальний десктопний застосунок з теми «Рекурсивні алгоритми» для дистанційного навчального курсу «Аналіз алгоритмів».

Розроблений навчальний десктопний застосунок можна використовувати для професійно-орієнтованих дисциплін під час вивчення алгоритмів сортування взагалі і теми «Рекурсивні алгоритми» зокрема.

За темою кваліфікаційної роботи було опубліковано статтю у збірнику

наукових робіт магістів кафедри комп'ютерних наук та інформаційних технологій у 2023 р.

Тренажер може бути впроваджений в навчальний процес з теми «Рекурсивні алгоритми» дистанційного навчального курсу «Аналіз алгоритмів».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Черненко О. О. Курсовий проєкт із фаху : методичні рекомендації щодо оформлення пояснювальних записок до курсового проєкту для студентів спеціальності 122 Комп'ютерні науки освітня програма «Комп'ютерні науки» ступеня бакалавра, магістра /О. О. Черненко. – Полтава : ПУЕТ, 2022. – 58 с. Режим доступу: URL: <http://dspace.puet.edu.ua/handle/123456789/12852> - Назва з екрану
2. Що таке Figma та як її використовувати digital-фахівцю [Електронний ресурс]. – Режим доступу до ресурсу: URL: <https://aboutmarketing.info/internet-marketynh/instrumenty/shcho-take-figma-ta-yak-yiyi-vykorystovuvaty-digital-fakhivtsyu/> – Назва з екрану.
3. What is C# Programming? A Beginner's Guide [Електронний ресурс]. – Режим доступу до ресурсу: URL: <https://www.pluralsight.com/blog/software-development/everything-you-need-to-know-about-c-> – Назва з екрану.
4. What is XAML [Електронний ресурс]. – Режим доступу до ресурсу: URL: <https://www.c-sharpcorner.com/blogs/what-is-xaml122> – Назва з екрану.
5. What Is WPF [Електронний ресурс]. – Режим доступу до ресурсу: URL: <https://www.c-sharpcorner.com/blogs/what-wpf-is1> – Назва з екрану.
6. Microsoft Visual Studio [Електронний ресурс]. – Режим доступу до ресурсу: URL: https://www.wiki-data.uk-ua.nina.az/Visual_Studio.html – Назва з екрану.
7. Фесенко Д. І. Пояснювальна записка до дипломної роботи на тему «Розробка програмного комплексу для виконання та оцінювання завдань з теми «Аналіз алгоритму сортування злиттям» дисципліни «Аналіз алгоритмів»» / Д. І. Фесенко. Полтава : Кафедра КНІТ ПУЕТ, 2021.

ДОДАТОК А

Код програми

```

<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="15.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <Import
Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.props"
Condition="Exists('$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.props
')" />
  <PropertyGroup>
    <Configuration Condition=" '$(Configuration)' == " ">Debug</Configuration>
    <Platform Condition=" '$(Platform)' == " ">AnyCPU</Platform>
    <ProjectGuid>{4BD77BE0-E178-43E1-9CDE-29EEB4E8E9D5}</ProjectGuid>
    <OutputType>WinExe</OutputType>
    <RootNamespace>Quiz</RootNamespace>
    <AssemblyName>Quiz</AssemblyName>
    <TargetFrameworkVersion>v4.7.2</TargetFrameworkVersion>
    <FileAlignment>512</FileAlignment>
    <ProjectTypeGuids>{60dc8134-eba5-43b8-bcc9-bb4bc16c2548};{FAE04EC0-301F-11D3-
BF4B-00C04F79EFBC}</ProjectTypeGuids>
    <WarningLevel>4</WarningLevel>
    <AutoGenerateBindingRedirects>true</AutoGenerateBindingRedirects>
    <Deterministic>true</Deterministic>
    <NuGetPackageImportStamp>
    </NuGetPackageImportStamp>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugSymbols>true</DebugSymbols>
    <DebugType>full</DebugType>
    <Optimize>>false</Optimize>
    <OutputPath>bin\Debug</OutputPath>
    <DefineConstants>DEBUG;TRACE</DefineConstants>
    <ErrorReport>prompt</ErrorReport>
    <WarningLevel>4</WarningLevel>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugType>pdbonly</DebugType>
    <Optimize>true</Optimize>
    <OutputPath>bin\Release</OutputPath>
    <DefineConstants>TRACE</DefineConstants>
    <ErrorReport>prompt</ErrorReport>
    <WarningLevel>4</WarningLevel>
  </PropertyGroup>
  <ItemGroup>
    <Reference Include="Microsoft.Web.WebView2.Core, Version=1.0.1774.30, Culture=neutral,
PublicKeyToken=2a8ab48044d2601e, processorArchitecture=MSIL">
    <HintPath>..\packages\Microsoft.Web.WebView2.1.0.1774.30\lib\net45\Microsoft.Web.WebView
2.Core.dll</HintPath>

