

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)

« \_\_\_\_ » \_\_\_\_\_ 2023 р.

**КВАЛІФІКАЦІЙНА РОБОТА**  
на тему

**РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА АЛГОРИТМІЗАЦІЯ  
ТРЕНАЖЕРА З ТЕМИ «ОПЕРАТОРИ 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** Шатохін Микита Олегович  
\_\_\_\_\_ « \_\_\_\_ » \_\_\_\_\_ 2023 р.  
(підпис)

**Науковий керівник** к. ф.-м. н., доцент, Чілікіна Тетяна Василівна  
\_\_\_\_\_ « \_\_\_\_ » \_\_\_\_\_ 2023 р.  
(підпис)

**Рецензент**

**ПОЛТАВА 2023 р.**

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ .....	3
ВСТУП .....	4
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ.....	5
РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	6
2.1. Мова програмування VBA.....	6
РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА .....	10
3.1. Оператори мови VBA .....	10
3.2. Алгоритм роботи тренажеру з теми «Оператори Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації» .....	15
3.3. Блок-схема тренажеру .....	20
РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА .....	21
4.1. Обґрунтування вибору програмних засобів.....	21
4.2. Опис програмної реалізації.....	22
4.3. Опис роботи тренажеру .....	25
ВИСНОВКИ.....	31
СПИСОК ЛІТЕРАТУРИ.....	32
ДОДАТОК А. КОД ПРОГРАМИ .....	34

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,  
СКОРОЧЕНЬ, ТЕРМІНІВ**

Умовні позначення, символи, скорочення, терміни	Пояснення умовних позначень, скорочень, символів
PictureBox	Це елемент керування інтерфейсом користувача, який використовується для відображення зображень на формі або контейнері
ComboBox	Дозволяє створювати список варіантів для вибору, і цей список може бути попередньо заповнений або динамічно визначатися під час роботи програми.
Forms	Є одним з популярних способів розробки десктопних додатків для операційної системи Windows у мовах програмування, таких як C# або Visual Basic.
Public Class	Це ключове слово в мові програмування Visual Basic, яке використовується для оголошення нового класу з глобальним доступом. Ключове слово "Public" вказує на те, що клас і його члени (методи, властивості, події тощо) можуть бути доступними і використовуватися з будь-якого місця в програмі.

## ВСТУП

Досягнення інформаційних технологій дозволяють вирішити проблему під час воєнного стану шляхом широкого застосування у сфері освіти дистанційного навчання та впровадження програм тренажерів. Проте, для забезпечення навчального процесу з цією формою навчання на належному рівні необхідне масштабне застосування потужних комп'ютерних систем, які будуть у змозі забезпечити віддалений (дистанційний) доступ до захищених інформаційних систем і ресурсів навчального призначення таких як віртуальні центри знань, електронні бібліотеки, курси дистанційного навчання, електронні підручники, віртуальні лабораторні практикуми, системи тестування, відеоконференції тощо, та інтелектуальне керування цим доступом.

Навчальні електронні інформаційні ресурси, що розміщуються в Інтернеті в цілому і різних інформаційних системах зокрема, розрізняються винятковою різноманітністю. У сучасному світі це вітається, проте кожна така розробка для реалізації її на прийнятному змістовному та методичному рівні вимагає великих часових і фінансових ресурсів.[2]

Мета кваліфікаційної роботи – розробка програмного забезпечення тренажера з теми «Оператори Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації».

Об'єкт розробки – алгоритм, блок-схема, створення тренажера.

Предмет розробки – програмна реалізація тренажера з теми «Оператори Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації».

## РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

Першочерговим завданням роботи є програмування додатку тренажеру з теми «Оператори Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації»

Основні завдання роботи:

- Подивитись матеріал з теми «Оператори Excel VBA»;
- Алгоритмізувати роботу додатку тренажеру;
- Зробити блок-схему функціональної частини тренажеру;
- Програмування програмного забезпечення;
- Опис результатів програмної реалізації навчального додатку;

При розробці програмного забезпечення для тренажера варто врахувати такі основні компоненти:

1. Початкове вікно: Це перше вікно або форма тренажера, де міститься інформація про тематику тренажера. Тут також можуть бути розміщені кнопки для початку тренування, а також дані про розробників і наукового керівника проекту.

