

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)
«__» _____ 202_ р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З ТЕМИ «РОБОТА З ДІАПАЗОНАМИ (RANGE) В EXCEL VBA» НАВЧАЛЬНОГО КУРСУ «ОРГАНІЗАЦІЯ ТА ОБРОБКА ЕЛЕКТРОННОЇ ІНФОРМАЦІЇ»

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

Виконавець роботи Мартинюк Гліб Віталійович
_____ «__» _____ 202_ р.
(підпис)

Науковий керівник к. ф.- м. н., доцент, Чілікіна Тетяна Василівна
_____ «__» _____ 202_ р.
(підпис)

Рецензент _____

Полтава – 2023

ЗАТВЕРДЖУЮ
 Завідувач кафедри _____ Олена ОЛЬХОВСЬКА
 «__» вересня 202__р.

ЗАВДАННЯ І КАЛЕНДАРНИЙ ГРАФІК ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

на тему «Розробка програмного забезпечення з теми «Робота з діапазонами (Range) в Excel VBA» навчального курсу «Організація та обробка електронної інформації»

зі спеціальності 122 Комп'ютерні науки»

освітня програма «Комп'ютерні науки»

ступеня магістр

Прізвище, ім'я, по батькові Мартинюк Гліб Віталійович

Затверджена наказом ректора № ____-Н від «__» _____ 202__ р.

Термін подання студентом роботи «__» _____ 202__ р.

Вихідні дані до кваліфікаційної роботи: публікації з теми, навчальні тренажери в дистанційних курсах із комп'ютерних наук.

Зміст пояснювальної записки (перелік питань, які потрібно розробити)

ВСТУП

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1 Перегляд основ дистанційного навчання та його складових

2.2 Огляд програмних продуктів подібних до тематики

2.3 Позитивні аспекти оглянутих програмних продуктів

2.4 Негативні аспекти оглянутих програмних продуктів

2.5 Огляд засобів для створення навчальних симуляторів

РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

3.1 Огляд матеріалу з теми

3.2 Алгоритм роботи тренажера

3.3 Блок-схема роботи тренажера

РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА

4.1 Обґрунтування вибору мови програмування

4.2 Обґрунтування вибору програмних засобів

4.3 Опис програмної реалізації тренажеру

4.4 Тестування створеного продукту

ВИСНОВКИ

СПИСОК ЛІТЕРАТУРИ

ДОДАТОК А. ДИСК З МАТЕРІАЛАМИ

Перелік графічного матеріалу: 2-3 аркуші блок-схем, інші необхідні ілюстрації.

Консультанти розділів кваліфікаційної роботи

Розділ	ППП, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Постановка задачі	Чілікіна Т.В.		
Інформаційний огляд	Чілікіна Т.В.		
Теоретична частина	Чілікіна Т.В.		
Практична частина	Чілікіна Т.В.		

Календарний графік виконання кваліфікаційної роботи

Зміст роботи	Термін виконання	Фактичне виконання
1. Вступ		
2. Вивчення методичних рекомендацій і 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. Доопрацювання (за необхідності), рецензування		

Дата видачі завдання «__» _____ 202_р.

Здобувач вищої освіти

Гліб МАРТИНЮК

Науковий керівник

к. ф.-м. н Тетяна ЧІЛКІНА**Результати захисту кваліфікаційної роботи**

Кваліфікаційна робота оцінена на _____

Протокол засідання ЕК № ____ від «__» _____ 202_р.

Секретар ЕК _____

*(підпис)**(ініціал та прізвище)*

Затверджую

Зав. кафедрою _____
 к.ф.-м.н. Олена Ольховська
 « ____ » _____ 202__ р.

Погоджено

Науковий керівник _____
 к.ф.-м.н. Тетяна ЧІЛКІНА
 « ____ » _____ 202__ р.

План

кваліфікаційної роботи ступеня магістр
 зі спеціальності 122 Комп'ютерні науки
 освітня програма 122 Комп'ютерні науки
Мартинюк Гліб Віталійович
 Прізвище, ім'я, по батькові

на тему «Розробка програмного забезпечення з теми «Робота з діапазонами (Range) в Excel VBA» навчального курсу «Організація та обробка електронної інформації»

ВСТУП**РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ****РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД**

- 2.1 Перегляд основ дистанційного навчання та його складових
- 2.2 Огляд подібних програмних продуктів
- 2.3 Позитивні аспекти оглянутих програмних продуктів
- 2.4 Негативні аспекти оглянутих програмних продуктів
- 2.5 Огляд засобів для створення навчальних симуляторів

РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

- 3.1. Огляд матеріалу з теми
- 3.2. Алгоритм роботи тренажера
- 3.3 Блок-схема роботи тренажера

РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА

- 4.1 Обґрунтування вибору мови програмування
- 4.2 Обґрунтування вибору програмних засобів
- 4.3 Опис програмної реалізації тренажеру
- 4.4 Тестування створеного продукту

ВИСНОВОК

Здобувач вищої освіти _____ Гліб МАРТИНЮК
 « ____ » _____ 202__ р.

АНОТАЦІЯ

Записка: 41 с., 20 рис., 1 додаток, 10 джерел.

Мета роботи – алгоритмізація та програмування тренажера з теми «Робота з діапазонами (Range) в Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації».

Об'єкт розробки – є програмна реалізація навчального симулятора з теми «Робота з діапазонами (Range) в Excel VBA»

Методи дослідження та інформаційне забезпечення - методи теорії числових послідовностей, середовище розробки Visual Studio Code та мова програмування Python.

Результати дослідження. Розроблено алгоритм навчального тренажера з теми "Робота з діапазонами (Range) в Excel VBA", побудовано його блок-схему. Здійснена програмна реалізація в середовищі Visual Studio Code за допомогою мови програмування Python, проведення тестування створеного продукту.

Наукова апробація. За матеріалами роботи опубліковано тези (науково-практичний семінар «Комп'ютерні науки та інформаційні технології» (КНІТ-2023)).

Ключеві слова: *VBA EXCEL, Організація та обробка електронної інформації, мова програмування Python, програмна реалізація.*

Зміст

ВСТУП	7
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ.....	9
РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	11
2.1. Перегляд основ дистанційного навчання та визначення тренажерів	11
2.2. Огляд подібних програмних продуктів	12
2.3. Позитивні аспекти оглянутих програмних продуктів	17
2.4. Негативні аспекти оглянутих програмних продуктів.....	18
2.5. Огляд засобів для створення навчальних симуляторів	18
РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА	20
3.1. Огляд матеріалу з теми «Робота з діапазонами (Range) в Excel VBA»	20
3.2. Алгоритм роботи тренажера	21
3.3 Блок-схема роботи тренажеру	23
РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА	26
4.1 Обґрунтування вибору мови програмування.....	26
4.2 Обґрунтування вибору програмних засобів	27
4.3 Опис програмної реалізації тренажеру	29
4.4 Тестування створеного продукту	36
ВИСНОВКИ	38
СПИСОК ЛІТЕРАТИ.....	39
ДОДАТОК А. ДИСК З МАТЕРІАЛАМИ.....	41

ВСТУП

Наслідком бурхливого розвитку інформатизації суспільства, постійного зниження вартості послуг на використання глобальної мережі Internet та суттєве поглиблення процесів упровадження інформаційних технологій в освітню практику стала поява дистанційного навчання як найбільш перспективної, гуманістичної, інтегральної і орієнтованої на індивідуалізацію форми освітнього процесу.

Для підтримки навичок студента, створюються тренажери з розв'язання задач зв'язаних з їхніми курсами. Так звані програми-тренажери забезпечують:

- Можливість перейти на офіційний сайт за допомогою програми;
- Виведення на екран питань з вибраної теми;
- Можливість вибрати відповідь, яка на вашу думку правильна;
- Реакція програми на цю саму відповідь;
- Демонстрація правильної відповіді;
- Можливість подивитися теорію пов'язану з питанням.

Мета проекту – Розробка програмного забезпечення з теми «Робота з діапазонами (Range) в Excel VBA» навчального курсу «Організація та обробка електронної інформації».

Об'єктом розробки є програмна реалізація навчального тренажеру з теми «Організація та обробка електронної інформації» дистанційного навчального курсу «Організація та обробка електронної інформації».

Предмет розробки — тренажер з теми «Робота з діапазонами (Range) в Excel VBA».

Перелік використаних методів: при створенні програми-тренажера було використано мову програмування Python;

Пояснювальна записка складається з чотирьох розділів. У першому розділі розглянуто постановку задачі. У другому розділі було оглянуто подібні програмні продукти та виділено позитивні та негативні аспекти. У третьому розділі представлено огляд матеріалу з теми, алгоритм роботи тренажера та блок-схеми роботи тренажера. У четвертому розділі розглянуто якими методами створювався тренажер та яке програмне середовище було обрано для розробки.

У додатку міститься диск з матеріалами проекту.

Обсяг пояснювальної записки: 41 стор., в т.ч. основна частина - 31 стор., додатки - 1 стор., рис. 20, джерела - 10 назв.

Враховуючи актуальні тенденції у сучасній освіті та використання передових технологій, можна успішно поєднувати традиційні методи навчання із зручними дистанційними форматами. Розроблений навчальний тренажер надає унікальну можливість оптимізувати процес вивчення «Excel VBA», дозволяючи студентам отримувати доступ до необхідних ресурсів у будь-який час і в будь-якому місці.

Забезпечуючи відповідність сучасним вимогам до методів навчання, створений програмний продукт інтегрується в освітній процес, сприяючи підвищенню якості викладання обробки електронної інформації. Активне використання цього тренажера в освітніх закладах акцентує його актуальність і великий внесок у сучасну освітню практику. Навички отримані перевірені тренажером, які надають компетенцію у роботі з діапазонами для студентів, що вивчають предмети пов'язані з «Excel».

В рамках даного проекту створено не лише ефективний дидактичний інструмент, але й впроваджено інтерактивні завдання, спрямовані на перевірку знань учня для праці з діапазонами у «Excel VBA». Це значно підвищує практичну цінність навчання та сприяє більш ефективному засвоєнню матеріалу.

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

Мета кваліфікаційної роботи - створення тренажера із легким та зручним дизайном, а також інтуїтивно зрозумілим функціоналом.

Для вирішення задач, поставлених за допомогою комп'ютерних технологій. Які повинна проходити людина, а безпосередньо комп'ютер повинен їх перевіряти. Задачі розбиті на такі етапи:

- Вибір теми;
- Аналіз теми для створення задач;
- Формулювання мети цього тренажера;
- Вибір мови програмування та середовища розробки, які найбільш підходять для створення тренажера специфічної теми;
- Організація даних для створення, тренажера та алгоритмів для нього;
- Створення коду та візуального інтерфейсу тренажера;
- Впровадження питань з вибраної теми;
- Тестування тренажера;
- Перевірка візуального інтерфейсу;
- Виявлення помилок;
- Виправлення помилок;
- Компілювання в фінальну версію тренажера.

Для створення візуально простого тренажуру, я вибрав «Python», а саме бібліотеку «Tkinter».

Поставити цілі які повинен виконати цей тренажер, це одна з найперших та найбільш важливих задач, яка стоїть перед програмістом. Якщо на цьому етапі допустити помилку або ж не передбачити подальші зміни у програмі. То їх вирішення у подальшому створюванні програми буде вкрай проблематично. Не говорячи про не правильно поставленні питання або питання які не відносяться до теми.

