

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

**«АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ТРЕНАЖЕРА З
ТЕМИ «ЦИКЛИ EXCEL VBA» ДИСТАНЦІЙНОГО НАВЧАЛЬНОГО
КУРСУ «ОРГАНІЗАЦІЯ ТА ОБРОБКА ЕЛЕКТРОННОЇ ІНФОРМАЦІЇ»**

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

Виконавець роботи Малишевський Андрій Олександрович

_____«____» _____ 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

2022р.

**ЗАВДАННЯ І КАЛЕНДАРНИЙ ГРАФІК ВИКОНАННЯ
КВАЛІФІКАЦІЙНОЇ РОБОТИ**

на тему «Алгоритмізація та програмна реалізація тренажера з теми «Цикли Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації»»

зі спеціальності 122 Комп'ютерні науки

освітня програма «Комп'ютерні науки» ступеня
бакалавр

Прізвище, ім'я, по батькові Малишевський Андрій Олександрович

Затверджена наказом ректора № 144-Н від «1» вересня 2022р.

Термін подання студентом роботи «__» _____202_р.

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

Зміст пояснювальної записки

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

2.1. Огляд робіт зі схожим завданням

2.2. Позитивні аспекти розглянутих робіт

2.3. Вади розробок розглянутих робіт

2.4. Актуальність розробки тренажера

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Теоретичні матеріали

3.1.1. Цикл For ... Next

3.1.2. Цикл For Each ... Next

3.1.3. Цикл Do While ... Loop

3.1.4. Цикл Do Until ... Loop

3.2. Алгоритм роботи тренажеру

3.3. Блок-схема програми-тренажера

4. ПРАКТИЧНА ЧАСТИНА

4.1 Опис програмного забезпечення

4.2 Методи розробки та тестування програмного забезпечення

ВИСНОВОК

СПИСОК ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

ДОДАТОК А.

Перелік графічного матеріалу: 2 аркуш блок-схем, 19 ілюстрацій.

Консультанти розділів кваліфікаційної роботи

Розділ	ПІБ, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Постанова задачі	Чілікіна Т. В.		
Інформаційний огляд	Чілікіна Т. В.		
Теоретична частина	Чілікіна Т. В.		
Практична реалізація	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 завдання		
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 Малишевський Андрій Олександрович

Науковий керівник к. ф.-м. н., Чілікіна Т.В.

Результати захисту кваліфікаційної роботи

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

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 ЧІЛІКІНА

«__» _____202_ р.

План

кваліфікаційної роботи на тему

«Алгоритмізація та програмна реалізація тренажера з теми «Цикли Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації»»

зі спеціальності 122 Комп'ютерні науки

освітня програма 122 «Комп'ютерні науки» ступеня бакалавр

Прізвище, ім'я, по батькові Малишевський Андрій Олександрович**ВСТУП****1. ПОСТАНОВКА ЗАДАЧІ****2. ІНФОРМАЦІЙНИЙ ОГЛЯД****2.1. Огляд робіт зі схожим завданням****2.2. Позитивні аспекти розглянутих робіт****2.3. Вади розробок розглянутих робіт****2.4. Актуальність розробки тренажера****3. ТЕОРЕТИЧНА ЧАСТИНА****3.1. Теоретичні матеріали****3.1.1 Цикл For ... Next****3.1.2 Цикл For Each ... Next****3.1.3 Цикл Do While ... Loop****3.1.4 Цикл Do Until ... Loop****3.2. Алгоритм роботи тренажеру****3.3. Блок-схема програми-тренажера****4. ПРАКТИЧНА ЧАСТИНА****4.1. Опис програмного забезпечення****4.2. Методи розробки та тестування програмного забезпечення****ВИСНОВКИ****СПИСОК ІНФОРМАЦІЙНИХ ДЖЕРЕЛ****ДОДАТОК А.**

Здобувач вищої освіти _____ Андрій МАЛИШЕВСЬКИЙ

«__» _____202_ р.

РЕФЕРАТ

Записка: 45 с., 19 рис., 0 таблиць, 1 додаток, 8 джерел.

ЦИКЛИ EXCEL VBA, НАВЧАЛЬНИЙ ТРЕНАЖЕР, VISUAL STUDIO 2019

Об'єкт розробки – програмне забезпечення у вигляді тренажеру з теми «Цикли в Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації».

Мета роботи – програмно реалізувати програмне забезпечення з теми «Цикли в Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації».

Методи дослідження – програмна реалізація була проведена за допомогою платформи Microsoft Windows. Для створення програми було використано мову програмування C++ та Microsoft Visual Studio.

Було розглянуто вже створені тренажери та виведено їх позитивні та негативні сторони. Розроблено алгоритм програми з теми «Цикли в Excel VBA». Програмно реалізовано тренажер з дисципліни «Організація та обробка електронної інформації».

Здійснено програмну реалізацію тренажеру з можливістю вибору типу роботи (Теоретична частина та практична частина).

ЗМІСТ

ВСТУП.....	7
1 Постановка задачі	9
2 Інформаційний огляд	10
2.1. Огляд робіт зі схожим завданням.....	10
2.2. Позитивні аспекти розглянутих робіт.....	10
2.3. Вади розробок розглянутих робіт	10
2.4. Актуальність розробки тренажера	11
3 Теоретична частина	12
3.1 Теоретичні матеріали.....	12
3.1.1 Цикл For ... Next	12
3.1.2 Цикл For Each ... Next.....	12
3.1.3 Цикл Do While ... Loop	13
3.1.4 Цикл Do Until ... Loop.....	13
3.2 Алгоритм роботи тренажера.....	15
3.3 Блок-схема програми-тренажера.....	24
4 Практична частина	26
4.1 Опис програмного забезпечення	26
4.2 Методи створення та тестування програмного забезпечення.....	27
ВИСНОВКИ.....	43
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	44
ДОДАТОК А.....	45

ВСТУП

З року у рік розвиваються усі галузі людської активності. Навчання також не стоїть на місці - з'являються нові підручники, методики та види проведення та закріплення знань. Одними з таких нововведень за останні роки є різні програмні забезпечення у вигляді різноманітних тренажерів.

Перш за все, такі тренажери призначені для закріплення та вдосконалення знань та умінь у розв'язанні певних завдань у конкретній предметній області. Так як у наш час такий підхід до навчання можна рахувати новітнім, зараз дуже актуальним є створення нових, або вдосконалення існуючих тренажерів, з метою спрощення та покращення навчання майбутніх фахівців.

Мета розробки – алгоритмізація та розробка програмного забезпечення для тренажеру дистанційного курсу «Організація та обробка електронної інформації».

Об'єкт розробки – створення програмного забезпечення для дистанційного навчання з теми «Цикли в Excel VBA» дистанційного курсу «Організація та обробка електронної інформації».

Предмет розробки – програмне забезпечення для тренажера з теми «Цикли в Excel VBA» дистанційного курсу «Організація та обробка електронної інформації», для навчання та кращого засвоєння теми циклів у мові VBA.

Методи розробки – для реалізації програмного забезпечення було використано мову програмування C++ та середовище розробки Visual Studio 2019.

Структура пояснювальної записки до бакалаврської роботи:

- титульний аркуш;
- завдання;
- реферат;
- зміст;
- вступ;
- теоретична частина;

- практична частина;
- висновки;
- рекомендації;
- список використаних джерел;

Обсяг пояснювальної записки: 45 стор., в т.ч. основна частина 39 стор., джерел - 8.

1 Постановка задачі

1.1 Постановка задачі розробки тренажеру

Основною метою даної бакалаврської роботи є створення та програмна реалізація тренажеру з теми «Цикли в Excel VBA».

Завдання бакалаврської роботи:

1. Обрання способу реалізації тренажеру у вигляді програми.
2. Розробка алгоритму роботи програмного забезпечення.
3. Створення Блок-схеми алгоритму роботи програми.
4. Програмна реалізація тренажеру з теми «Цикли в Excel VBA».
5. Проводення тестів створеного тренажеру.

Основними вимогами до програмного забезпечення є:

1. Зрозумілість та покроковість видачі інформації.
2. Навігація у застосунку за допомогою кнопок.
3. Можливість переходу між завданнями.
4. Вивід результату після відповіді.

2 Інформаційний огляд

2.1 Огляд робіт, зі схожим завданням

Для огляду, було обрано наступні дві роботи зі схожим завданням, а саме:

1. Срібний С.Ю. Пояснювальна записка до бакалаврської роботи на тему «Цикл While в Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації».
2. Малишева Д.В. Пояснювальна записка до бакалаврської роботи на тему «Цикл For в Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації».

2.2 Позитивні аспекти розглянутих робіт

Робота Срібний С.Ю.

- Зручне вікно налаштувань перед запуском;
- Наявність кнопок для переходу до наступного кроку, або повернення до попереднього;
- Можливість повернутися до вводу та виправити свою відповідь за допомогою відповідної кнопки;

Робота Малишева Д.В.

- Наявність теоретичного матеріалу у вигляді зображень;
- Короткий посібник над кожним питанням протягом роботи програми;
- Можливість обрати підказку та саме вікно підказки;

2.3 Вад розробок розглянутих робіт

Після проведення аналізу обраних робіт, було виявлено наступні недоліки:

Робота Срібний С.Ю.

1. Відсутність можливості ще раз повторно ввести дані після повторного введення відповіді при етапі роботи з практичними завданнями.

Робота Малишева Д.В.

1. Відсутність можливості обрати практичну частину програми, не пройшовши теоретичну частину.
2. Не зручне та перевантажене меню з завданнями;

2.4 Актуальність розробки тренажеру

Після розглядання схожих за темою робіт було вирішено створити нове програмне забезпечення типу тренажер, взявши до уваги недоліки та позитивні аспекти вже існуючих тренажерів, з метою покращення процесу роботи та результати роботи з тренажером нових користувачів.

Наразі існуючі тренажери мають свої недоліки, тож можна вважати актуальним та необхідним створення нових тренажерів.

3 Теоретична частина

3.1 Теоретичні матеріали

У мові VBA, як і у інших мов, існують зручні оператори, створені задля спрощення або прискорення виконання тих чи інших завдань. Таким чином, для випадків де потрібно повторювати одну і ту ж дію декілька разів, існують рішноманітні оператори циклів. Всього у мові VBA існує 3 різні оператори циклу.