```

```

</Reference>
<Reference Include="Microsoft.Web.WebView2.WinForms, Version=1.0.1774.30,
Culture=neutral, PublicKeyToken=2a8ab48044d2601e, processorArchitecture=MSIL">

<HintPath>..\packages\Microsoft.Web.WebView2.1.0.1774.30\lib\net45\Microsoft.Web.WebView
2.WinForms.dll</HintPath>
</Reference>
<Reference Include="Microsoft.Web.WebView2.Wpf, Version=1.0.1774.30, Culture=neutral,
PublicKeyToken=2a8ab48044d2601e, processorArchitecture=MSIL">

<HintPath>..\packages\Microsoft.Web.WebView2.1.0.1774.30\lib\net45\Microsoft.Web.WebView
2.Wpf.dll</HintPath>
</Reference>
<Reference Include="Newtonsoft.Json, Version=13.0.0.0, Culture=neutral,
PublicKeyToken=30ad4fe6b2a6aeed, processorArchitecture=MSIL">
  <HintPath>..\packages\Newtonsoft.Json.13.0.3\lib\net45\Newtonsoft.Json.dll</HintPath>
</Reference>
<Reference Include="System" />
<Reference Include="System.Data" />
<Reference Include="System.Drawing" />
<Reference Include="System.Windows.Forms" />
<Reference Include="System.Xml" />
<Reference Include="Microsoft.CSharp" />
<Reference Include="System.Core" />
<Reference Include="System.Xml.Linq" />
<Reference Include="System.Data.DataSetExtensions" />
<Reference Include="System.Net.Http" />
<Reference Include="System.Xaml">
  <RequiredTargetFramework>4.0</RequiredTargetFramework>
</Reference>
<Reference Include="WindowsBase" />
<Reference Include="PresentationCore" />
<Reference Include="PresentationFramework" />
<Reference Include="WpfMath, Version=1.1.0.0, Culture=neutral,
processorArchitecture=MSIL">
  <HintPath>..\packages\WpfMath.1.1.0\lib\net462\WpfMath.dll</HintPath>
</Reference>
<Reference Include="XamlMath.Shared, Version=1.1.0.0, Culture=neutral,
processorArchitecture=MSIL">
  <HintPath>..\packages\XamlMath.Shared.1.1.0\lib\net462\XamlMath.Shared.dll</HintPath>
</Reference>
</ItemGroup>
<ItemGroup>
  <ApplicationDefinition Include="App.xaml">
    <Generator>MSBuild:Compile</Generator>
    <SubType>Designer</SubType>
  </ApplicationDefinition>
  <Compile Include="Classes\About.cs" />
  <Compile Include="Classes\Constants.cs" />
  <Compile Include="Classes\Helpers.cs" />
  <Compile Include="Classes\Question.cs" />