Щоб цього не допустити потрібно дотримуватися наступних етапів:

- Перевірити чи підтримає, вибрана вами мова, UI або функції які ви хочете використовувати у своєму тренажері.
- Порахувати скільки вам потрібно питань, щоб потім їх не додавати/видаляти.
- Перевірте щоб питання які ви вписуєте, були пов'язанні з темою, і могли бути реалізовані на вибраній програмній мові.
- Визначитися на якій Операційній системі буде працювати тренажер, так як в Україні більшість людей має Windows , то буде розумно вибрати саме цю Операційну систему.
- Перевірити тренажер на людях які зв'язані з цією темою.
- Опитати цих людей, та вияснити чи є в ній проблеми.

Якщо проблем не виявлено, то можна випускати свій тренажер у Інтернет, або його фізичну копію в освітніх закладах, таких як: Школа, Університет, Технікум, бібліотека та інші.

Але якщо це комерційний проект, тоді будуть використані інші правила для створення тренажеру, так як вам потрібно буде створити тренажер за основу якого ляже документ, через який ви узгоджували на розробку цього тренажеру.

З офіційним договором зазвичай передують:

- З'ясування чи необхідна така система в комп'ютерному вигляді або чи потрібна вона взагалі.
- Вияснення теми на основі якої будуть створенні питання для підтримки теоретичних знань працівників компанії.
- Оцінка можливості створення цієї програми та подальшої її підтримки.
- Оцінка бюджету який буде виділений на створення цього тренажеру.
- Тестування тренажеру на людях які будуть використовувати цю програму в комерційних цілях.
- На основі реакції працівник впроваджувати оновлення.

Після цих етапів, керівництво складає список вимог. В якому написано:

- Аналіз мов на які потрібно перекласти цей тренажер, якщо це міжнародна корпорація.
- Опис функцій які можливо замінити або оновити, тільки при необхідності.
- Обмеження які повинні враховуватися при створенні програми.

У більшості випадках, головним етапом вважається аналіз. Так як після нього набагато простіше знайти помилки або деякі відсутні, але потрібні функції тренажеру.

РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Перегляд основ дистанційного навчання та визначення тренажерів

Сучасний підхід до навчання, відомий як дистанційне навчання, є освітнім процесом, що використовує інформаційні технології для забезпечення студентам можливості отримання знань та взаємодії з викладачами, незалежно від їх фізичного місцеперебування. Цей метод відзначається відсутністю необхідності у фізичній присутності учасників, використанням дистанційних засобів комунікації та гнучкістю графіку навчання.

Створення тренажерів для дистанційного навчання впливає з необхідності забезпечення студентам можливості практичного відпрацювання навичок, які є ключовими для їхнього подальшого професійного розвитку. Це може стосуватися різних галузей, таких як медицина, техніка та військова підготовка, з урахуванням технічних можливостей, педагогічних цілей та конкретного предмету навчання.

Кожен вид тренажера виконує свою конкретну функцію. Наприклад, віртуальні імітатори дозволяють користувачам реплікувати реальні сценарії, створюючи віртуальні середовища для практичного навчання. Медичні тренажери, навпаки, спрямовані на тренування медичних навичок. Різні дидактичні інструменти можуть надавати різноманітні можливості для оцінки успішності та відстеження прогресу користувачів, а їхні особливості можуть виявлятися у методах створення сценаріїв та реалізації інтерактивних елементів. Також деякі тренажери можуть бути більш мобільними та доступними для використання на різних пристроях, що розширює їх універсальність.

В освітньому контексті, навчальні тренажери, які поєднують теоретичне вивчення з виконанням практичних завдань, представляють собою інноваційний підхід до дистанційного навчання. Ці електронні системи інтегрують теоретичні знання з можливістю практичного застосування, створюючи комплексний освітній досвід для студентів.

2.2. Огляд подібних програмних продуктів

Під час підготовки магістерської роботи проведено аналіз навчальних тренажерів кількох випадково вибраних студентів, які навчалися за спеціальністю "122 Комп'ютерні науки", таких як: Антоненко Анною; Русіном Владиславом; Голубенко Владиславом.

Антоненко Анна. Розробила тренажер з теми «Дефазифікація нечітких множин з дискретним носієм», цей тренажер був розроблений на мові програмування Object Pascal у середовищі Delphi.

На стартовому вікні програми ми можемо побачити: Хто автор тренажер; Велику кнопку старт; Вибір розділу завдань. (рис. 2.1)

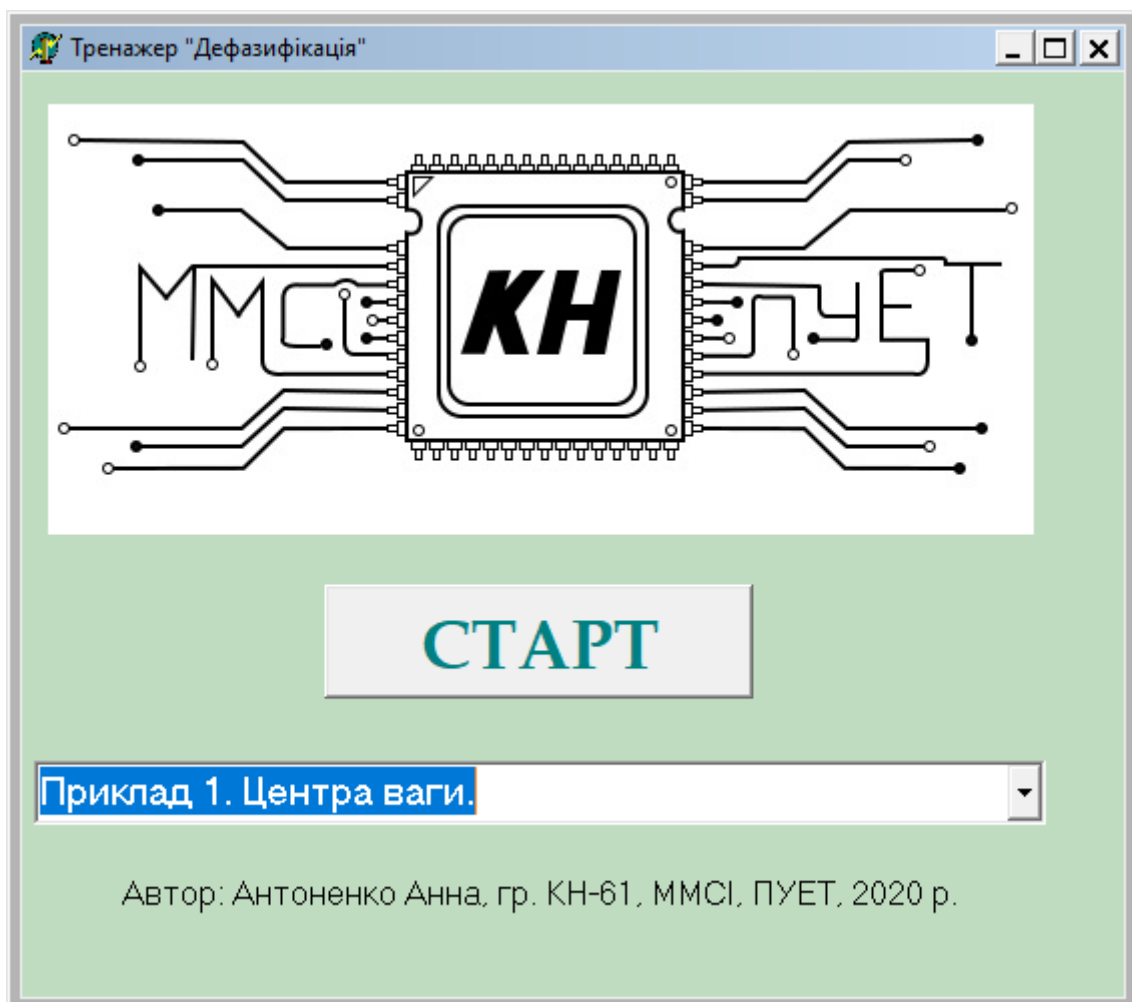


Рисунок 2.1 – Стартове вікно

Після вибору першого розділу нас зустрічає (рис. 2.2) вікно з питанням один, на якому ми можемо побачити наступні віджети: Умова; Питання; Комірка для відповіді; Кнопка перевірки відповіді.

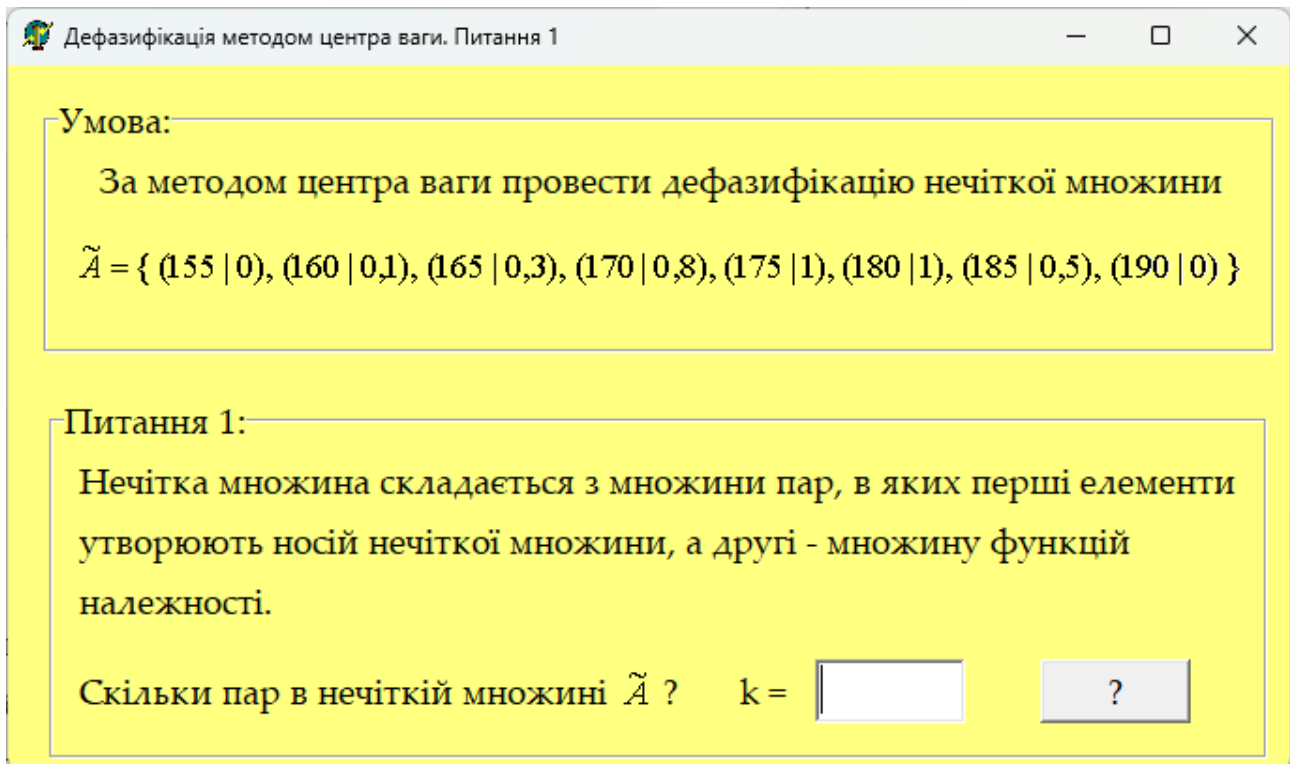


Рисунок 2.2 – Вікно з питаннями

У випадку неправильної відповіді, відкриється вікно з підказкою. (рис. 2.3)