3.1.1 Цикл For ... Next

Цикл **For ... Next** працює за принципом послідовного прийняття змінною значення з вказаного діапазону, та при кожній зміні значення змінної, виконання певних дій що прописані у тілі циклу. Наглядно це можна побачити у прикладі :

```
For i = 1 To 10
    Debug.Write(i.ToString & " ")
Next
```

Як ми бачимо, цикл запрограмований після кожного прийняття змінною нового значення виводити цю змінну та пробіл. Як результат ми отримаємо :

Output : 1 2 3 4 5 6 7 8 9 10

3.1.2 Цикл For Each ... Next

Якщо Вам потрібно виконати набір інструкцій для кожного елементу масиву або колекції, зручніше всього використати саме варіант циклу **For Each ... Next**. Для наочності розглянемо найпростіший приклад :

Створимо масив з елементами типу Integer :

```
Dim nums As Integer() = {0, 1, 2, 3, 4, 5}
```

Після цього використаємо цикл For Each ... Next, щоб вивести усі елементи нашого масиву :

```
For Each i As Integer in nums
```

```
Debug.Write(i.ToString & “ ”)
```

Next

У результаті отримаємо наступне :

Output : 0 1 2 3 4 5

3.1.3 Цикл Do While ... Loop

Ще один вид циклів, особливістю якого, є виконання блоку коду до тих пір, допоки виконується задана умова. Виходячи з цього, за допомогою циклу **Do While ... Loop** можна зручно та швидко виконати деякі операції з конкретними умовами. Розглянемо приклад :

Створимо змінну **checker** типу **String** та присвоїмо їй значення “OK”:

```
Dim checker As String
```

```
Checker = ”OK”
```

Тепер за допомогою функції ми будемо надавати користувачу змогу змінювати значення змінної **checker**:

```
Do While checker = “OK”
```

```
checker = InputBox(“Введіть ”OK” для повторення циклу”)
```

```
Loop
```

У результаті ми отримаємо нескінченне повторення циклу, допоки користувач буде вводити у вікно “OK”.

3.1.4 Цикл Do Until ... Loop

Також, дуже часто зустрічаються ситуації, при яких нам потрібно виконувати циклічні дії зі змінною, до тих пір, поки вона не прийме якесь із значень. Такі задачі дуже легко виконати саме за допомогою циклу **Do Until ... Loop**. Пропоную для наочності, змінити та розглянути минулий приклад, згідно нової умови:

Знову створюємо змінну **checker** типу **String**, але цього разу не будемо присвоювати їй значення :

```
Dim checker As String
```

Далі прописуємо цикл **Do Until ... Loop** що буде надавати користувачу змогу

змінювати нашу змінну **checker** :

Do Until checker = "OK"

checker = InputBox("Не вводьте "OK" для повторення
циклу")

Loop

У результаті ми отримали протилежний цикл, який буде нескінченно повторюватися, допоки користувач не введе значення "OK".

3.2 Алгоритм роботи тренажера

Після запуску програми, користувачу на вибір пропонується два варіанти роботи, а саме :

- Теоретична частина
- Практична частина

Теоретична частина передбачає ознайомлення із матеріалами за якими далі користувач буде проходити опитування.

Практична частина передбачає питання у виді тестів та самостійного вводу інформації. Після відповіді та натиснення кнопки “Відповісти”, користувач отримує результат. Якщо відповідь була вірною, на екран виводиться напис “Відповідь вірна!”. Якщо відповідь була не вірною, на екран виводиться напис “Відповідь не вірна” та правильний варіант відповіді.

Практична частина має питання з різних підтем та поділена на три блоки :

- Практична частина з циклу For ... Next (5 Завдань)
- Практична частина з циклу Do While ... Loop (5 Завдань)
- Практична частина з циклу Do Until ... Loop (5 Завдань)

Робота програми покроково :

Крок 1: Після запуску тренажера, користувач переходить до основного меню з вибором варіанту роботи.

Крок 2: Після натискання кнопки “Теоретична частина”, користувач отримує на ознайомлення матеріали з теми.

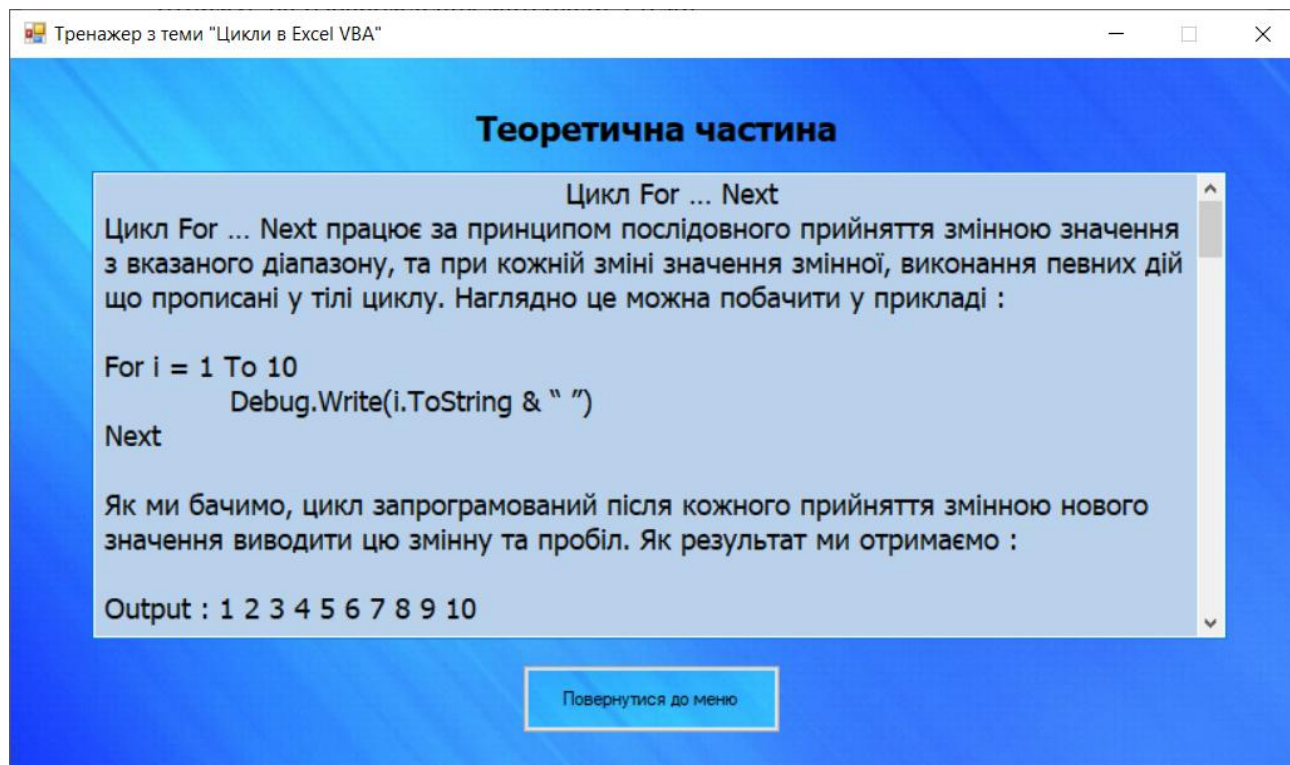


Рисунок 3.1 – Вікно ознайомлення з теоретичним матеріалом

Крок 3: Після натискання кнопки “Практична частина”, користувач переходить до виконання першого з 15 завдань

Крок 4: Отримання практичного завдання :

«Оберіть варіант, що описує алгоритм роботи циклу For ... Next

Варіанти відповіді :

Варіант 1. Працює, доки виконується задана умова циклу;

Варіант 2. Послідовно прийма змінною цикла значення з вказаного діапазону, та виконує певні дії з тілу циклу, при кожному прийнятті нового значення;

Варіант 3. Працює, доки не виконується задана умова циклу;

Правильною є відповідь - №2»

У випадку невірної відповіді, випадає повідомлення «Відповідь не вірна. Правильною є відповідь - №2»

Крок 5: Отримання практичного завдання :

«Скільки ітерацій повторення можна зробити за допомогою цикла For ...

Next?

Варіанти відповіді :

Варіант 1. Безліч;

Варіант 2. Не більше 2 147 483 647 повторень;

Варіант 3. Не менше 5 повторень;

Правильною є відповідь - №1»

Крок 6: Отримання практичного завдання :

«Яким буде результат після виконання циклу?

For i = 6 to 13

i=i+1

Debug.Write(i.ToString & “ ”)

Next

Варіанти відповіді :

Варіант 1. 7 9 11 13;

Варіант 2. 6 7 8 9 10 11 12 13;

Варіант 3. 6 8 10 12;

Правильною є відповідь - №1»

Крок 7: Отримання практичного завдання :

«Яким буде результат після виконання циклу?

Dim nums As Integer() = {9, 18, 23, 43, 24, 5}

For each i As Integer in nums

i=i-10

Debug.Write(i.ToString & “ ”)

Next

Варіанти відповіді :

Варіант 1. 9 18 23 43 24 5;

Варіант 2. 19 28 33 53 34 15;

Варіант 3. -1 8 13 33 14 -5;

Правильною є відповідь - №3»

Крок 8: Отримання практичного завдання :

«Що потрібно записати замість ... , щоб у результаті виконання циклу отримати : 4 6 8?