```



```

<Compile Include="Classes\SubQuestion.cs" />
<Compile Include="Classes\TestControl.cs" />
<Compile Include="Classes\Test.cs" />
<Compile Include="Components\CustomComboBox.xaml.cs">
  <DependentUpon>CustomComboBox.xaml</DependentUpon>
</Compile>
<Compile Include="Components\CustomRadioButton.xaml.cs">
  <DependentUpon>CustomRadioButton.xaml</DependentUpon>
</Compile>
<Compile Include="Components\CustomTextBox.xaml.cs">
  <DependentUpon>CustomTextBox.xaml</DependentUpon>
</Compile>
<Compile Include="Components\MathTextBlock.xaml.cs">
  <DependentUpon>MathTextBlock.xaml</DependentUpon>
</Compile>
<Compile Include="Components\SelectInputQuestion.xaml.cs">
  <DependentUpon>SelectInputQuestion.xaml</DependentUpon>
</Compile>
<Compile Include="Components\TextInputQuestion.xaml.cs">
  <DependentUpon>TextInputQuestion.xaml</DependentUpon>
</Compile>
<Compile Include="Views\Popup.xaml.cs">
  <DependentUpon>Popup.xaml</DependentUpon>
</Compile>
<Compile Include="Views\Result.xaml.cs">
  <DependentUpon>Result.xaml</DependentUpon>
</Compile>
<Compile Include="Views\Testing.xaml.cs">
  <DependentUpon>Testing.xaml</DependentUpon>
</Compile>
<Compile Include="Views\TestChoice.xaml.cs">
  <DependentUpon>TestChoice.xaml</DependentUpon>
</Compile>
<Compile Include="Views\WebView.xaml.cs">
  <DependentUpon>WebView.xaml</DependentUpon>
</Compile>
<Page Include="Components\CustomComboBox.xaml">
  <SubType>Designer</SubType>
  <Generator>MSBuild:Compile</Generator>
</Page>
<Page Include="Components\CustomRadioButton.xaml">
  <SubType>Designer</SubType>
  <Generator>MSBuild:Compile</Generator>
</Page>
<Page Include="Components\CustomTextBox.xaml">
  <SubType>Designer</SubType>
  <Generator>MSBuild:Compile</Generator>
</Page>
<Page Include="Components\MathTextBlock.xaml">
  <SubType>Designer</SubType>
  <Generator>MSBuild:Compile</Generator>

```

```

</Page>
<Page Include="Components\SelectInputQuestion.xaml">
  <SubType>Designer</SubType>
  <Generator>MSBuild:Compile</Generator>
</Page>
<Page Include="Components\TextInputQuestion.xaml">
  <SubType>Designer</SubType>
  <Generator>MSBuild:Compile</Generator>
</Page>
<Page Include="Views\Main.xaml">
  <Generator>MSBuild:Compile</Generator>
  <SubType>Designer</SubType>
</Page>
<Compile Include="App.xaml.cs">
  <DependentUpon>App.xaml</DependentUpon>
  <SubType>Code</SubType>
</Compile>
<Compile Include="Views\Main.xaml.cs">
  <DependentUpon>Main.xaml</DependentUpon>
  <SubType>Code</SubType>
</Compile>
<Page Include="Views\Popup.xaml">
  <SubType>Designer</SubType>
  <Generator>MSBuild:Compile</Generator>
</Page>
<Page Include="Views\Result.xaml">
  <SubType>Designer</SubType>
  <Generator>MSBuild:Compile</Generator>
</Page>
<Page Include="Styles\ScrollBar.xaml">
  <SubType>Designer</SubType>
  <Generator>MSBuild:Compile</Generator>
</Page>
<Page Include="Views\Testing.xaml">
  <SubType>Designer</SubType>
  <Generator>MSBuild:Compile</Generator>
</Page>
<Page Include="Views\TestChoice.xaml">
  <SubType>Designer</SubType>
  <Generator>MSBuild:Compile</Generator>
</Page>
<Page Include="Views\WebView.xaml">
  <SubType>Designer</SubType>
  <Generator>MSBuild:Compile</Generator>
</Page>
</ItemGroup>
<ItemGroup>
  <Compile Include="Properties\AssemblyInfo.cs">
    <SubType>Code</SubType>
  </Compile>
  <Compile Include="Properties\Resources.Designer.cs">