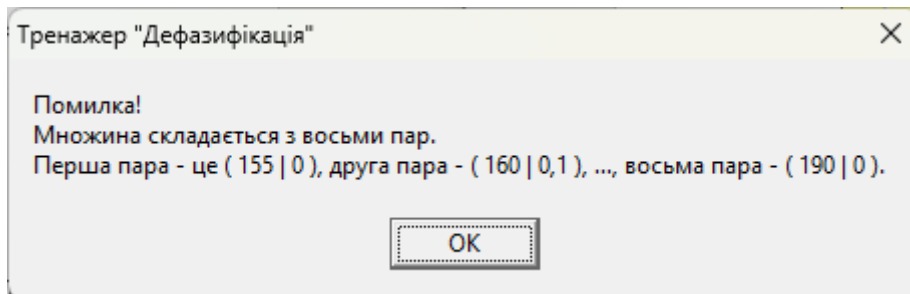


Рисунок 2.3 – Вікно з підказкою

Після вирішення усіх питань, нас зустріне фінальне сповіщення. (рис. 2.4)

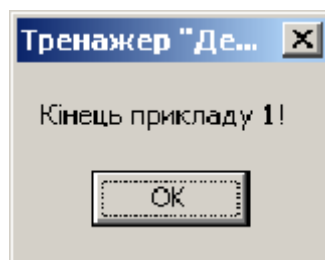


Рисунок 2.4 – Фінальне оповіщення

Тренажер, створений студентом Русіном Владиславом, який є веб-Сторінкою, на якій зображено: Тема тренажера; Автор; Керівник; Кафедра; Кнопка для початку тренінгу. (рис. 2.5).

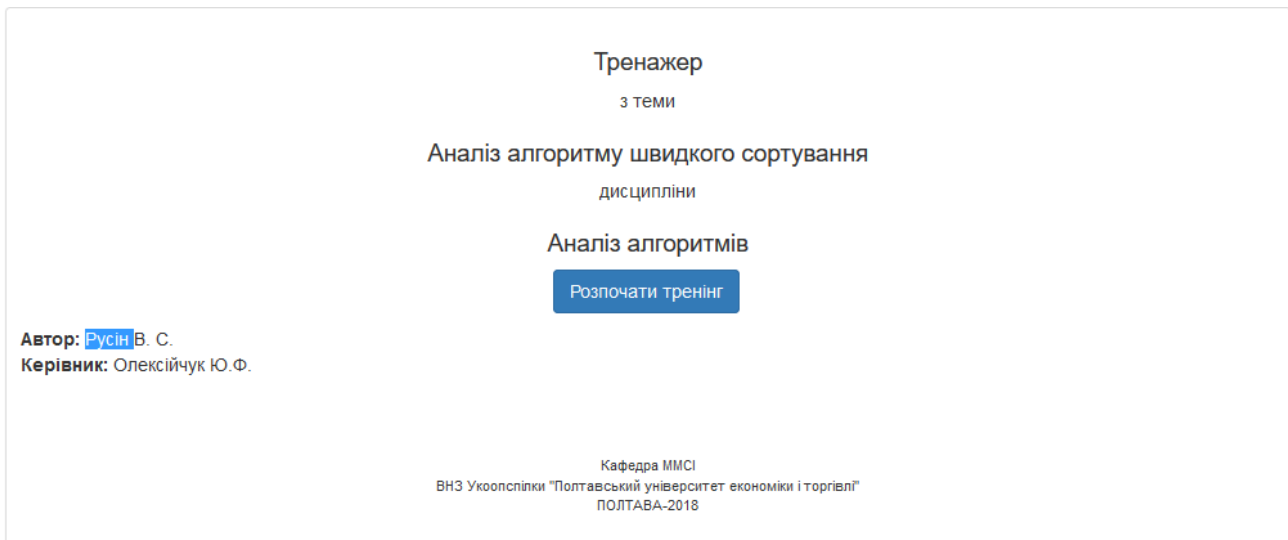


Рисунок 2.5 – Головне вікно

Після початку тренінгу, відкривається нове вікно «Налаштування тренінгу» (рис. 2.6), на якому є такі віджети: Описання що робити; Вибір кількості елементів сортування від 4 до 8; Можливість власноруч заповнити кожен елемент або сгенерувати його; Кнопка «Далі».

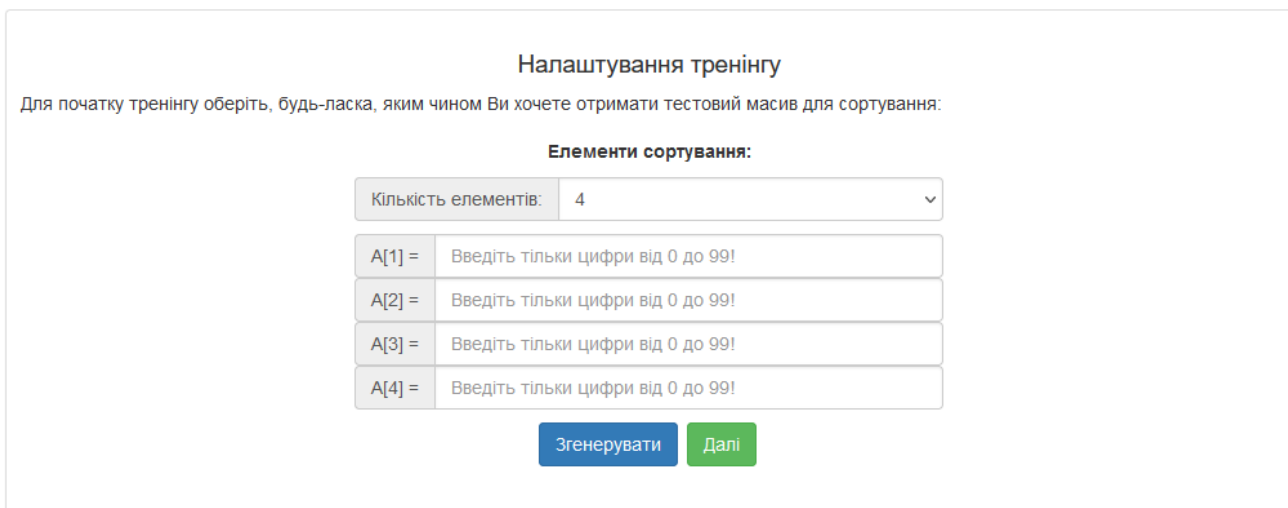


Рисунок 2.6 – Налаштування тренінгу

Після натиснення кнопки «Далі», нас переносить у вікно питань (рис. 2.7) у якому ми бачимо наступну інформацію: Алгоритм сортування; Вхідні дані, як масиву на старті так і у поточному стані; Запис дій; На якому ми питанні і скільки ще залишилося; Кількість допущених помилок; Вибір функції і який крок мизараз виконуємо; Кнопка для перевірки кроку.

Алгоритм сортування:

```

QuickSORT(A,p,r)
1.  if p < r
2.  then q ← PARTITION(A,p,r)
3.      QuickSort(A,p,q-1)
4.      QuickSort(A,q+1,r)

PARTITION(A,p,r)
1.  x ← A[r]
2.  i ← p - 1
3.  for j ← p, to r - 1
4.      do if A[j] ≤ x
5.          then i ← i + 1
6.              swap A[i] ↔ A[j]
7.  swap A[i+1] ↔ A[r]
8.  return i + 1

```

Вхідні дані:

Елементи масиву A[i]:

1	2	3	4
---	---	---	---

Поточний стан масиву A[i]:

1	2	3	4
---	---	---	---

Лог дій:

```

1 if 1 < 4 // TRUE

```

Питання: 3 з 111

Помилки: 0

Вкажіть функцію та крок алгоритму, яка буде виконуватися:
Оберіть ім'я функції з списку та вкажіть крок
 Функція , крок , або

Рисунок 2.7 – Вікно завдання

Тренажер, створений студентом Голубенко Владиславом, який написаний на мові програмування Java.

На першій сторінці (рис. 2.8) ми можемо помітити наступну інформацію: Задача; Алгоритм з яким нам потрібно працювати; Перехід до технічного матеріалу; Кнопка для запуску тренажера.

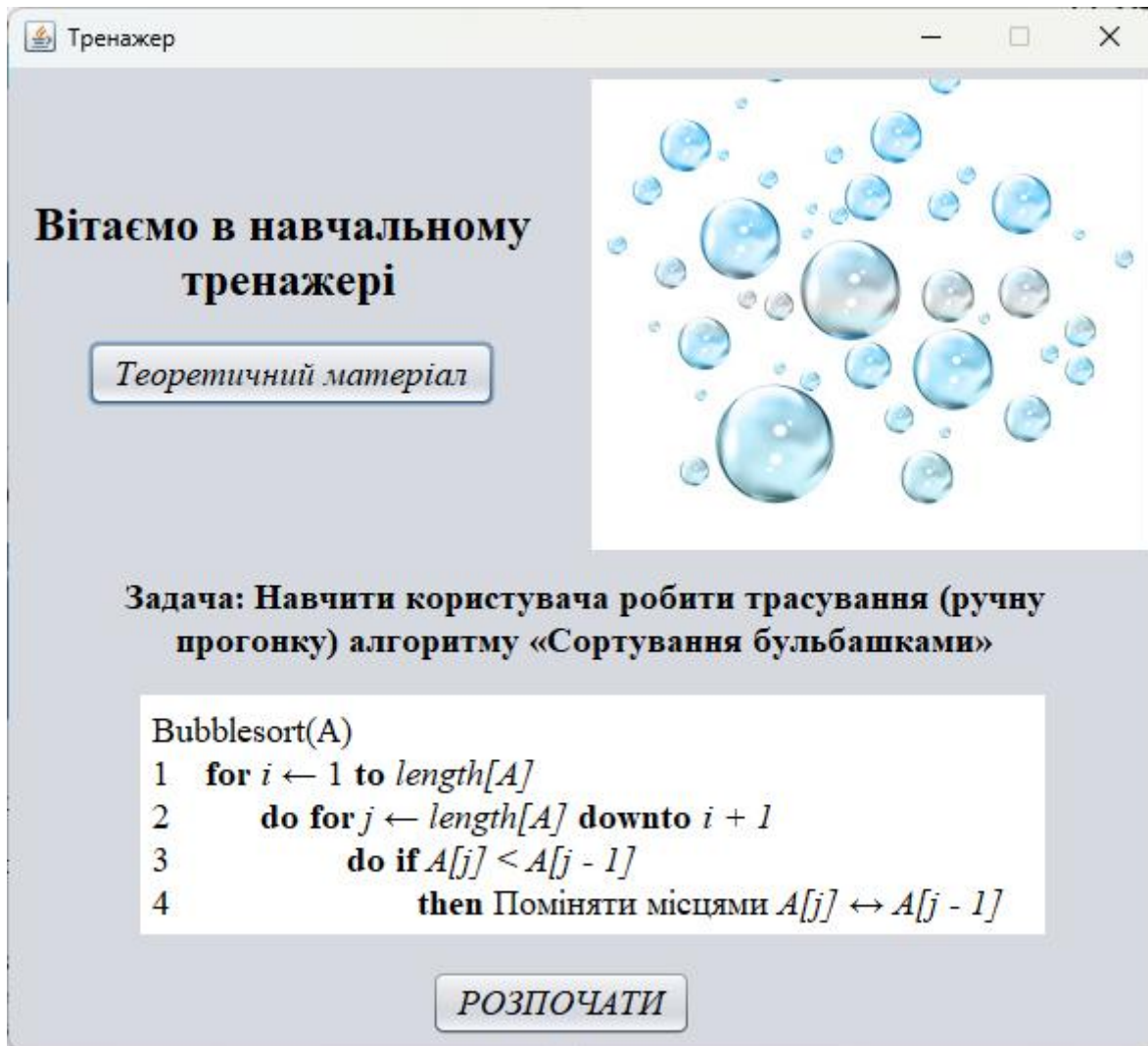


Рисунок 2.8 – Вступне вікно

Після того як ми розпочинаємо тестування, програма автоматично генерує нам масив.