For i = ... to 18

i=(i-14)*2

Debug.Write(i.ToString & “ ”)

Next

Правильна відповідь - 16»

Крок 9: Отримання практичного завдання :

«Оберіть варіант, що підходить до циклу Do While ... Loop

Варіанти відповіді :

Варіант 1. Працює, допоки виконується задана умова циклу;

Варіант 2. Послідовно прийма змінною цикла значення з вказаного діапазону, та виконує певні дії з тілу цикла, при кожному прийнятті нового значення;

Варіант 3. Працює, допоки не виконується задана умова циклу;

Правильною є відповідь - №1»

Крок 10: Отримання практичного завдання :

«Оберіть один з наведених фактів, що є хибним :

Варіанти відповіді :

Варіант 1. Цикл Do While ... Loop, працює до тих пір, поки значення Condition = True;

Варіант 2. Оператор Loop використовується для позначення місця, з якого VBA повертається у початок циклу після перевірки умови;

Варіант 3. Формат циклу Do While ... Loop запускається 1 або більше разів, поки виконується умова;

Правильною є відповідь - №3»

Крок 11: Отримання практичного завдання :

«Яким буде результат після виконання циклу?

```
Dim res, i As Integer
```

```
res = 0
```

```
i = 1
```

```
Do While i < 11
```

```
res=res+i
```

```
Loop
```

```
Debug.Write(res.ToString)
```

Варіанти відповіді :

Варіант 1. 10;

Варіант 2. 55;

Варіант 3. 66;

Правильною є відповідь - №2»

Крок 12: Отримання практичного завдання :

«Яким буде результат після виконання циклу?

```
Dim res, i As Integer
```

```
res = 0
```

```
i = 1
```

```
Do
```

```
res=res+i
```

```
Loop While i<1
```

```
Debug.Write(res.ToString)
```

Варіанти відповіді :

Варіант 1. 1;

Варіант 2. 0;

Варіант 3. -1;

Правильною є відповідь - №1»

Крок 13: Отримання практичного завдання :

«Що потрібно записати замість ... , щоб у результаті виконання циклу отримати : 29 22 15 8 1?

```
Dim i As Integer
```

```
i = ...
```

```
Do While i>0
```

```
i=i-7
```

```
Debug.Write(i.ToString)
```

```
Loop
```

Правильна відповідь - 36»

Крок 14: Отримання практичного завдання :

«Оберіть варіант, що підходить до циклу Do Until ... Loop

Варіанти відповіді :

Варіант 1. Працює, доки виконується задана умова циклу;

Варіант 2. Послідовно приймає значення з вказаного діапазону, та виконує певні дії з тілу циклу, при кожному прийнятті нового значення;

Варіант 3. Працює, доки не виконується задана умова циклу;

Правильною є відповідь - №3»

Крок 15: Отримання практичного завдання :

«Оберіть один з наведених фактів, що є хибним :

Варіанти відповіді :

Варіант 1. Цикл Do Until ... Loop, працює до тих пір, поки значення Condition = True;

Варіант 2. Оператор Loop використовується для позначення місця, з якого VBA повертається у початок циклу після перевірки умови;

Варіант 3. Формат циклу Do Until ... Loop запускається 0 або більше разів, поки виконується умова;

Правильною є відповідь - №1»

Крок 16: Отримання практичного завдання :

«Яким буде результат після виконання циклу?

```
Dim res, i As Integer
```

```
res = 0
```

```
i = 1
```

```
Do Until i > 50
```

```
res=(res+i)*2
```

```
Loop
```

```
Debug.Write(res.ToString)
```

Варіанти відповіді :

Варіант 1. 30;

Варіант 2. 62;

Варіант 3. 66;

Правильною є відповідь - №2»

Крок 17: Отримання практичного завдання :

«Яким буде результат після виконання циклу?

```
Dim res, i As Integer
```

```
res = 100
```

```
i = 10
```

```
Do
```

```
res=res-i
```

```
i=i-1
```

```
Loop Until i=0
```

```
Debug.Write(res.ToString)
```

Варіанти відповіді :

Варіант 1. 45;

Варіант 2. 10;

Варіант 3. 100;

Правильною є відповідь - №1»

Крок 18: Отримання практичного завдання :

«Що потрібно записати замість ... , щоб у результаті виконання циклу отримати : 4 6 10 18?

```
Dim i As Integer
```

```
i = ...
```

```
Do
```

$i = (i-1) * 2$

Debug.Write(i.ToString)

Loop Until $i < 11$

Правильна відповідь - 2»

Крок 19. Якщо відповідь на будь-яке завдання буде Не вірною, на екран виведеться вірна відповідь та користувач перейде до наступного завдання.

Крок 20. Після закінчення роботи з тренажером, користувач має змогу повернутися до головного меню тренажеру, натиснувши на кнопку “Повернутися до Меню”.

3.3 Блок-схема програми-тренажера

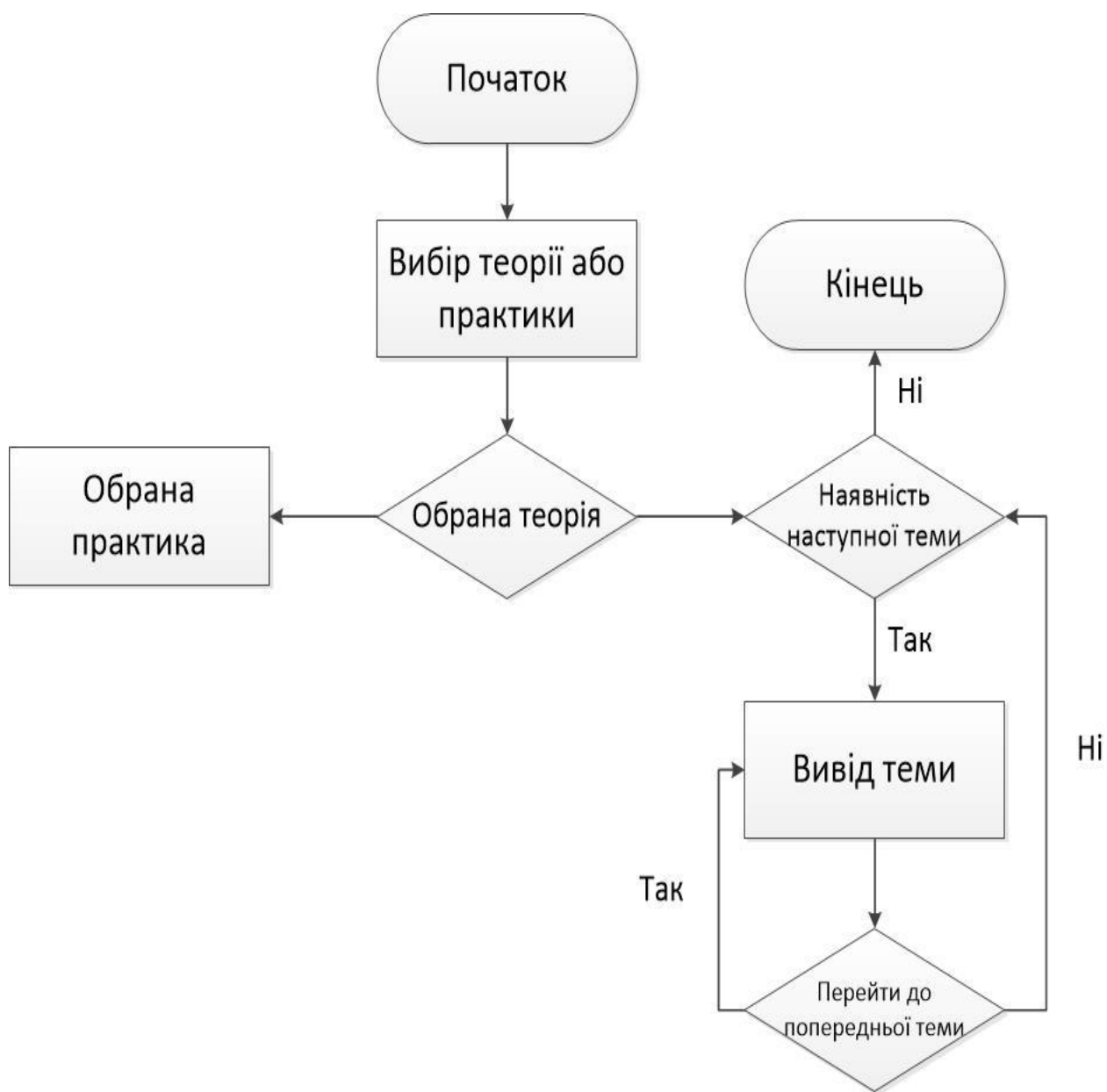


Рисунок 3.2 – Блок-схема програми-тренажера



Рисунок 3.3 – Блок-схема роботи з практичним матеріалом

4 Практична частина

4.1 Опис програмного забезпечення

Обґрунтування вибору мови та середовища програмування

Тренажер був створений з використанням середовища розробки Visual Studio 2019 та Windows Form. Програма складається з великої кількості самостійних форм, які взаємодіють між собою шляхом послідовного закриття та відкриття форм. Для зручного переходу між формами було реалізовано відповідні кнопки.

Мова програмування C++ була обрана через наступні переваги:

- Швидкодія та ефективність. C++ є мовою програмування, що забезпечує швидку та ефективну роботу програм.
- Кросплатформеність. C++ дозволяє розробляти програми що працюють на різних операційних системах.
- Популярність та актуальність. C++ є однією з переводи́х та найпопулярніших мов програмування у світі, тож вона постійно отримує оновлення і нові можливості у покращенні продукту.

Середовище розробки Visual Studio 2019 було обрано через те, що воно дозволяє швидко та ефективно створювати програмне забезпечення та допомагає у його тестуванні та налагодженні, через свої вбудовані інструменти.

Під час створення тренажеру було використано такі інструменти:

- Button – Інструмент, за допомогою якого реалізовані усі кнопки у програмі.
- Label – Інструмент, за допомогою якого реалізовані всі написи у програмі.
- RadioButton – Інструмент, за допомогою якого було реалізовано можливість вибору правильної відповіді на думку користувача.

- TextBox – Інструмент, за допомогою якого було реалізовано поле для вводу відповіді користувачем з клавіатури, та для зручного вмісту інформації на формі з теоретичною частиною.

4.2 Методи створення та тестування програмного забезпечення

Опис роботи та тестування тренажера

Для того щоб почати роботу з тренажером потрібно запустити відповідний .exe файл. Після запуску програмного забезпечення, користувача зустрічає головне меню програми. Користувач може обрати тип роботи з тренажером(Ознайомитись з теоретичними матеріалами, або перейти до практичної частини з тестуванням) натиснувши на відповідну кнопку.