```

```

    <AutoGen>True</AutoGen>
    <DesignTime>True</DesignTime>
    <DependentUpon>Resources.resx</DependentUpon>
</Compile>
<Compile Include="Properties\Settings.Designer.cs">
    <AutoGen>True</AutoGen>
    <DependentUpon>Settings.settings</DependentUpon>
    <DesignTimeSharedInput>True</DesignTimeSharedInput>
</Compile>
<Resource Include="Views\matemat.png" />
<Resource Include="Views\recurse.png" />
<Resource Include="Views\recurseimg.png" />
<Resource Include="Assets\Images\Board1.png" />
<Resource Include="Assets\Images\Board2.png" />
<EmbeddedResource Include="Properties\Resources.resx">
    <Generator>ResXFileCodeGenerator</Generator>
    <LastGenOutput>Resources.Designer.cs</LastGenOutput>
</EmbeddedResource>
<Content Include="Assets\Fonts\Inter\Inter-Regular.ttf">
    <CopyToOutputDirectory>PreserveNewest</CopyToOutputDirectory>
</Content>
<None Include="Assets\Documents\Theoretical material.pdf" />
<None Include="ClassDiagram.cd" />
<None Include="Config\about.json" />
<None
Include="Config\Tests\ElementsOfCombinatoricsInProbabilityCalculations\questions1.json" />
    <None
Include="Config\Tests\ElementsOfCombinatoricsInProbabilityCalculations\questions2.json" />
    <None Include="packages.config" />
    <None Include="Properties\Settings.settings">
        <Generator>SettingsSingleFileGenerator</Generator>
        <LastGenOutput>Settings.Designer.cs</LastGenOutput>
    </None>
    <None Include="Config\Tests\ElementsOfCombinatorics\questions.json" />
</ItemGroup>
<ItemGroup>
    <None Include="App.config" />
</ItemGroup>
<ItemGroup>
    <Resource Include="Assets\Images\Close.png" />
</ItemGroup>
<ItemGroup>
    <Resource Include="Assets\Images\GroupMath.png" />
</ItemGroup>
<ItemGroup>
    <Resource Include="Assets\Images\Comp.png" />
</ItemGroup>
<ItemGroup>
    <Resource Include="Assets\Images\PiMath.png" />
</ItemGroup>
<ItemGroup>

```

```

    <Resource Include="Assets\Images\QuizMan.png" />
  </ItemGroup>
  <ItemGroup>
    <Resource Include="Assets\Images\Result1.png" />
  </ItemGroup>
  <ItemGroup>
    <Resource Include="Assets\Images\Result2.png" />
  </ItemGroup>
  <ItemGroup />
  <Import Project="$(MSBuildToolsPath)\Microsoft.CSharp.targets" />
  <Import
Project="..\packages\Microsoft.Web.WebView2.1.0.1774.30\build\Microsoft.Web.WebView2.targets"
Condition="Exists('..\packages\Microsoft.Web.WebView2.1.0.1774.30\build\Microsoft.Web.WebV
iew2.targets')" />
  <Target Name="EnsureNuGetPackageBuildImports" BeforeTargets="PrepareForBuild">
    <PropertyGroup>
      <ErrorText>This project references NuGet package(s) that are missing on this computer. Use
NuGet Package Restore to download them. For more information, see
http://go.microsoft.com/fwlink/?LinkID=322105. The missing file is {0}.</ErrorText>
    </PropertyGroup>
    <Error
Condition="!Exists('..\packages\Microsoft.Web.WebView2.1.0.1774.30\build\Microsoft.Web.Web
View2.targets')"
Text="$([System.String]::Format('$(ErrorText)',
'..\packages\Microsoft.Web.WebView2.1.0.1774.30\build\Microsoft.Web.WebView2.targets'))" />
    </Target>
  </Project>

```

Microsoft Visual Studio Solution File, Format Version 12.00

Visual Studio Version 16

VisualStudioVersion = 16.0.30114.105

MinimumVisualStudioVersion = 10.0.40219.1

Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "Quiz", "Quiz\Quiz.csproj",
"{4BD77BE0-E178-43E1-9CDE-29EEB4E8E9D5}"

EndProject

Global

GlobalSection(SolutionConfigurationPlatforms) = preSolution

Debug|Any CPU = Debug|Any CPU

Release|Any CPU = Release|Any CPU

EndGlobalSection

GlobalSection(ProjectConfigurationPlatforms) = postSolution

{4BD77BE0-E178-43E1-9CDE-29EEB4E8E9D5}.Debug|Any CPU.ActiveCfg =
Debug|Any CPU

{4BD77BE0-E178-43E1-9CDE-29EEB4E8E9D5}.Debug|Any CPU.Build.0 =
Debug|Any CPU

{4BD77BE0-E178-43E1-9CDE-29EEB4E8E9D5}.Release|Any CPU.ActiveCfg =
Release|Any CPU

{4BD77BE0-E178-43E1-9CDE-29EEB4E8E9D5}.Release|Any CPU.Build.0 =
Release|Any CPU

EndGlobalSection

```
GlobalSection(SolutionProperties) = preSolution
    HideSolutionNode = FALSE
EndGlobalSection
GlobalSection(ExtensibilityGlobals) = postSolution
    SolutionGuid = {460A3DEB-836C-44D4-BAE1-97F83C2FD834}
EndGlobalSection
EndGlobal
```