У цьому тренажері ми повині визначити, як буде крокувати наш алгоритм сортування (рис. 2.9), для цього у нас є наступна інформація: Вхідний масив; Алгоритм сортування; Питання; Наші дії; Стан масиву; Кнопка перевірки кроку.

У разі неправильної відповіді, програма дасть підказку. (рис. 2.10)

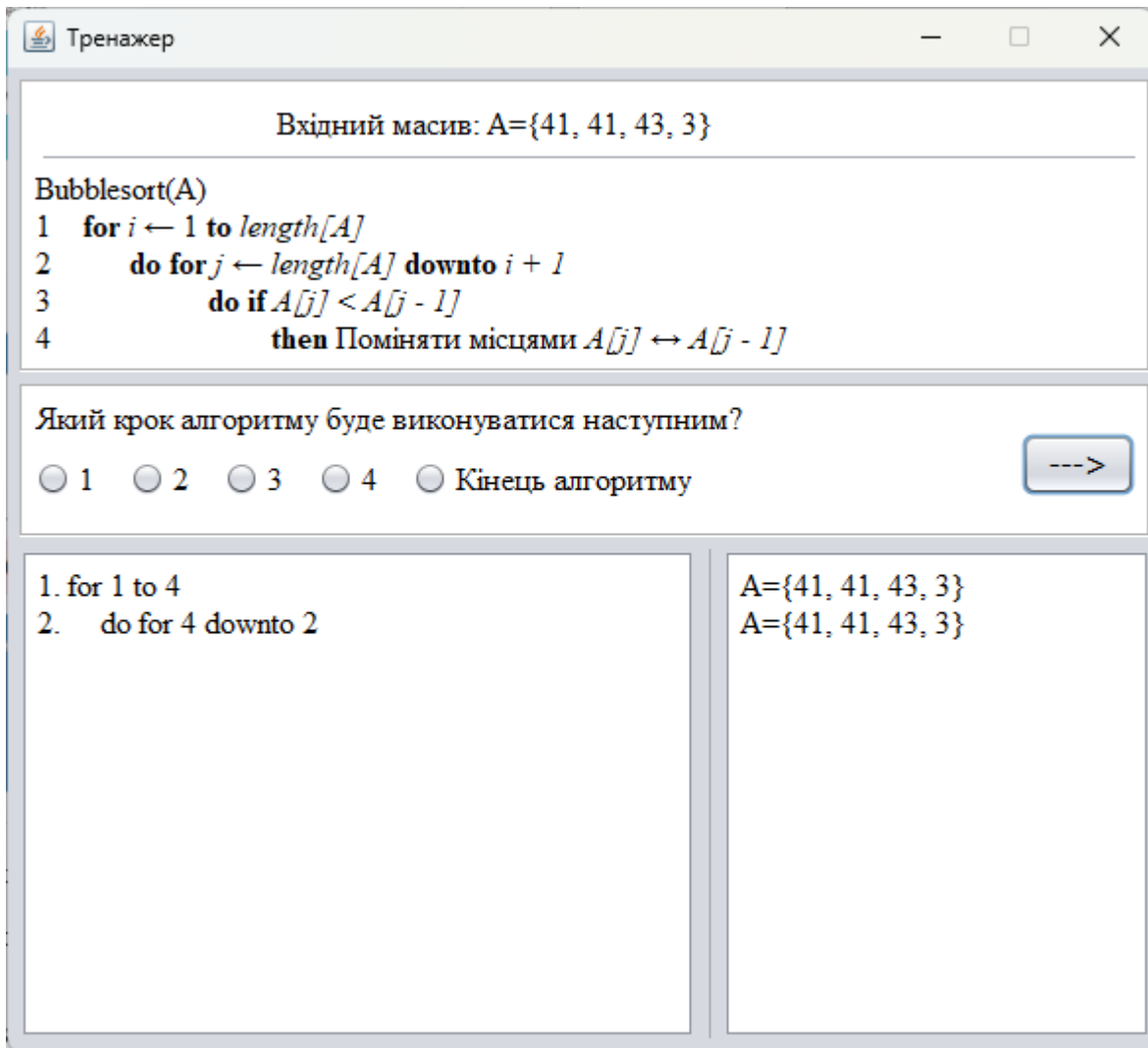


Рисунок 2.9 – Вікно з питаннями

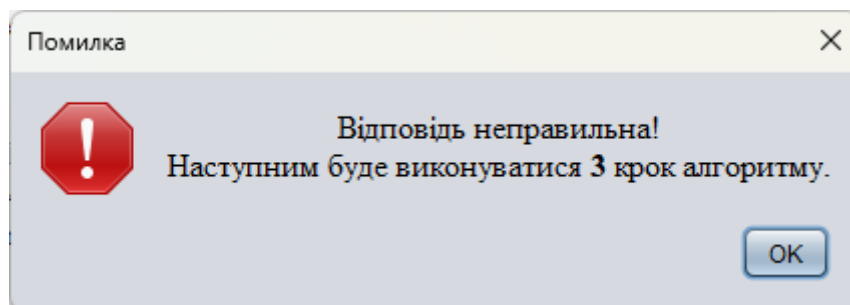


Рисунок 2.10 – Сповіщення у разі помилки

2.3. Позитивні аспекти оглянутих програмних продуктів

Розглянуті програми мають такі позитивні аспекти:

1. У тренажері 2 та 3 передбачено ознайомлення з теоретичним матеріалом, тому вони допомагають підготуватися до перевірки тренажером та надають змогу одразу знайти потрібну інформацію.

2. Усі програми мають підказки для розв'язку завдання в випадку неправильної відповіді, що дає змогу користувачу отримати причину неправильної відповіді одразу.
3. Всі навчальні інструменти мають зрозумілий інтерфейс.
4. У 2 та 3 тренажері чудове оформлення.
5. Також у 2 та 3 тренажерах є можливість побачити на якому етапі користувач зараз знаходиться.
6. У 2 та 3 програмах, присутня випадкова генерація массивів, що дає гарантію що кожне наступне проходження буде відрізнятися від попереднього.

2.4. Негативні аспекти оглянутих програмних продуктів

Розглянуті тренажери мають такі негативні аспекти:

1. У 1 та 2 програмі відсутні теоритичні частини.
2. Для запуску другої програми, потрібно мати доступ у мережу.
3. Для запуску третьої програми, потрібно встановити програмне забезпечення Java.
4. У третьому тренажері, у програмі не вказаний автор.
5. У випадку генерації великого масиву, 2 та 3 тренажер можуть забрати багато часу у користувача.

2.5. Огляд засобів для створення навчальних симуляторів

Навчальні симулятори в сучасному освітньому середовищі відіграють важливу роль у наданні студентам інтерактивного та ефективного досвіду навчання. Засоби для створення таких симуляторів є ключовим елементом цього процесу, сприяючи розвитку різних навичок та набуттю практичного досвіду у різних галузях. В даному огляді розглянемо різні інструменти та підходи до створення навчальних симуляторів, їхні переваги та обмеження.

Інтерактивні віртуальні середовища є одним із ключових напрямків розвитку навчальних симуляторів. Такі середовища дозволяють створювати реалістичні віртуальні сценарії, в яких студенти можуть відтворювати різні ситуації та взаємодіяти з ними. Одним із популярних інструментів у цьому напрямку є Unity, який забезпечує розширені можливості для розробки 3D-симуляторів з великою кількістю готових ресурсів.

Переваги таких інтерактивних віртуальних середовищ включають реалістичність, можливість відтворення різних умов та сценаріїв. Однак ці інструменти також можуть бути складними у використанні, вимагаючи від розробників певного рівня експертизи.

Електронні дидактичні платформи є ще однією групою інструментів для створення навчальних симуляторів, спрямованих на інтерактивне навчання. Такі платформи, як Articulate Storyline чи Adobe Captivate, дозволяють легко створювати інтерактивні уроки та симуляції без необхідності в глибоких технічних знаннях.

Головною перевагою електронних дидактичних платформ є їхній інтуїтивно зрозумілий інтерфейс та можливість швидко створювати навчальні матеріали. Однак у порівнянні з віртуальними середовищами, вони можуть обмежувати можливості візуалізації та реалістичності.

Симулятори в реальному часі є ефективними інструментами для навчання та тренування у реальних умовах. Наприклад, програмне забезпечення для авіасимуляторів, такі як X-Plane чи Microsoft Flight Simulator, дозволяють тренувати навички пілотування в реалістичних умовах.

Ці інструменти надають велику реалістичність та імерсивність, що є важливим для тренувань у високотехнологічних галузях. Проте, їхнє використання може бути дорогим та вимагати спеціального обладнання.

Створення навчальних симуляторів вимагає ретельного вибору інструментів залежно від конкретних цілей та вимог. Від інтерактивних віртуальних середовищ до електронних дидактичних платформ та симуляторів в реальному часі, кожен інструмент має свої переваги та обмеження. Інноваційність у цьому напрямку полягає в поєднанні різних інструментів для створення комплексних та ефективних навчальних досвідів. Залежно від потреб та можливостей, вчителі та розробники можуть вибрати оптимальний інструмент для реалізації своїх навчальних концепцій.

РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Огляд матеріалу з теми «Робота з діапазонами (Range) в Excel VBA»

Один із важливих аспектів роботи з пакетом Microsoft Office, зокрема з Excel, - це ознайомлення з середовищем розробки VBA. Visual Basic for Application є мовою програмування, створеною Microsoft, і використовується для автоматизації операцій в програмах пакета MS Office, таких як Access, Word та Excel. В цьому огляді розглянемо основні аспекти VBA та його застосування у роботі з Excel.

VBA є мовою програмування, спеціально адаптованою для роботи з продуктами MS Office. Вона використовується для автоматизації завдань та створення скриптів для підвищення ефективності роботи з документами, таблицями та базами даних. Однією з ключових переваг VBA є його простота в освоєнні, що дозволяє навіть початківцям швидко вчитися та застосовувати цю мову.

Однією з головних областей застосування VBA є автоматизація операцій з комітками в Excel. Ви можете створювати скрипти для переміщення, додавання та сортування даних в таблицях. Також VBA дозволяє програмувати вивід повідомлень, створювати власні функції та формули, що розширює можливості роботи з Excel.

Основний принцип роботи з VBA полягає в оперуванні об'єктами, що відносяться до об'єктно-орієнтованого програмування. Один з найбільш базових об'єктів - це Range (діапазон), який використовується для маніпулювання комітками. Властивості Range дозволяють вам змінювати значення, формули, вибирати рядки чи стовпці, а методи - виконувати операції копіювання, видалення та інші.

Основні властивості Range включають:

- Value: встановлює або отримує значення діапазону.
- Formula: дозволяє використовувати формули у виділених комітках.
- Row та Columns: вибирають рядки та стовпці відповідно.