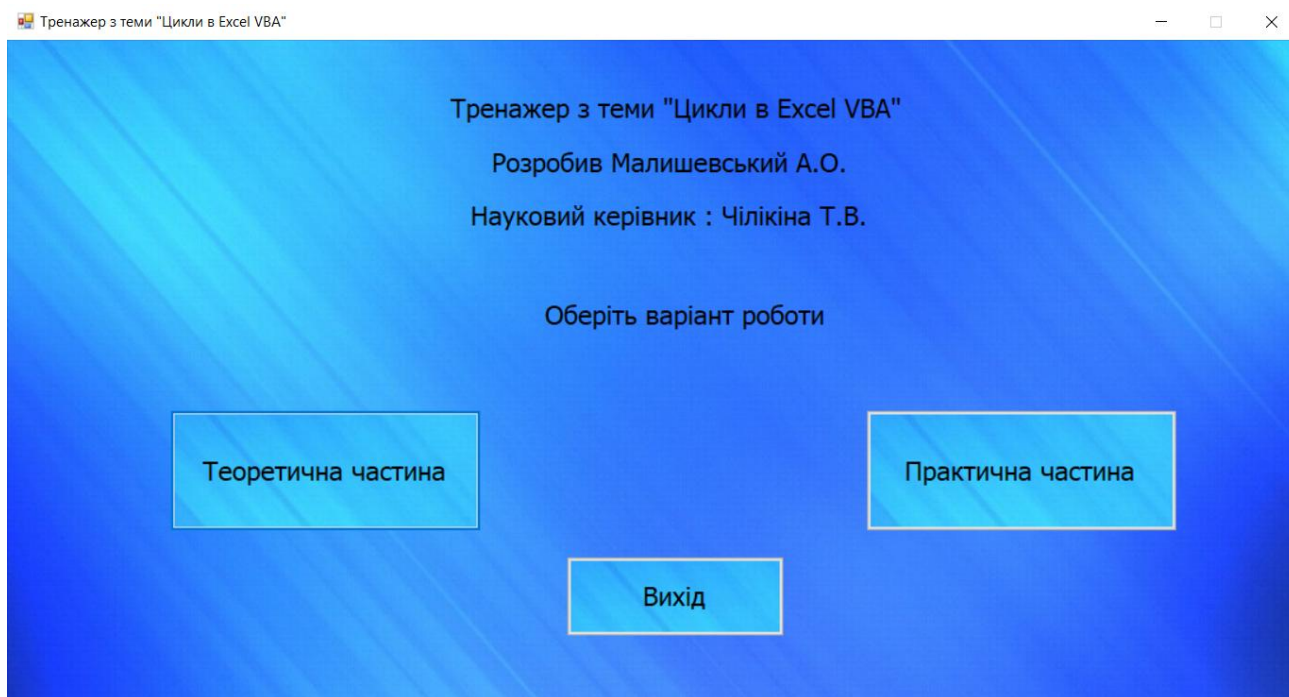


Рисунок 4.1 – Головне меню тренажера

Після натискання на кнопку «Теоретична частина» користувачу відкривається нове вікно де він може ознайомитися з теоретичними матеріалами. Після ознайомлення, користувач може натиснути на кнопку «Повернутися до меню» і повернутися до головного меню.

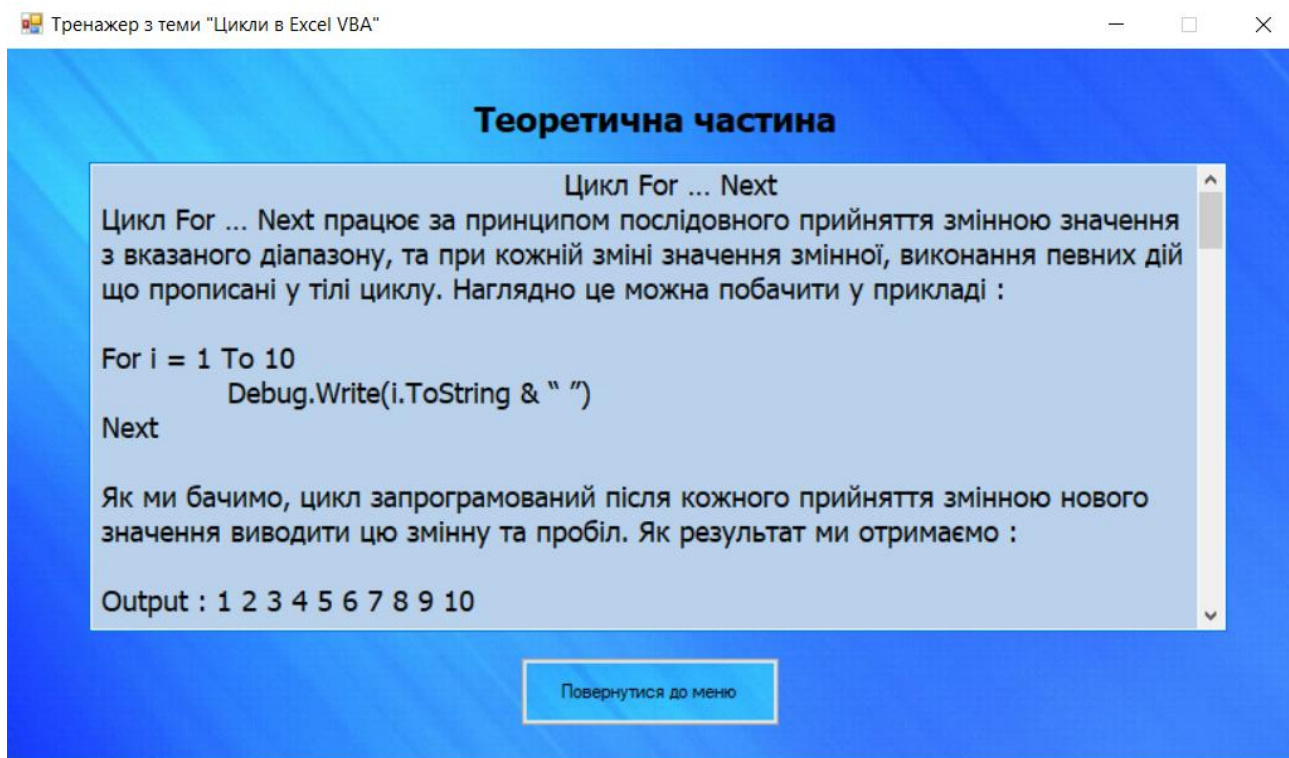


Рисунок 4.2 – Вікно з теоретичними матеріалами

Після натискання на кнопку «Практична частина» користувач переходить до 1 завдання.

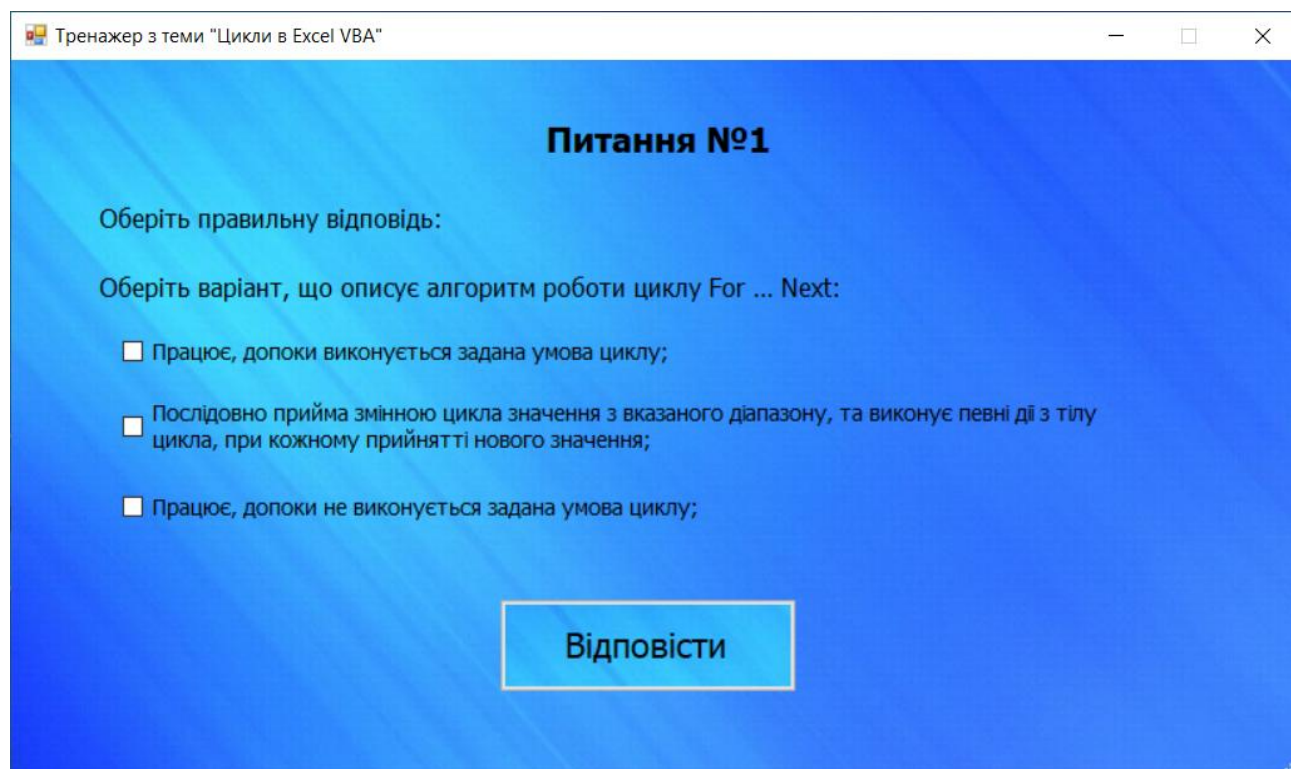


Рисунок 4.3 – Зовнішній вигляд тестових завдань тренажера

Питання №10

Введіть правильну відповідь:

Що потрібно записати замість ... , щоб у результаті виконання циклу отримати : 29 22 15 8 1?

```
Dim i As Integer  
i = ...  
Do While i>0  
    i=i-7  
    Debug.Write(i.ToString)  
Loop
```

Відповідь:

Відповісти

Рисунок 4.4 – Зовнішній вигляд завдань з самостійним вводом відповіді

Після проходження усіх тестових та практичних завдань, користувач переходить до екрану з вибором подальшої роботи з тренажером, а саме вибором між кнопками «Повернутися до головного меню» та «Вийти з програми».