- Cells: представляє клітинки у вказаному діапазоні.

Основні методи Range включають:

- Copy: копіює діапазон до вказаного діапазону чи до буфера обміну.
- Clear: очищає весь об'єкт.
- Cut: вирізання та вставлення об'єкта до буфера обміну або в указане місце призначення.

VBA - потужний інструмент для автоматизації операцій в MS Office, зокрема в Excel. Здатність працювати з об'єктами та використовувати їхні властивості та методи дозволяє користувачам створювати ефективні та індивідуалізовані рішення для роботи з даними та документами. Ознайомлення з основами VBA відкриває широкі можливості для оптимізації робочих процесів та автоматизації рутинних завдань.

3.2. Алгоритм роботи тренажера

Коли користувач запускає тренажер, то перед ним з'являється наступна інформація: Тема на основі якій був створений тренажер, Кнопка «Перейти до тестування» та «Перейти до теорії», П.І.Б та група того хто розробив тренажер та П.І.Б керівника курсової роботи.

Після натискання кнопки користувача перенесе на наступний етап, у якому будуть присутні запитання з декількома варіантами відповіді, але серед них лише один правильний варіант. У разі натискання на неправильний варіант, програма повідомить користувачу, що він вибрав неправильний варіант і автоматично вкаже на правильну відповідь. Також після натиску на будь яку кнопку, стане доступна кнопка теорії, яка поясне правильну відповідь.

А тепер розглянемо запитання які будуть зустрічатися у тренажері.

Питання 1 Діапазон(Range) це?

- 1: Об'єкт.
- 2: Книга.
- 3: Аркуш.
- 4: Властивість.

Правильна відповідь: 1.

Питання 2: Для чого існує Діапазон?

- 1: Для маніпулювання усією роботою.
- 2: Для створення нових клітинок.
- 3: Для змінення параметрів аркуша.
- 4: Для маніпулювання комірками.

Правильна відповідь: 4.

Питання 3: Чи можна завдяки Діапазону виділяти стовпці та рядки?

- 1: Тільки рядки.

2: Тільки стовбці.

3: Так можна.

4: Ні, не можна.

Правильна відповідь: 3.

Питання 4: Що робить метод Select?

1: Вибирає об'єкт.

2: Видаляє об'єкт.

3: Створює об'єкт.

4: Активує об'єкт.

Правильна відповідь: 1.

Питання 5: За допомогою якої властивості можливо використати формулу в комірках?

1: Method.

2: Formula.

3: Spec.

4: Vector.

Правильна відповідь: 2.

Питання 6: Що зайве у цьому коді? Range ("A1:A2") .Select End sub

1: Select.

2: Range.

3: End sub.

4: Нічого.

Правильна відповідь: 3.

Питання 7: За допомогою якого методу можливо використати формули у комірках?

1: Method.

2: Formula.

3: SPEC.

4: Vector.

Правильна відповідь: 2.

Питання 8: У чому різниця між Cells та Range?

1: Немає.

2: Cells вибирає лише вказані клітинки.

3: Різні типи.

4: По різному впливають на вказану клітинку.

Правильна відповідь: 2.

Питання 9: Для чого існує властивість Value?

1: Для задавання комірці властивості.

2: Для створення нової властивості.

3: Для виведення тексту.

4: Це метод.

Правильна відповідь: 1.

Питання 10: Навіщо існує метод Activate?

1: Робить клітинку прозорою.

- 2: Деактивує клітинку.
 - 3: Активує клітинку.
 - 4: Переносить на наступну сторінку.
- Правильна відповідь: 3.

3.3 Блок-схема роботи тренажеру

Перед тим як розпочати програмування, була створена блок-схема для загального виконання програми (зображена на рис. 3.1), а також блок-схема для визначення роботи практичної частини тренажера (представлена на рис. 3.2).

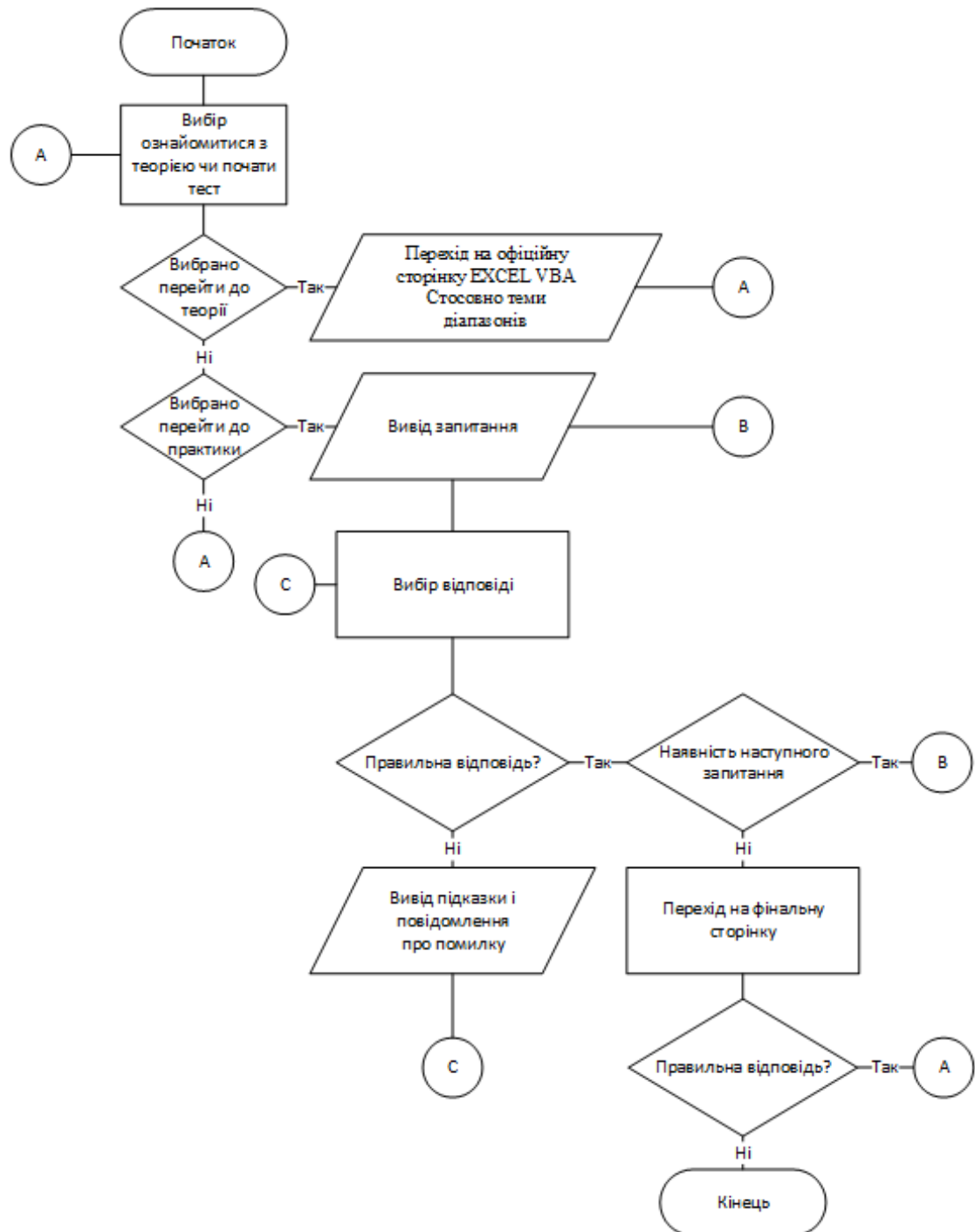


Рисунок 3.1 Блок-схема загальної роботи тренажера

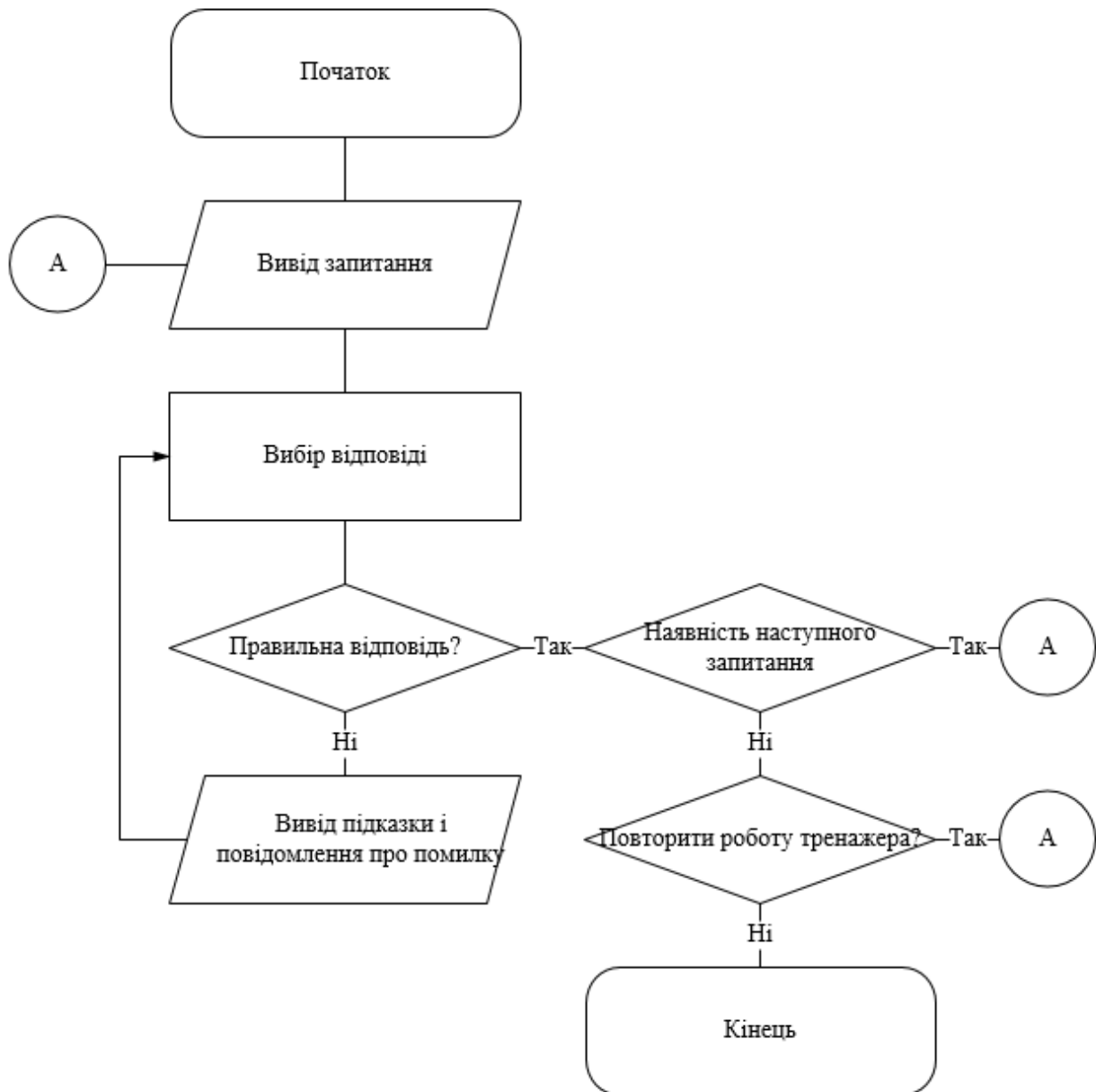


Рисунок 3.2-Блок-схема практичної частини тренажера

РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

4.1 Обґрунтування вибору мови програмування

Створення тренажеру - це завдання, що передбачає вибір платформи та інструментів для розробки. Один із популярних виборів для створення графічних інтерфейсів в Python - Tkinter. У цьому тексті ми розглянемо причини вибору Python Tkinter для реалізації тренажеру.

1. Простота та Доступність

Tkinter входить до стандартної бібліотеки Python, що робить його дуже доступним для користувачів. Він не вимагає окремого встановлення або налаштувань, оскільки є частиною стандартного пакета. Це дозволяє вам швидко розпочати роботу над проектом, не втрачаючи часу на конфігурацію зовнішніх бібліотек чи інструментів.

2. Легкість Використання

Tkinter славиться своєю легкістю використання. Він має простий та зрозумілий синтаксис, що робить його ідеальним вибором для початківців та тих, хто швидко хоче отримати результат. Створення графічних інтерфейсів у Tkinter вимагає мінімум коду, що полегшує розробку.

3. Крос-платформеність

Tkinter є крос-платформеним інструментом, що означає, що програми, створені з його використанням, можуть запускатися на різних операційних системах без будь-яких значущих модифікацій. Це важливо, оскільки забезпечує велику гнучкість та розповсюдження вашого тренажеру серед різних користувачів.

4. Активна Підтримка та Спільнота

Tkinter є добре встановленою та широко використовуваною технологією в спільноті Python. Це означає, що ви можете легко знайти відповіді на свої питання, навіть якщо вони стосуються складних аспектів інтерфейсу. Крім того, активність спільноти сприяє постійному вдосконаленню та оновленню Tkinter.

5. Інтеграція з Python

Однією з ключових переваг Tkinter є його повна інтеграція з Python. Ви можете легко використовувати всі функціональні можливості мови програмування для реалізації різних завдань у вашому тренажері. Це забезпечує гнучкість та розширює можливості розробки.

6. Розширюваність

Tkinter дозволяє вам легко розширювати функціональність вашого тренажеру за допомогою різноманітних віджетів та бібліотек, що підтримуються. Ви можете створювати складні та інтерактивні інтерфейси, додаючи нові елементи та можливості.

Обираючи Tkinter для створення тренажеру, ви обираєте простоту, доступність та гнучкість. Його крос-платформенність, легкість використання та активна підтримка роблять його привабливим вибором для розробників, які шукають ефективні та швидкі рішення у сфері створення графічних інтерфейсів для своїх проектів. Tkinter дозволяє вам фокусуватися на важливих аспектах вашого тренажеру, забезпечуючи при цьому надійність та продуктивність вашого розвитку.

4.2 Обґрунтування вибору програмних засобів

В сучасному світі програмування розробники мають великий вибір інструментів для роботи. Однак Visual Studio Code (VS Code) став популярним та широко використовуваним середовищем для розробки з численними перевагами. У цьому тексті ми розглянемо, чому Visual Studio Code є чудовим вибором для розробників та як він полегшує роботу та поліпшує продуктивність.

1. Безкоштовність та Відкритий Код

Visual Studio Code випускається під ліцензією MIT та є відкритим програмним забезпеченням. Це означає, що ви можете завантажити, використовувати, модифікувати та розповсюджувати його абсолютно

безкоштовно. Такий підхід дозволяє розробникам вносити власні внески та допомагає спільноті активно вдосконалювати інструмент.

2. Розширюваність та Екосистема Розширень

VS Code надає потужну систему розширень, що дозволяє розробникам налаштовувати та розширювати функціональність за допомогою різноманітних плагінів. Завдяки цій розширюваності, кожен може адаптувати середовище роботи до своїх власних потреб та вподобань.

3. Крос-платформенність

Однією з важливих переваг VS Code є його крос-платформенність. Він підтримує операційні системи Windows, macOS та Linux, що дає розробникам можливість працювати в однаковому середовищі незалежно від операційної системи.

4. Легкість Використання та Мінімалістичний Дизайн

VS Code визначається своєю легкістю використання та чистим, мінімалістичним дизайном. Його інтерфейс є інтуїтивно зрозумілим, що полегшує навігацію та швидкість освоєння новими користувачами.

5. Вбудовані Інструменти для Керування Версіями

VS Code має вбудовану підтримку систем керування версіями, таких як Git. Це полегшує відстеження змін у коді та спільну роботу у команді.

6. Підтримка Різних Мов Програмування

Ще однією вагомою перевагою є широка підтримка різних мов програмування. VS Code надає розширення для практично будь-якої мови, включаючи Python, JavaScript, Java, C++, і багато інших.

7. Автоматичне Доповнення та Виділення Синтаксису

VS Code володіє потужними засобами автоматичного доповнення коду та виділення синтаксису. Це робить процес написання коду більш продуктивним та допомагає у виявленні помилок на ранніх етапах розробки.

8. Інтеграція з Іншими Інструментами та Сервісами

VS Code легко інтегрується з різноманітними сервісами, такими як GitHub, Docker, Azure, що полегшує розробку та взаємодію з іншими інструментами розробки.

4.3 Опис програмної реалізації тренажеру

Мій вибір пав на «Python», тому що ця мова, дуже проста та її програми займають не багато місця.

Для створення візуальної частини, я використовував бібліотеку «Tkinter» Для того щоб підєднати цю бібліотеку, потрібно ввести наступну команду:

```
import tkinter as tk
```

Код реалізація вікна:

```
root.title("Test Range")
root.geometry("640x480")
root.resizable(width=False, height=False)
font_main = ("Times New Roman", 20)
font_test_button = ("Calibri", 16)
```

На початковій сторінці (рис.4.1) розміщено тема яка буде у цьому тренажері, кнопка для початку тесту та кнопка для переходу на офіційний сайт «Microsoft» ініціали того хто розробив цей тренажер, та його керівника.

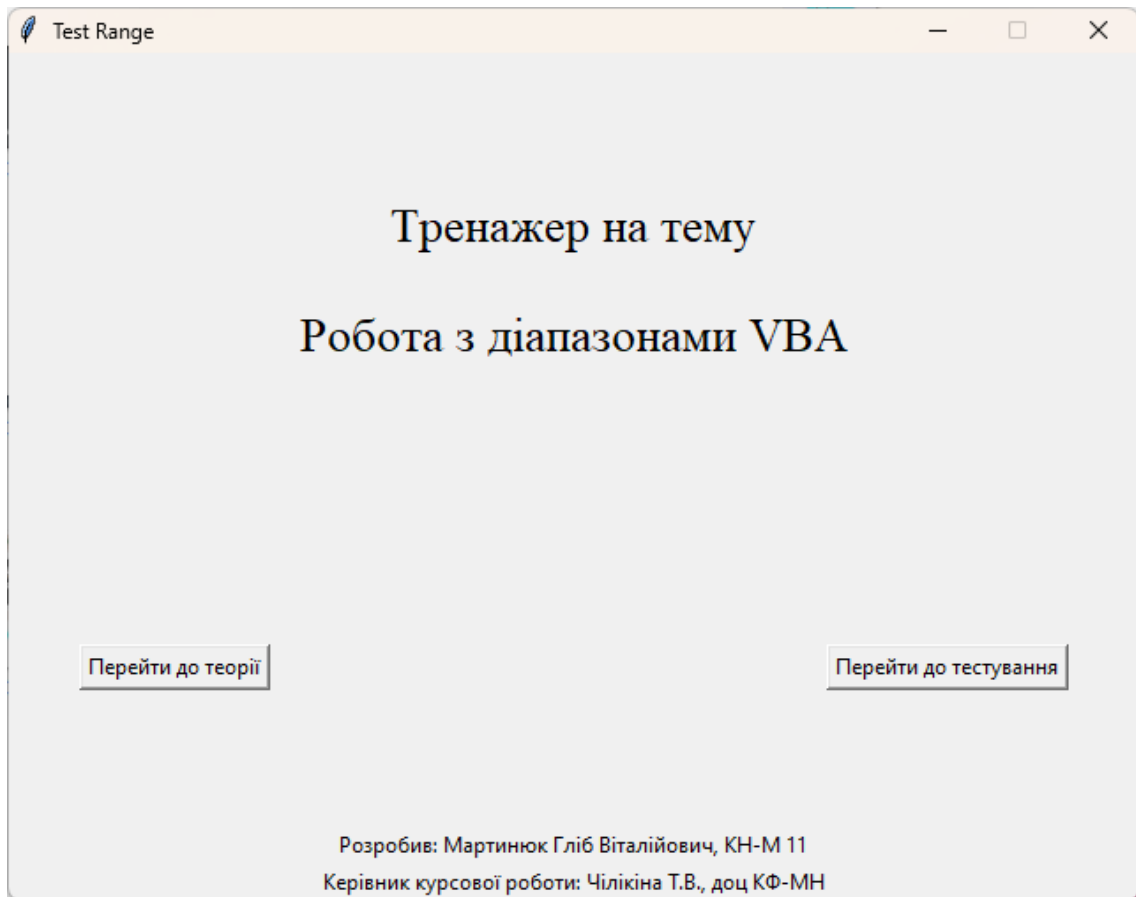


Рисунок 4.1 Початкова сторінка

Якщо натиснути на кнопку «Перейти до теорії», то відкриється браузер на офіційній сторінці Майкрософт. (рис.4.2)

Для можливості переходу на сайт з програми було впроваджено бібліотеку «webbrowser». Код реалізації кнопки теорії:

```
button_theory = tk.Button(root, text= "Перейти до теорії",command =  
lambda:webbrowser.open("https://learn.microsoft.com/en-  
us/office/vba/api/excel.range(object")))
```

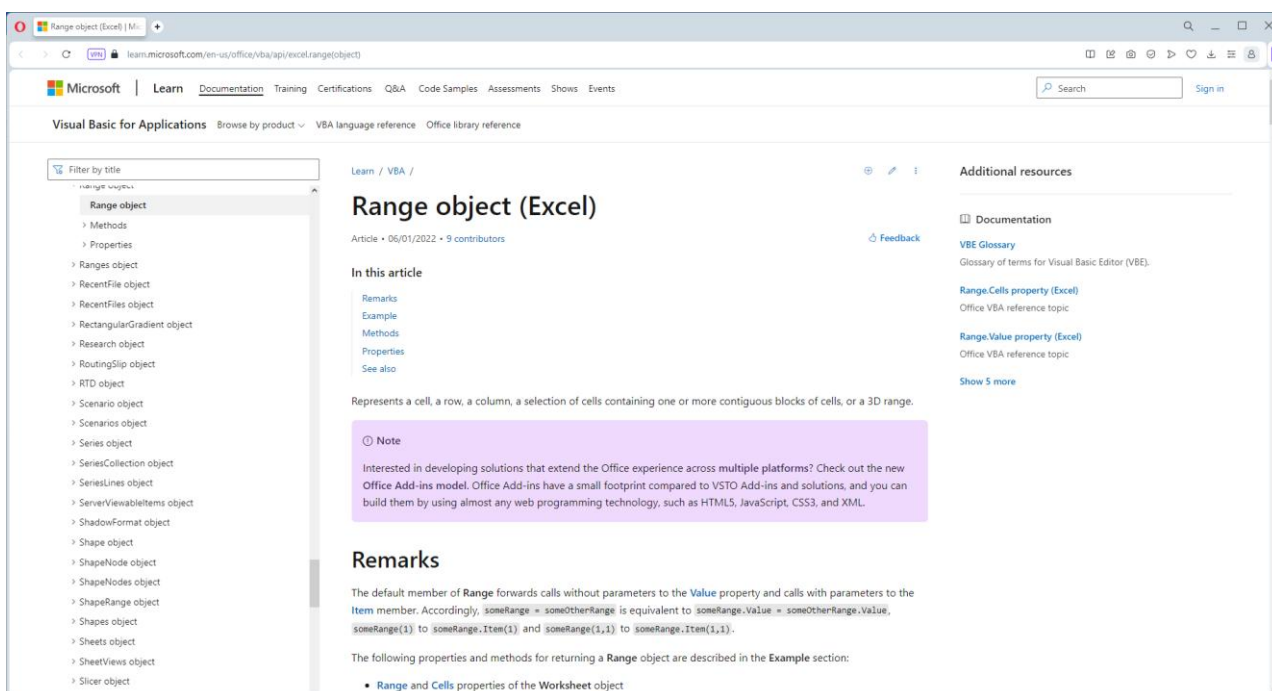


Рисунок 4.2 Офіційна документація «Range Excel VBA» На сайті майкрософт
Якщо вибрати клавішу «Перейти до тестування» то нас перенесе на (Рис 4.3) де
ми можемо побачити наступну інформацію: Питання; Варіанти відповідей;