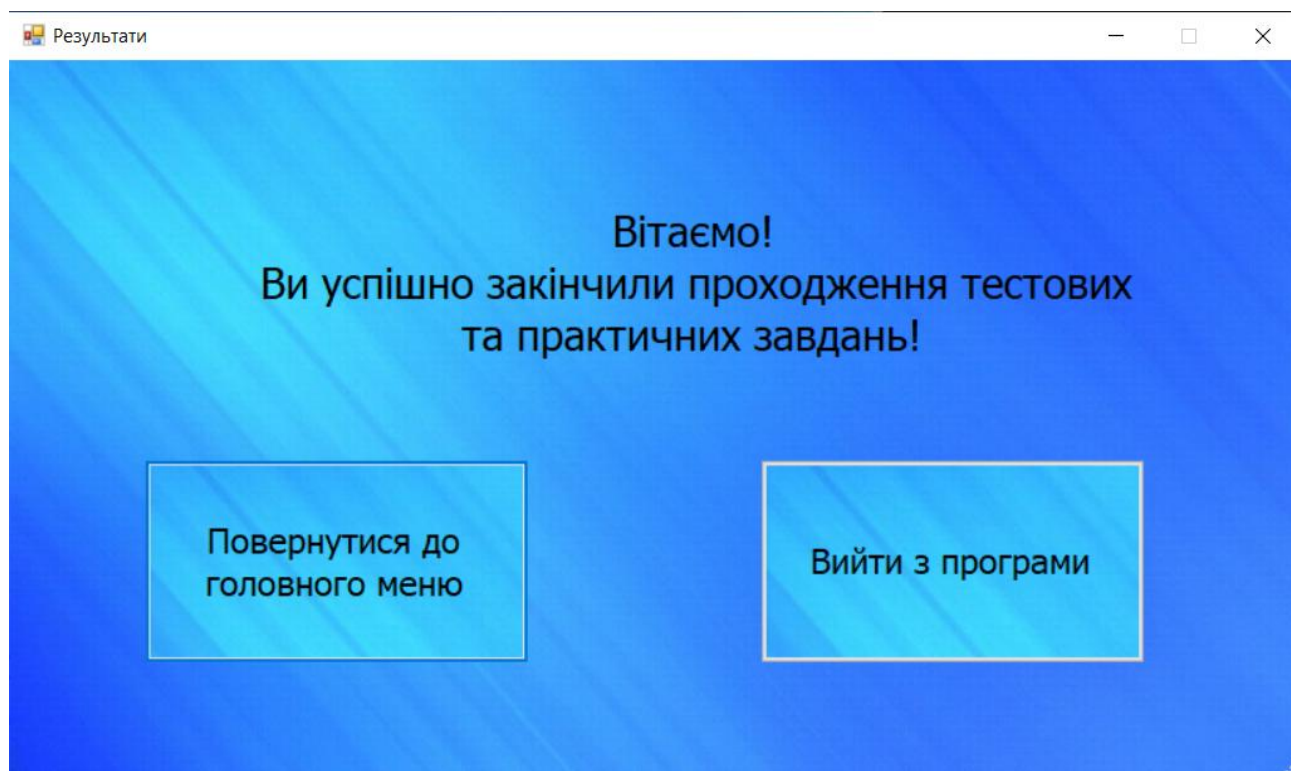


Рисунок 4.5 – Вікно тренажеру після проходження тестування

Процес створення програмного забезпечення

Запуск програми був реалізований за допомогою створення .exe файлу. Після завантаження тренажеру користувачу відкривається перша форма що містить головне меню програми. Кожне вікно, що буде бачити користувач, це окрема Форма Windows Form.

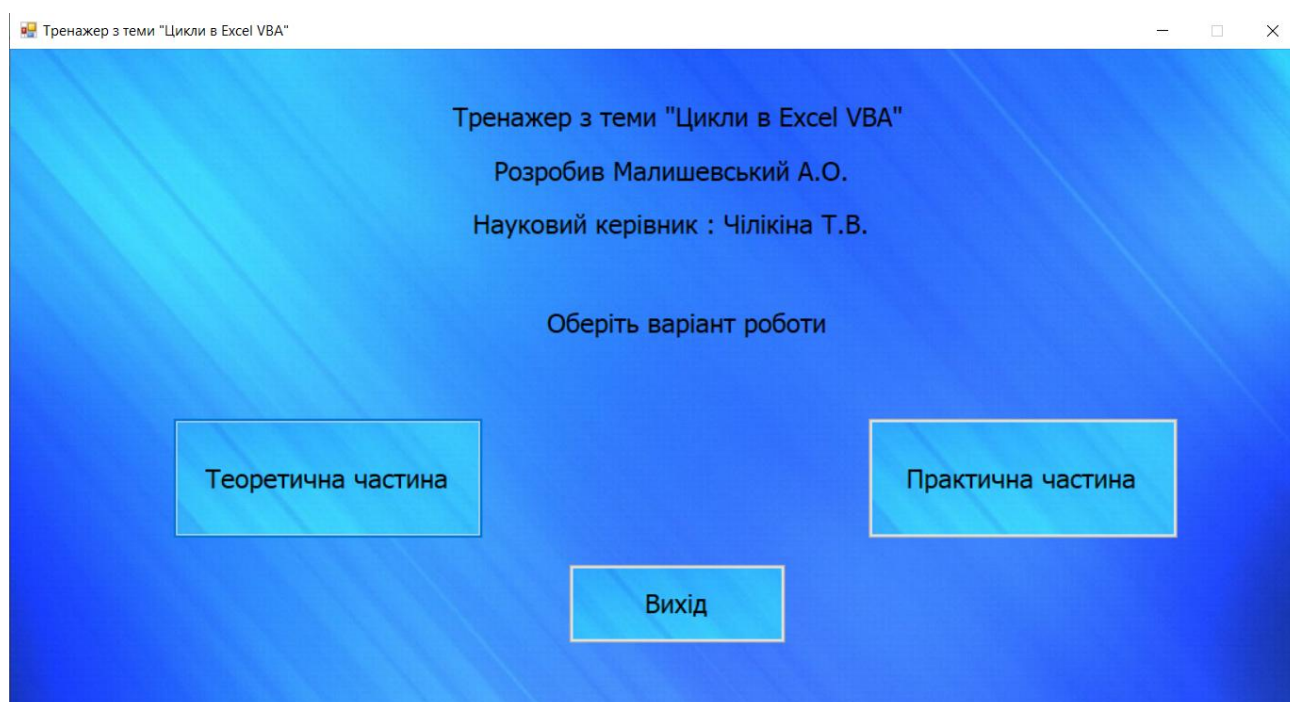


Рисунок 4.6 – Головне меню тренажера

Головне меню містить 3 кнопки управління :

- Теоретична частина – відкриває нову форму яка містить необхідні теоретичні матеріали з теми для ознайомлення перед самим тестуванням.
- Практична частина – відкриває нову форму яка містить перше тестове завдання.
- Вихід – повністю закриває програму.

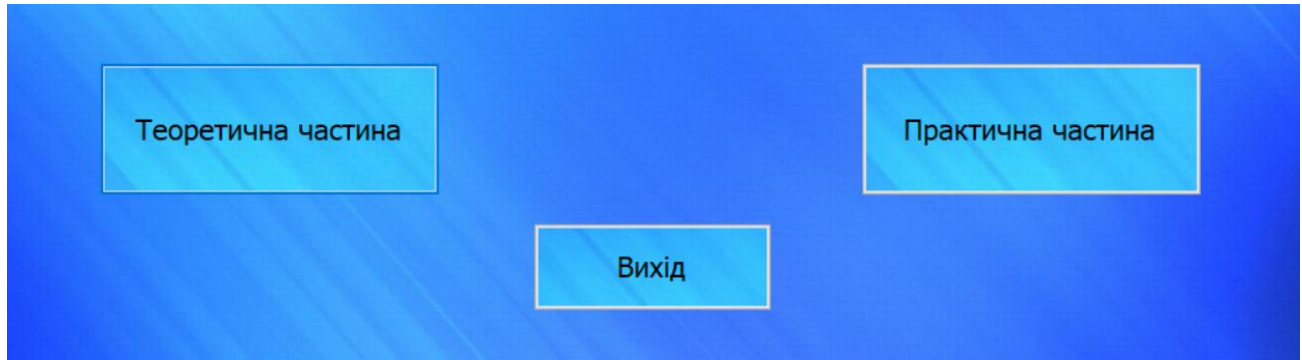


Рисунок 4.7 – Кнопки управління у головному меню тренажеру

Приклад коду для переходу на наступну форму по натисканню на кнопку:

```
MyForm1^ teory = gcnew MyForm1();
teory->Owner = this;
teory->Show();
this->Hide();
```

Розміщення теоретичного матеріалу реалізовано за допомогою такого елемента інтерфейсу як TextBox. Перегляд матеріалів здійснюється за допомогою скролл-бару(Вказано червоною стрілкою на рис. 4.9). Також містить 1 кнопку яка закриває форму з теоретичними матеріалами та повертає користувача до головного меню.

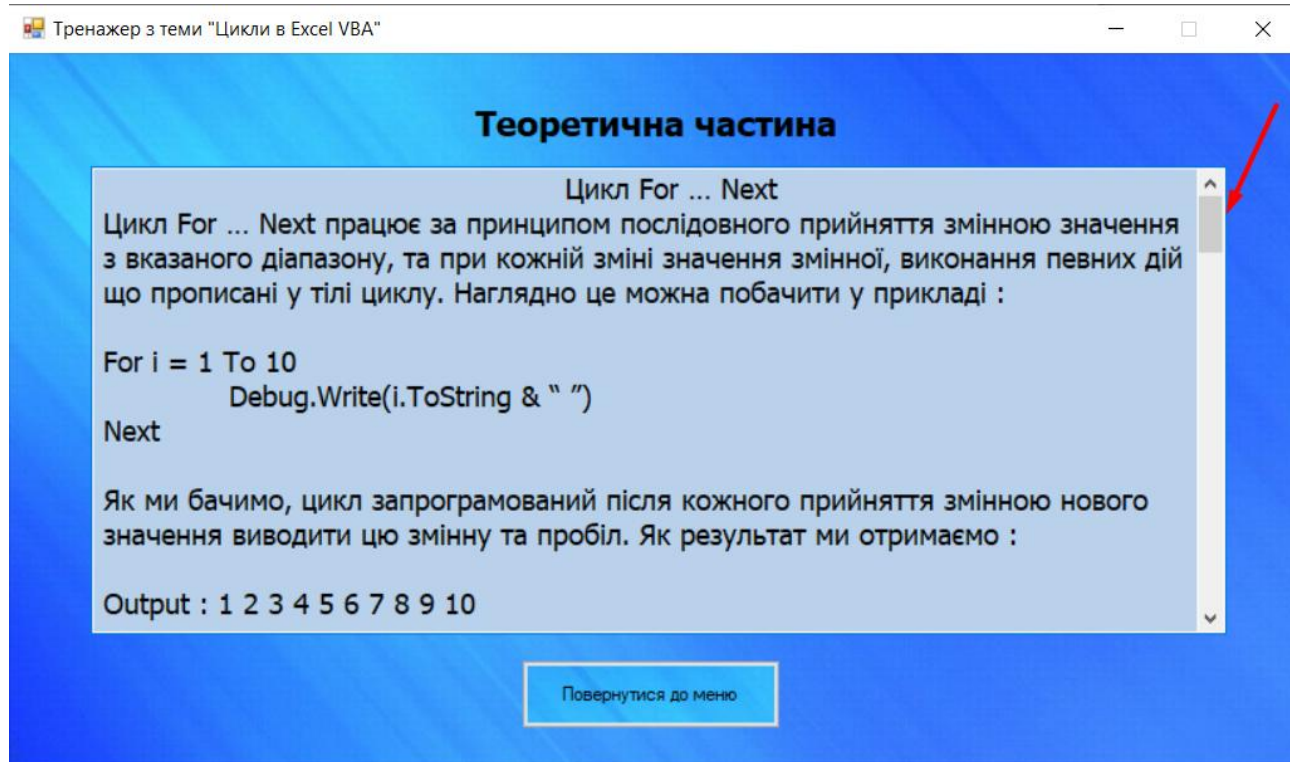


Рисунок 4.8 – Вигляд форми з теоретичними матеріалами

У програмі реалізовано 2 типи завдань :

- Тестові завдання – складаються з питання та 3 варіантів відповіді.
- Завдання з самостійним вводом відповіді – складаються з питання та полем для вводу відповіді користувачем.

Вибір одного з 3 відповідей реалізовано за допомогою такого елементу інтерфейсу як CheckBox. Після того як користувач натискає на елемент типу CheckBox, що знаходиться перед варіантом відповіді, з'являється позначка у виді галочки (Вказано червоною стрілкою на рис 4.10), що свідчить про те, що програма зареєструвала відповідь користувача.

Тренажер з теми "Цикли в Excel VBA"

Питання №1

Оберіть правильну відповідь:

Оберіть варіант, що описує алгоритм роботи циклу For ... Next:

- ☐ Працює, доки виконується задана умова циклу;
- ☒ Послідовно прийма змінною циклу значення з вказаного діапазону, та виконує певні дії з тілу циклу, при кожному прийнятті нового значення;
- ☐ Працює, доки не виконується задана умова циклу;

Відповісти

Рисунок 4.9 – Вигляд форми з тестовими завданнями

Приклад коду з використанням CheckBox:

```
If(checkbox1 = Checked){  
...  
}  
  
If(checkbox2 = Checked){  
...  
}  
  
If(checkbox3 = Checked){  
...  
}
```

Якщо користувач обере відразу 2, або 3 варіанти відповіді, на екран виведеться наступне повідомлення, та після натискання на кнопку “ОК” це вікно зникне та користувач знову матиме змогу обрати 1 варіант відповіді та відповісти.

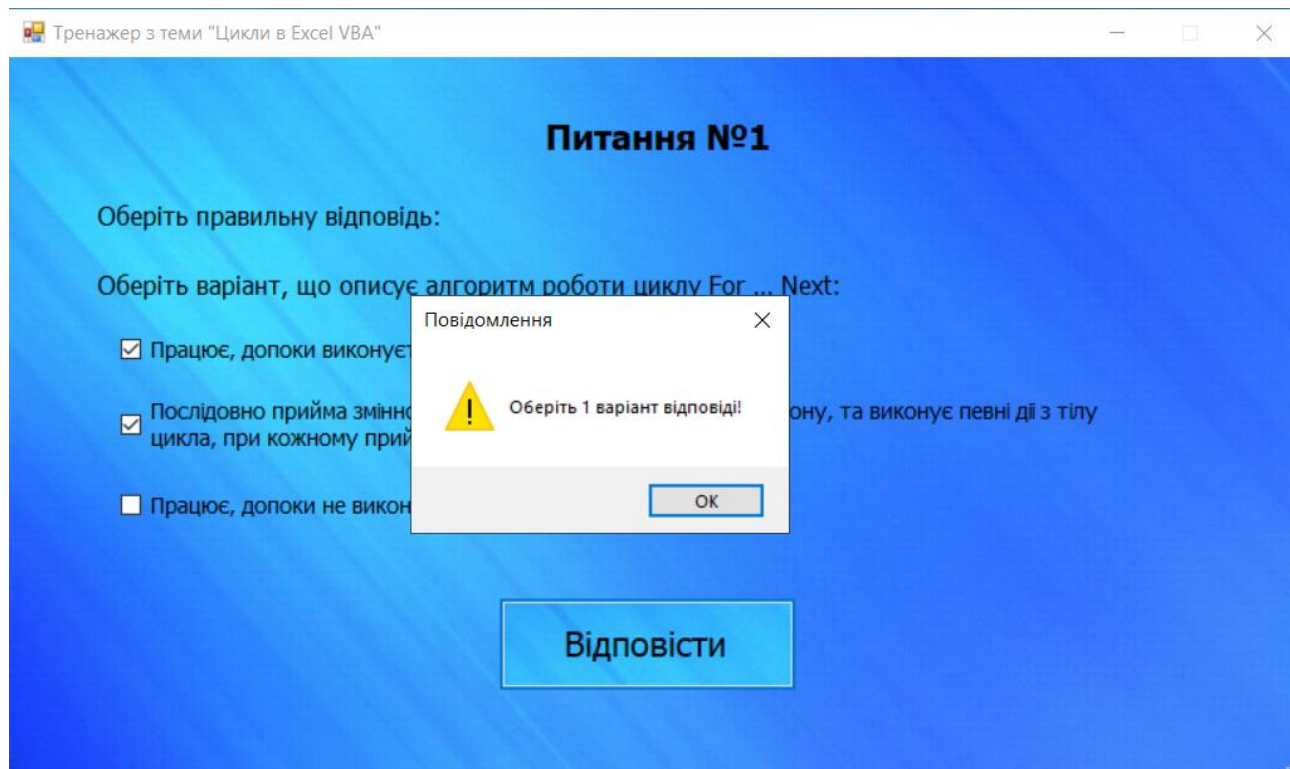


Рисунок 4.10 – Вигляд форми якщо користувач обрав декілька варіантів відповіді

Якщо користувач обирає неправильну відповідь, виводиться відповідне повідомлення з правильною відповіддю та користувач переходить до наступного завдання після того як натисне на кнопку “ОК”.

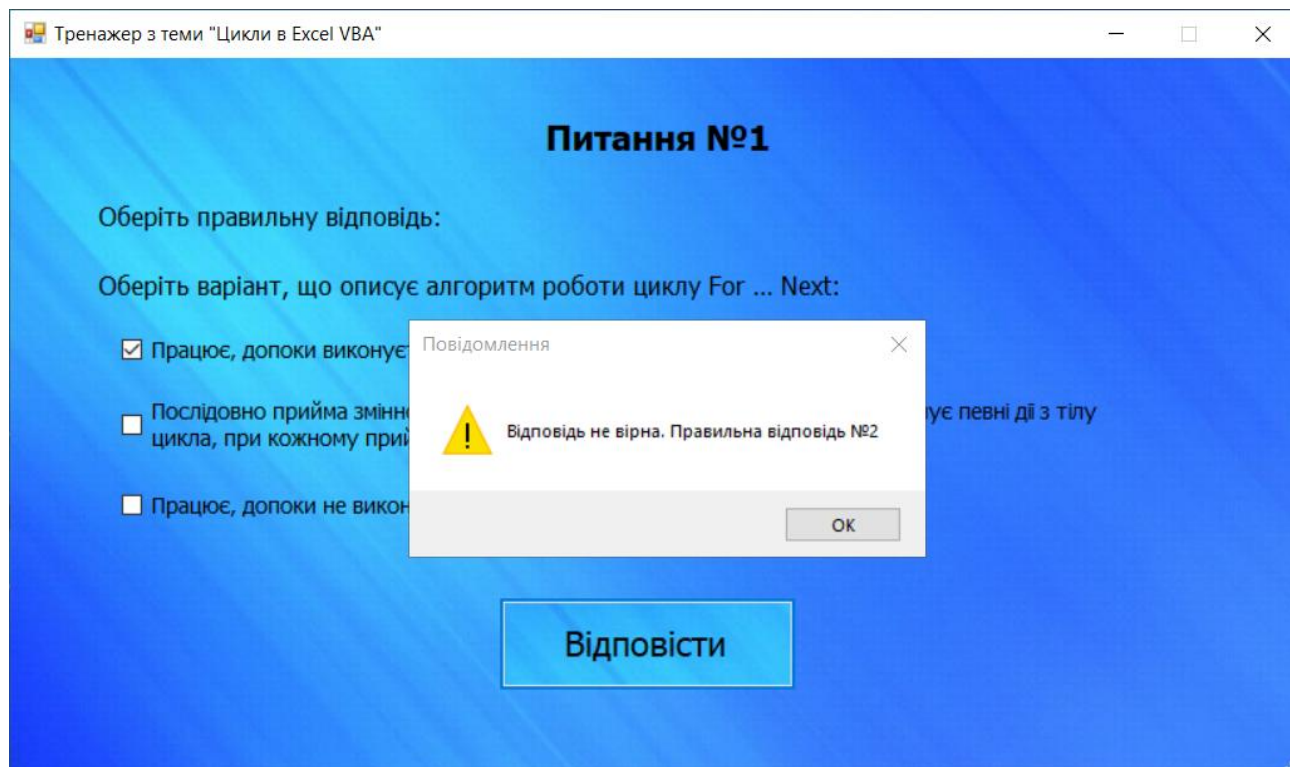


Рисунок 4.11 – Виведення повідомлення про помилку

Якщо користувач обирає правильну відповідь, виводиться відповідне повідомлення та користувач переходить до наступного завдання після того як натисне на кнопку “ОК”.