Індекс запитання;

Програма розуміє, яке запитання виводи на екран завдяки віджету
етикетка, який в свою чергу бере інформацію зі списку який у собі має питання.
Код реалізації етикетки:

```
question_label = tk.Label(font= font_main, text=questions[quest]["question"])
```

В той час, як варіанти відповідей, використовують віджет кнопки і беруть
інформацію з того самого списку. Код реалізації кнопки 1:

```
next_button_1 = tk.Button(font= font_test_button,width = 40, height = 1,  
text=questions[quest]["choices"][0],command =  
lambda:check_button(questions[quest]["choices"][0]))
```



Рисунок 4.3 Вікно з питаннями

Для вибору відповіді, нам просто потрібно натиснути на один із варіантів.

Код реалізації перевірки запитання:

```
def check_button(answer):
    if answer == questions[quest]["correct_answer"]:
        question_label.config(text="Правильно")
        next_layer()
    else:
        question_label.config(text="Неправильно\n правильна відповідь
№ {}".format(right_quest_info(questions[quest]["choices"])))
        next_layer()
```

Якщо ви вибрали правильний варіант (Рис 4.4), то програма про це повідомить.

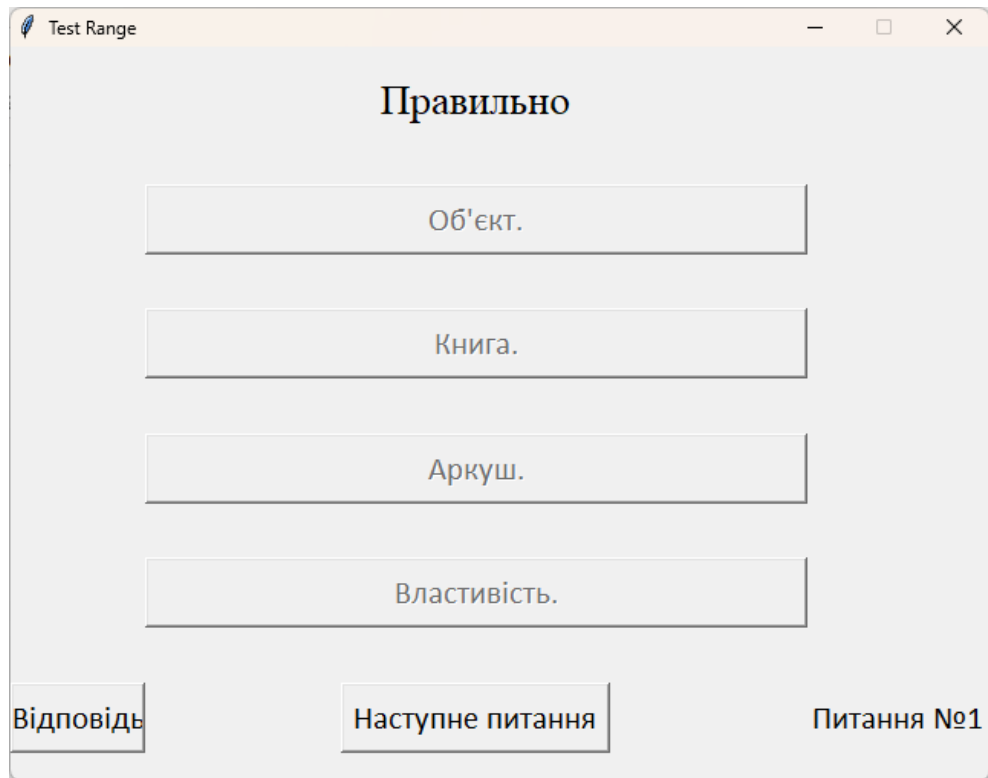


Рисунок 4.4 Вибрана правильна відповідь

Але якщо ви вибрали неправильну відповідь, то програма вам про це повідомить та вкаже вам на правильну відповідь (Рис 4.5).

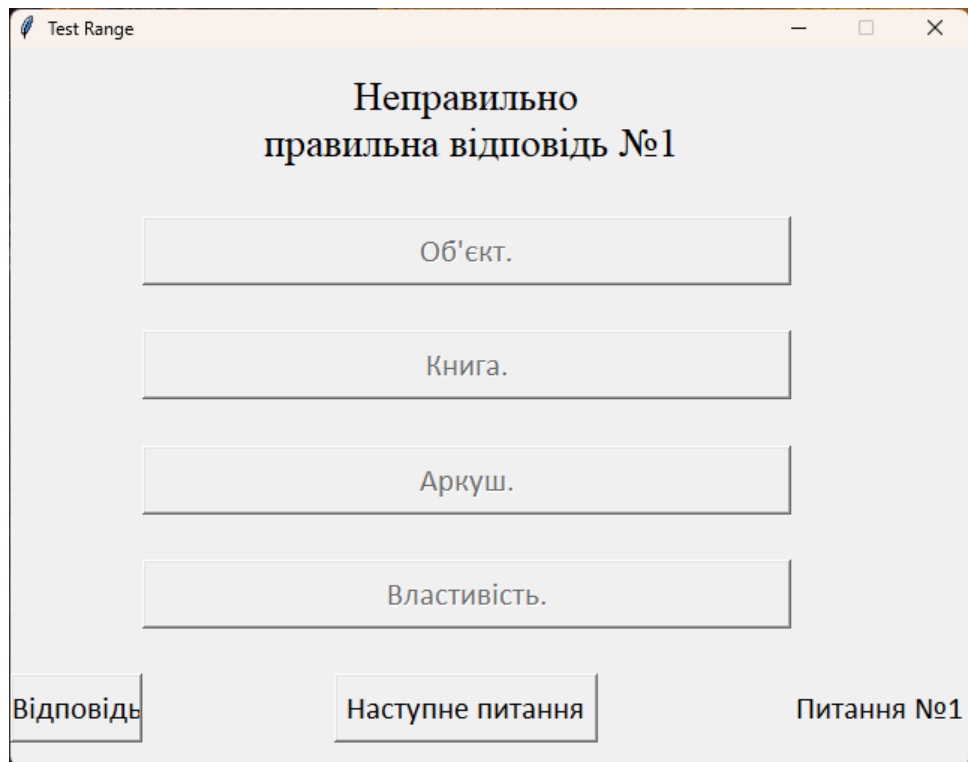


Рисунок 4.5 Вибір неправильної відповіді

У будь якого випадку, після вибору варіанту, нам стануть доступними дві кнопки «Відповідь» та «Наступне питання».

Якщо вибрати кнопку «Відповідь» то нам відкриється маленьке вікно, яке дасть теорію до питання питання (Рис 4.6).

Код реалізація пояснення відповіді:

```
def open_info():
    showinfo(title=f"Теорія до питання {quest+1}",
            message=questions[quest]["explanation"])
```

Якщо вибрати кнопку «Наступне питання» то нас перенесе на наступне питання у тренажері (Рис 4.7).

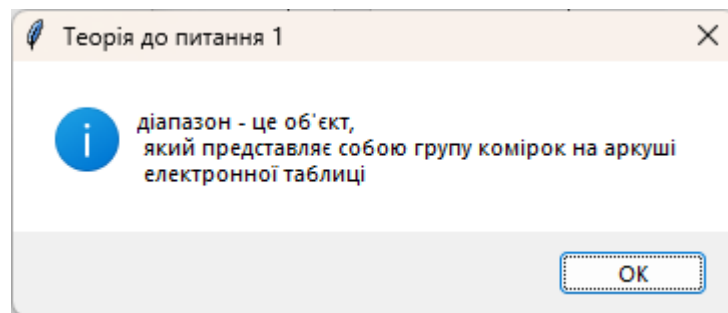


Рисунок 4.6 Теорія до питання

Код реалізація активації кнопок:

```
def next_layer():
    next_button_1.config(state='disabled')
    next_button_2.config(state='disabled')
    next_button_3.config(state='disabled')
    next_button_4.config(state='disabled')
    quest_btn.config(state='active')
    theory_btn.config(state="active")
```

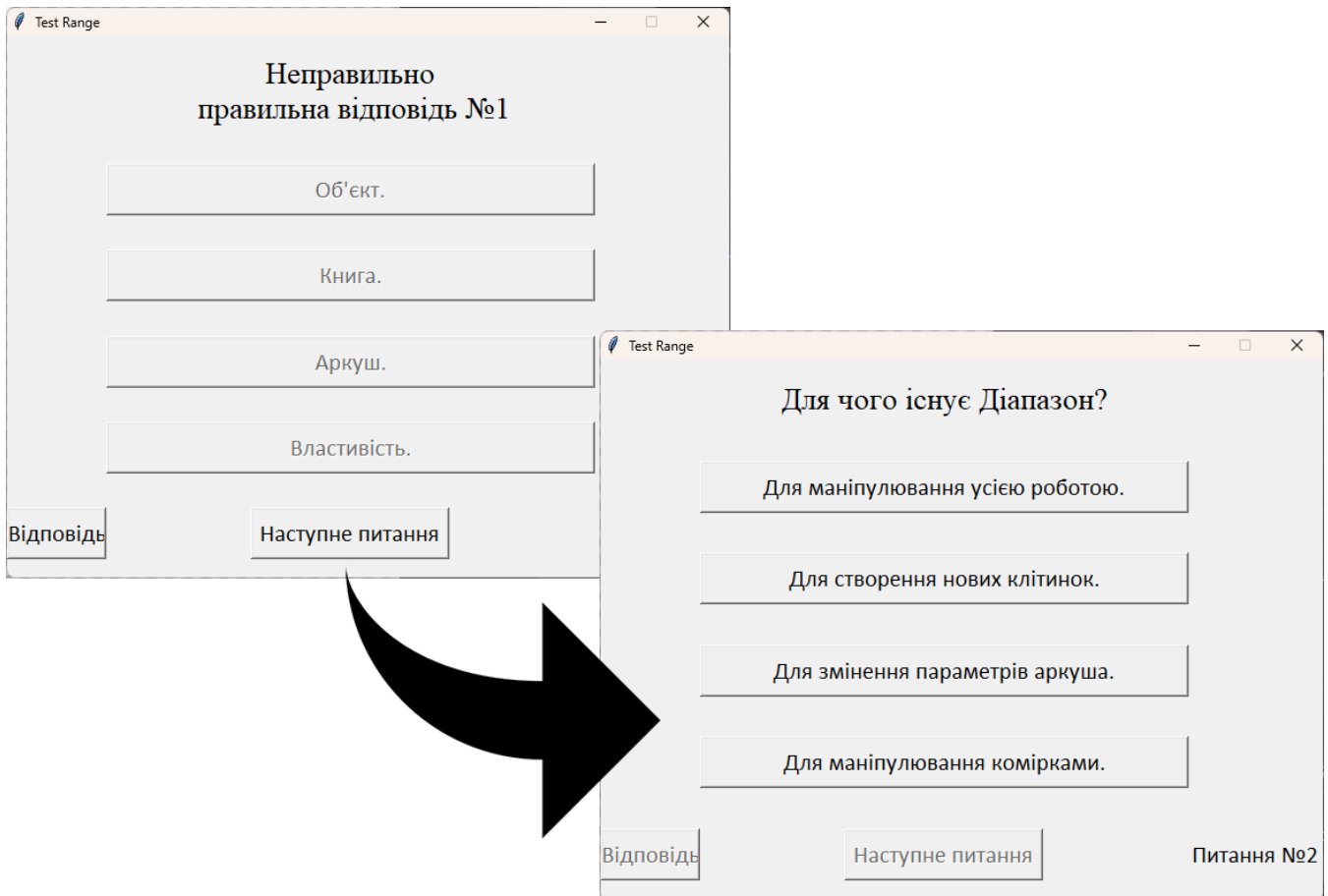


Рисунок 4.7 Приклад роботи кнопки «Наступне питання»