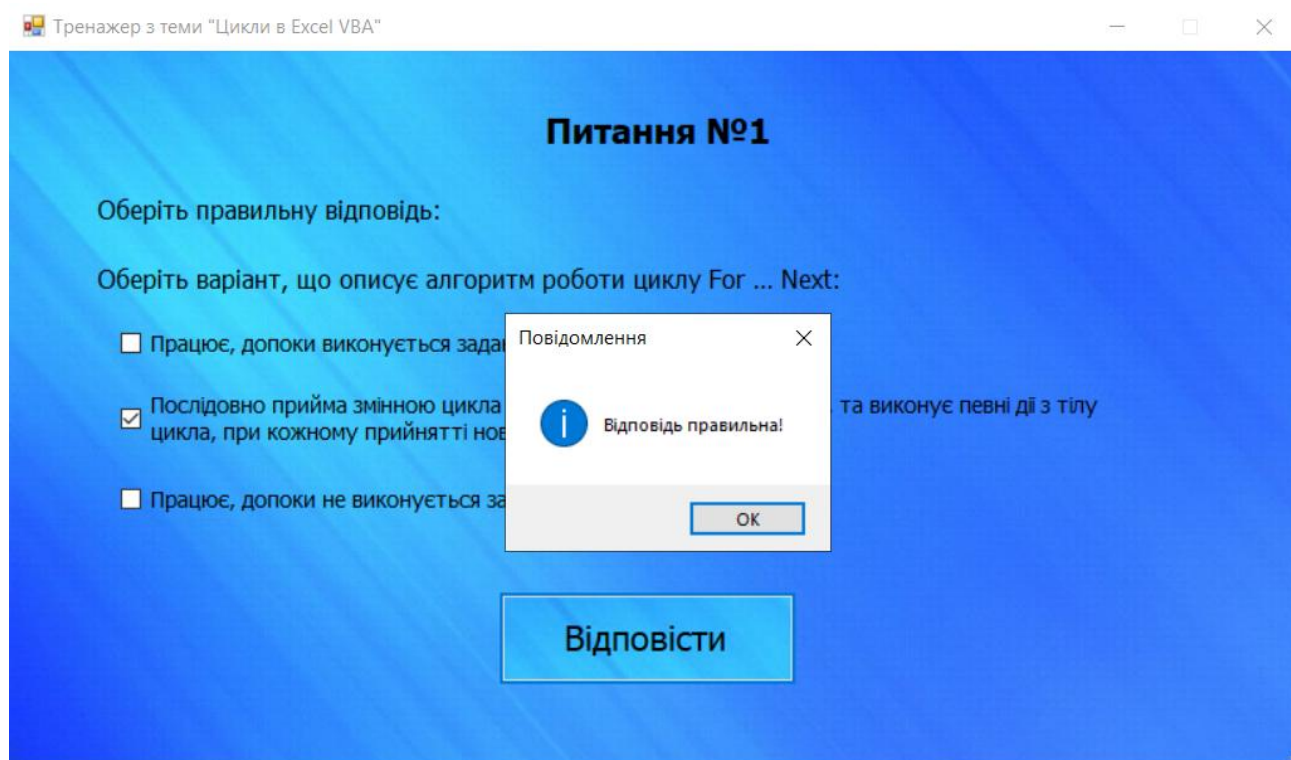


Рисунок 4.12 – Виведення повідомлення про правильну відповідь

Вивід повідомлення реалізовано за допомогою MessageBox. Приклад коду для виведення повідомлення:

```
MessageBox::Show(This, "...", MessageBoxButtons::OK,  
MessageBoxIcon::Information);
```

Також на формі існує кнопка “Відповісти” за допомогою якої користувач може дізнатися результат своєї відповіді та перейти до наступного питання.

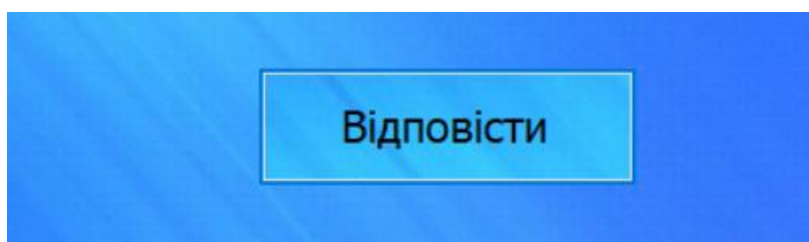


Рисунок 4.13 – Кнопка для перевірки відповіді

Форма з завданнями з самостійним введенням відповіді містить у собі поле для вводу що реалізовано за допомогою елементу інтерфейсу TextBox. Користувач може самостійно вводити відповідь яку він вважаю правильною, натиснувши на поле для вводу(Вказано червоною стрілкою на рис. 4.15) та за допомогою клавіатури ввести свою відповідь.

Питання №5

Введіть правильну відповідь:

Що потрібно записати замість ... , щоб у результаті виконання циклу отримати : 4 6 8?

```
For i = ... to 18
    i=(i-14)*2
    Debug.Write(i.ToString " ")
Next
```

Відповідь:

Відповісти

Рисунок 4.14 – Форма з завданням з самостійним вводом відповіді

Питання №5

Введіть правильну відповідь:

Що потрібно записати замість ... , щоб у результаті виконання циклу отримати : 4 6 8?

```
For i = ... to 18
    i=(i-14)*2
    Debug.Write(i.ToString " ")
Next
```

Відповідь:

Відповісти

Рисунок 4.15– Вигляд після вводу відповіді

Приклад коду з використанням TextBox:

```
String^ ans = textBox1->Text;
```

```

If(ans = ...){
...
}
else{
...
}

```

Якщо користувач обирає правильну відповідь, виводиться відповідне повідомлення та користувач переходить до наступного завдання після того як натисне на кнопку “OK”.

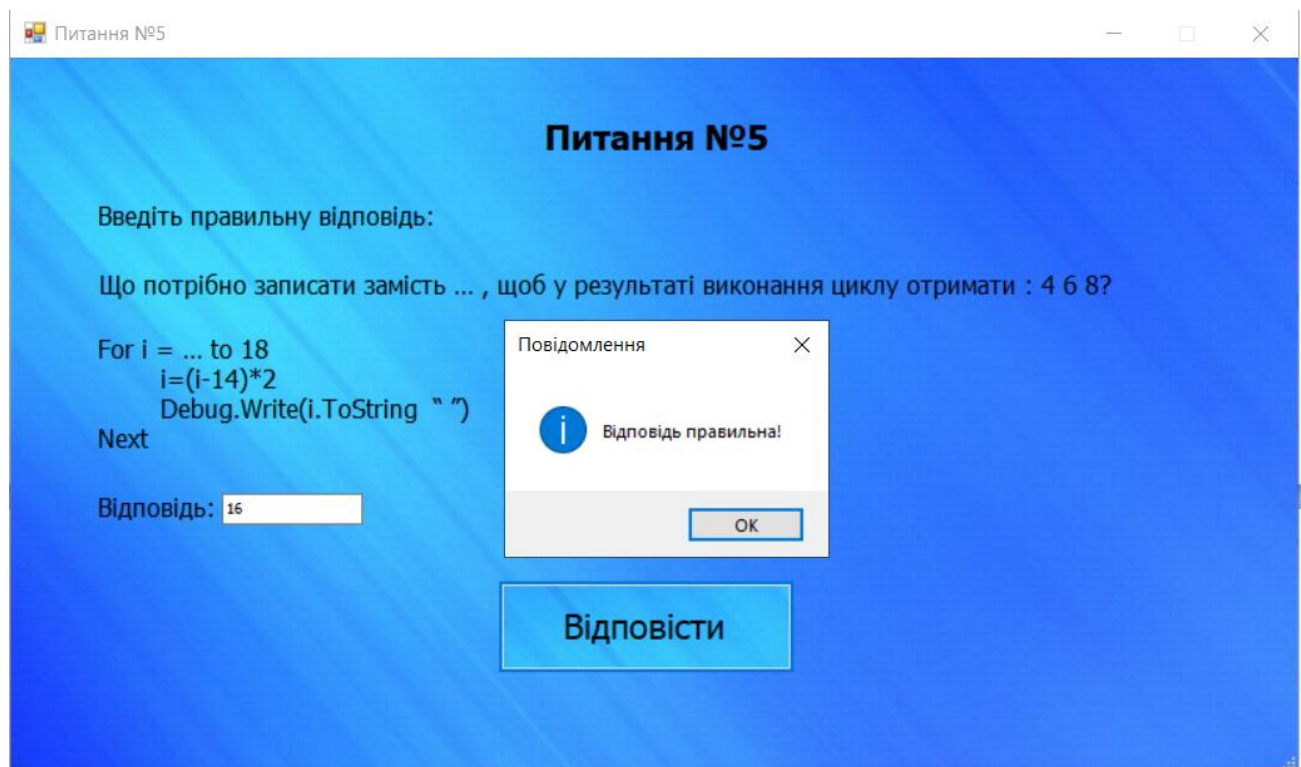


Рисунок 4.16 – Виведення повідомлення про правильну відповідь

Якщо користувач обирає неправильну відповідь, виводиться відповідне повідомлення з правильною відповіддю, після закриття якого він переходить до наступного завдання.

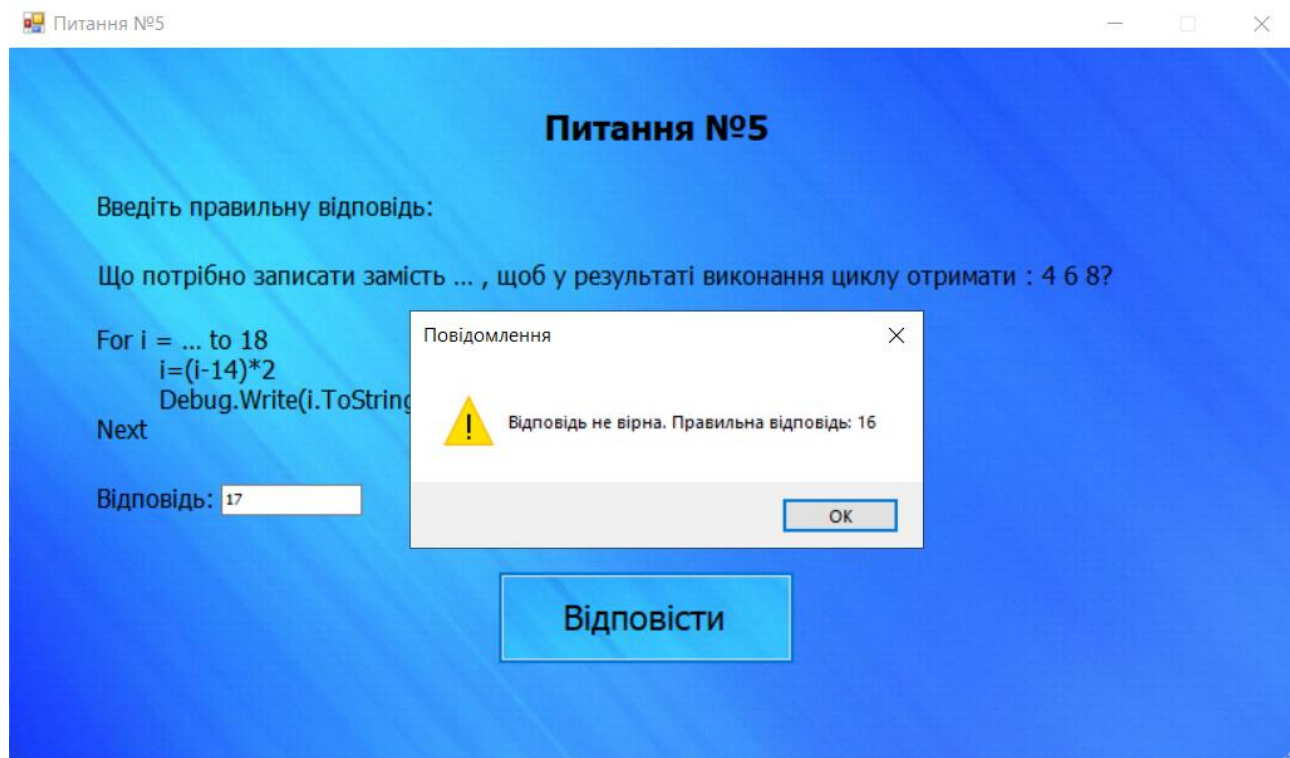


Рисунок 4.17 – Виведення повідомлення про неправильну відповідь

Після закінчення проходження тестових завдань, користувач переходить до останньої форми, яка складається з привітання з проходженням тестування та 2 кнопками :

- Повернутися до головного меню – закриває поточну форму та знову дає змогу користувачу працювати на першій формі з головним меню.
- Вихід з програми - повністю закриває програму.

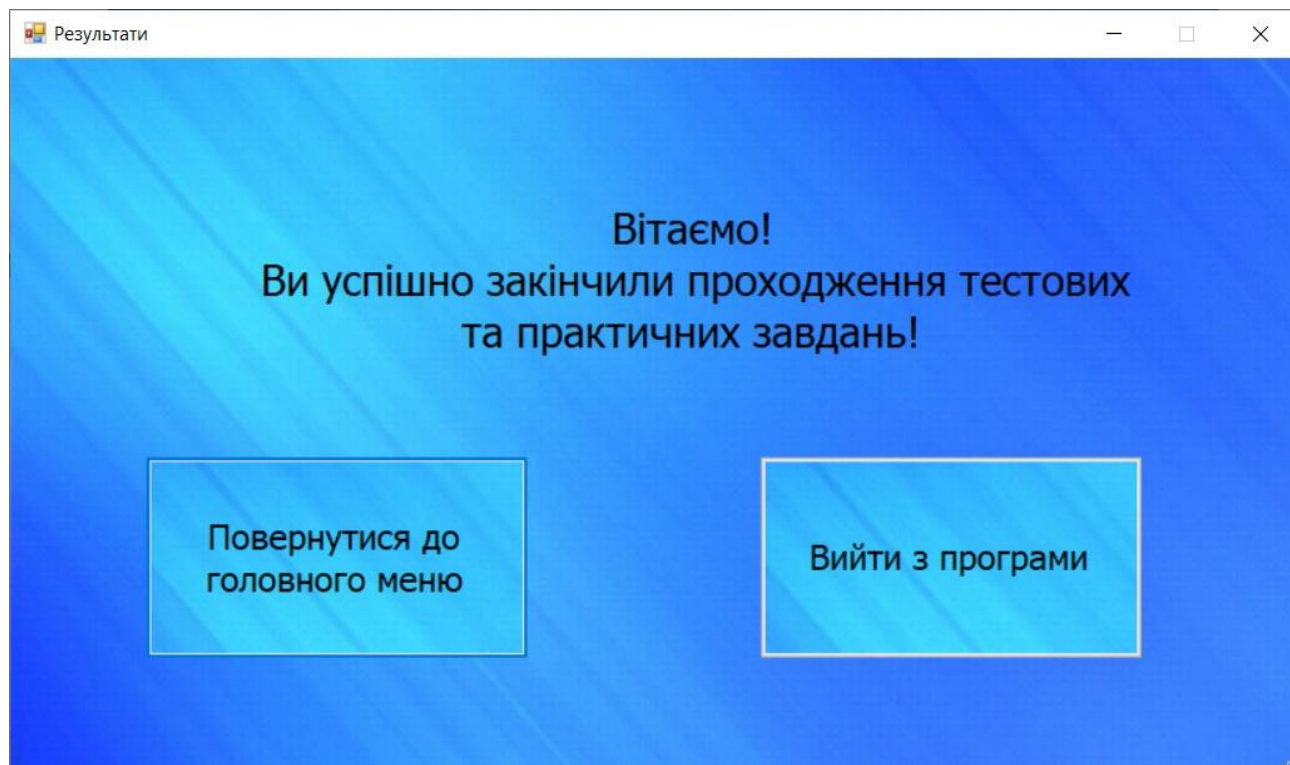


Рисунок 4.18 – Форма після проходження практичної частини

Приклад коду що передбачає закінчення роботи та повне закриття програми :

```
Application::Exit();
```

Також, у тренажері передбачене повне закриття програми за допомогою кнопки управління у верхньому барі програми. Ця кнопка працює на кожній формі та повністю закінчує роботу тренажеру.

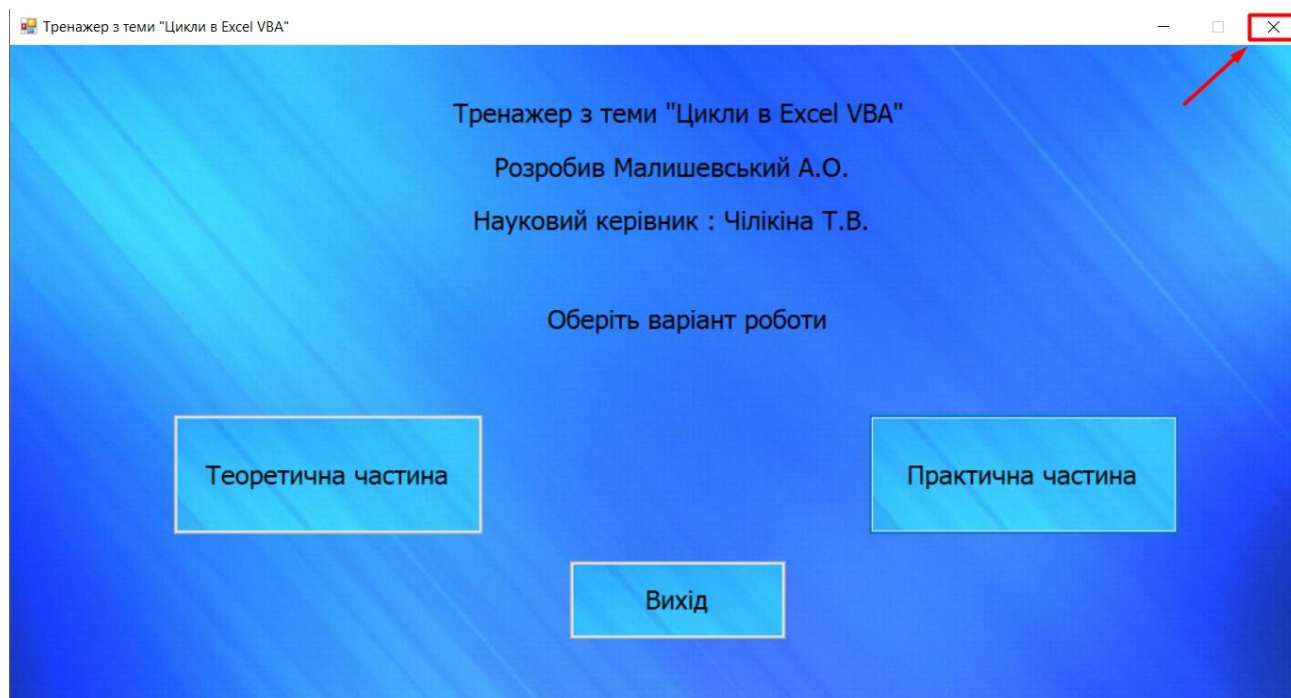


Рисунок 4.19 – Додаткова кнопка для завершення роботи тренажеру

ВИСНОВКИ

У результаті роботи над бакалаврською роботою, було розроблено тренажер з теми «Цикли в Excel VBA» для дистанційного курсу «Організація та обробка електронної інформації».

Результатами бакалаврської роботи є:

- Обрано мову програмування, методи та специфіку роботи тренажеру.
- Розглянуто схожі роботи для покращення створеного тренажеру.
- Розроблено алгоритм роботи тренажеру.
- Створено блок-схеми роботи програми-тренажеру.
- Виконано перевірку на валідність та працездатність програмного забезпечення.

Плюси розробленого програмного забезпечення:

- Зручний та інтуїтивний інтерфейс.
- Приємний дизайн програми.
- Можливість закриття програми декількома способами.

По закінченню бакалаврської роботи було розроблено та протестовано програмне забезпечення у вигляді тренажеру з теми «Цикли в Excel VBA» для дистанційного курсу «Організація та обробка електронної інформації».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ольховська О. В., Черненко О.О. МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ для студентів за освітньою програмою «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» ступеня бакалавра
2. Срібний С.Ю. Пояснювальна записка до бакалаврської роботи на тему «Програмна реалізація тренажеру з теми «Цикл While в Exel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації»» /Срібний С.Ю. [Електронний ресурс] .– Режим доступу до ресурсу: <http://dspace.puet.edu.ua/handle/123456789/11383>
3. w3schools.com [Електронний ресурс] .– Режим доступу до ресурсу: https://www.w3schools.com/cpp/cpp_constructors.asp
4. geeksforgeeks.org [Електронний ресурс] .– Режим доступу до ресурсу: <https://www.geeksforgeeks.org/constructors-c/>
5. Малишева Д.В. Пояснювальна записка до бакалаврської роботи на тему «Програмна реалізація тренажеру з теми «Цикл For в Exel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації»» /Малишева Д.В. [Електронний ресурс] .– Режим доступу до ресурсу: <http://dspace.puet.edu.ua/handle/123456789/11376>
6. Юрій Капітанчук, С++ 2 Forms. РОБОТА З ДВОМА ФОРМАМИ (ПЕРЕХОДИ МІЖ НИМИ). ПЕРЕДАЧА ТЕКСТУ МІЖ НИМИ. | VS 2019 [Електронний ресурс] .– Режим доступу до ресурсу: <https://www.youtube.com/watch?v=fnDlk66D8Xg&t=90s>
7. Доступно і зрозуміло, С++ українською. Моя перша програма [Електронний ресурс] .– Режим доступу до ресурсу: <https://www.youtube.com/watch?v=pvOxoUxSiuc>
8. Moonexcel [Електронний ресурс] .– Режим доступу до ресурсу: http://moonexcel.com.ua/main_ua

ДОДАТОК А

Зразок заповнення заяви бакалавра

Завідувачу кафедри КНІТ
Олені ОЛЬХОВСЬКІЙ
здобувача вищої освіти 4 курсу, групи КН 41 б
освітньо-професійної програми «Комп'ютерні науки»
зі спеціальності 122 «Комп'ютерні науки»
Навчально-наукового інституту денної освіти(заочно-
дистанційного навчання)
Малишевського Андрія Олександровича
моб. тел. 0733293522

Заява

Прошу дозволити мені виконувати кваліфікаційну роботу на тему „Алгоритмізація та програмна реалізація тренажера з теми «Цикли Excel VBA»
дистанційного навчального курсу «Організація та обробка електронної
інформації»„

Здобувач вищої освіти _____ Андрій МАЛИШЕВСЬКИЙ
Науковий керівник _____ Тетяна ЧІЛІКІНА