Після відповідання на всі питання які були у тренажері, програма покаже на скільки питань ми відповіли правильно з прешої спроби та запропонує перейти до головного меню (Рис 4.8)

Код реалізація підрахунку та вінального вікна: `def final(score):`

```

delete_all_widgets()
def label_score_correction():
    if score == 1:
        return "раз"
    if 1 < score < 5:
        return "рази"
    return "разів"
label_score = tk.Label(root, font= font_main ,text= f"Ви відповіли
правильно {score} {label_score_correction()}")

```

```
label_score.pack(expand=True , anchor="center")  
label_score = tk.Button(root, font= font_main ,text= "Повернутися до  
меню", command = main)  
label_score.pack(expand=True , anchor="center")
```

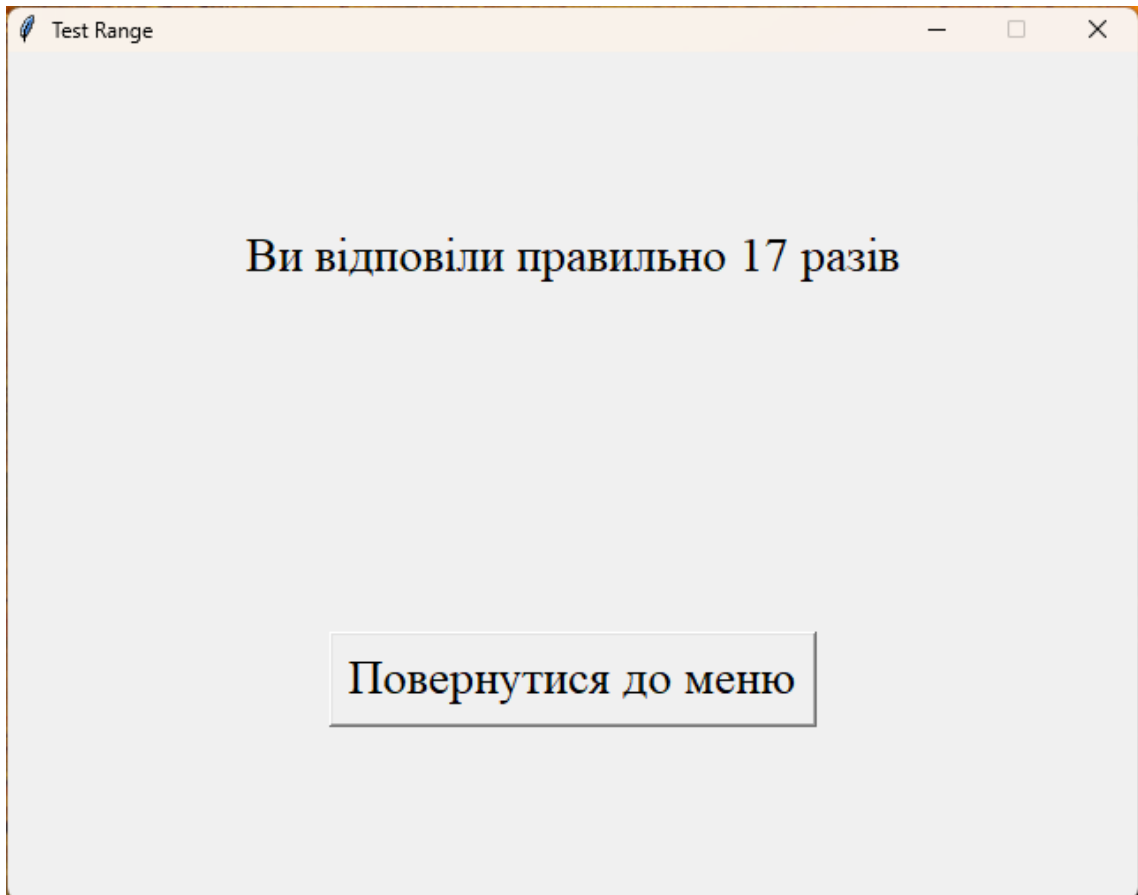


Рисунок 4.8 Фінальне вікно результатів.

Вихід із програми здійснюється завдяки натисканню на верхній правий хрестик.

4.4 Тестування створеного продукту

Тестування створеного продукту є важливим етапом в розробці, оскільки воно дозволяє визначити правильність обробки відповідей, ефективність реагування системи на різні ситуації та забезпечити коректну роботу на різних операційних системах.

У ході тестування було забезпечено правильну обробку відповідей на математичні питання. Реалізовано різноманітні сценарії для тестування реакції системи на різні ситуації, такі як введення неправильних відповідей, спроби обходу системи та інші.

Тренажер було протестовано на актуальних версіях операційних систем Windows, зокрема, Windows 10 та Windows 11. Слід зауважити, що продукт був розроблений виключно для операційних систем Windows, і тому тестування на мобільних пристроях, зокрема телефонах, не проводилося.

В ході тестування інтерфейсу були оцінені зручність та естетика користувацького інтерфейсу. Також було перевірено легкість навігації та взаємодії з елементами тренажера. Зауважено, що інтерфейс повинен бути не тільки естетичним, але й інтуїтивно зрозумілим для користувачів.

Продуктивність тренажера була виміряна в часі реакції на дії користувача, а також виявлення можливих затримок в роботі. Важливо, щоб продукт працював ефективно та швидко, забезпечуючи комфорт користувачам.

Під час загальної перевірки придатності були враховані відгуки користувачів та їхні враження від використання тренажера. Здобуті відгуки допомагають визначити, наскільки продукт відповідає вимогам освітнього середовища та чи відповідає він очікуванням користувачів.

Під час тестування було ідентифіковано та виправлено виявлені

ВИСНОВКИ

Реалізовано програмне забезпечення для тренажера з теми «Робота з діапазонами (Range) в Excel VBA» англomовного дистанційного навчального курсу «Організація та обробка електронної інформації»

Під час виконання дипломної роботи були виконані наступні завдання:

- сформульовано постановку задачі,
- проаналізована література, що стосується дистанційного навчання(дистанційних курсів),
- проаналізована література, що стосується використання «Роботи з діапазонами»,
- проведено інформаційний аналіз,
- проведений інформаційний огляд, ознайомлення зі схожими тренажерами,
- досліджені теоретичні відомості щодо теми,
- складений алгоритм роботи тренажера,
- створено блок-схеми роботи складеного алгоритму,
- розроблений тренажер у обраному середовищі,
- програма протестована на наявність помилок,

Основні вимоги до тренажера виконані:

- наявність блоку з теорією,
- можливість змінити мову тренажера, розробка простого та зрозумілого інтересу,
- сумісність середовища розробки з системою дистанційного навчання
- можливість зручної взаємодії з тренажером,
- створення алгоритму для перевірки відповідей, у разі помилки

повідомити користувача про неї.

Результати дипломної роботи опубліковані у Збірнику наукових статей магістрів Вищого навчального закладу Укоопспілки «Полтавський університет економіки і торгівлі»

СПИСОК ЛІТЕРАТИ

1. Гаркуша С. В. Методичні рекомендації щодо виконання кваліфікаційної роботи студентів спеціальності 122 Комп'ютерні науки освітня програма «Комп'ютерні науки» ступеня магістра / С. В. Гаркуша, О. В. Ольховська, О. О. Черненко. – Полтава : ПУЕТ, 2023. – 68 с.
2. Мельник Ю.В., Бороденко Н.Д., Богданова Н.В. Деякі проблеми організації дистанційного навчання в ВНЗ. URL: <https://2017.moodlemoot.in.ua/course/view.php?id=114>
3. Ткачук Г.В. Теоретичні аспекти та стан впровадження змішаного навчання у закладах вищої освіти України. URL: <https://dspace.udpu.edu.ua/handle/6789/8666>
4. Хоукінг, С. (2016). "Excel VBA: Step-By-Step Guide To Learning Excel Programming Language For Beginners." Independently published. URL: <https://www.simplilearn.com/tutorials/excel-tutorial/excel-vba>
5. Visual Basic for Applications [Електронний ресурс] / Матеріал з Вікіпедії вільної енциклопедії. URL: https://uk.wikipedia.org/wiki/Visual_Basic_for_Applications
6. Луковець І. О. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТРЕНАЖЕРУ З ТЕМИ «ПРОТОКОЛИ КОМП'ЮТЕРНИХ МЕРЕЖ» ДИСТАНЦІЙНОГО КУРСУ «ІНФОРМАЦІЙНІ МЕРЕЖІ». 2023. URL: <http://dspace.puet.edu.ua/handle/123456789/13069>
7. ДСТУ ГОСТ 7.1:2006 Система стандартів з інформації, бібліотечної та видавничої справи. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання / Нац. стандарт України. Вид. офіц. – [Чинний від 2007-07-01]. Київ : Держспоживстандарт України, 2007. 47 с
8. Алексов С. В. СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТРЕНАЖЕРУ «МЕТОД РЕЗОЛЮЦІЙ» ДИСТАНЦІЙНОГО КУРСУ «МАТЕМАТИЧНА ЛОГІКА». 2021. URL: <http://dspace.puet.edu.ua/handle/123456789/8998>
9. Любовецький Т. О. Алгоритмізація та програмування тренажера «Класифікація граматик, приклади використання» дистанційного

навчального курсу «Теорія програмування». 2021. URL:
<http://dspace.puet.edu.ua/handle/123456789/11374>

10. Стельнік А. І. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТРЕНАЖЕРУ З ТЕМИ «ПРОТОКОЛИ КОМП'ЮТЕРНИХ МЕРЕЖ» ДИСТАНЦІЙНОГО КУРСУ «ІНФОРМАЦІЙНІ МЕРЕЖІ». 2021. URL:
<http://dspace.puet.edu.ua/handle/123456789/10306>

ДОДАТОК А. ДИСК З МАТЕРІАЛАМИ