2. Вікна з питаннями: Ці вікна включають питання, які стосуються теми тренажера. Вони можуть бути представлені у вигляді тестування, де користувачеві потрібно дати правильну або неправильну відповідь. Ці вікна дозволяють користувачеві взаємодіяти з тренажером шляхом відповідей на питання.

Загалом, для розробки програмного забезпечення тренажера важливо врахувати інтерфейсні вікна, які надають інформацію, пояснення та можливості взаємодії користувача з тренажером.

## РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД

### 2.1. Мова програмування VBA

VBA - відносно легка мова програмування. Вона проста в освоєнні і дозволяє швидко отримувати відчутні результати - конструювати професійні додатки, що вирішують практично всі задачі, що зустрічаються в середо-вищі Windows. При цьому створення багатьох програм з використанням VBA простіше і швидше, ніж за допомогою інших мов програмування.

VBA застосовує технологію візуального програмування, тобто конструювання робочої поверхні програми та елементів його управління безпосередньо на екрані, а також запис всієї програми або її частин за допомогою MacroRecorder.

VBA відноситься до мов об'єктно-орієнтованого програмування (ООП). ООП можна описати як методику аналізу, проектування і створення програмного забезпечення за допомогою об'єктів.

Якщо розглядати офісні додатки з точки зору моделювання їхньої структури, то вони усі складаються з сукупності взаємодіючих об'єктів. Кожен з них наділений специфічним набором властивостей і методів. Сукупність об'єктів і зв'язків між ними називається об'єктною моделлю. Об'єктами виступають основні додатки (Application) та їх складові частини (компоненти) вкладені об'єкти.

Будь-який додаток MS Office має свою об'єктну модель. У вигляді об'єктів представлені всі доступні розробнику елементи офісних програм.

Існують стандартна бібліотека об'єктів VBA і стандартна бібліотека Office, бібліотеки об'єктів Word, Excel, Access та інші об'єктні бібліотеки, що надають різні функціональні можливості при розробці програм.

Структура об'єктної моделі Excel доволі складна, але в той же час найбільш досконала серед інших Windows – додатків. Досконалість Excel відносно VBA викликана наступними чинниками:

Excel був піонером серед Windows – додатків, де був інтегрований VBA. Тому звичайно взаємодія VBA з користувачем в Excel на мій погляд є найбільш комфортною і ефективною.

Повною спадковістю версій відносно VBA. Макроси будь-яких попередніх версій Excel успішно працюють у їх наступних – більш сучасних версіях.

Більше сотні об'єктів нараховує сучасний Excel і їх кількість зростає з кожною новою версією. Звичайно, що у повсякденній роботі вони не використовуються усі відразу. В залежності від поставлених для вирішення задач використовуються лише певний набір об'єктів.

При роботі з об'єктом програміст працює з його властивостями та методами.

Властивості об'єкта – це атрибути об'єкта.

Існує дві групи властивостей:

- властивості-учасники (accessors) – вкладені об'єкти
- термінальні властивості (terminals) – характеристики об'єкта, або його стан.

Властивості мають статус:

Read-Write (R / W) – дозволяє зміни властивості.

Read-Only (R / O) – дозволяє лише читання властивості.

Властивості об'єктів змінюються за допомогою оператора присвоєння, або після виконання методу.

Синтаксис:

Об'єкт. Властивість = Значення

Значення:

- приймає строго певний тип
- результат обчислення повинен відповідати типу

При використанні змінної для значення властивості, її тип повинен відповідати типу значення, можливого для цієї властивості.

Методи об'єкта – це дії над об'єктом.

Методи іноді впливають на значення властивостей.

Методи – це функції, або процедури.

Методи можуть приймати аргументи.

Метод викликається двома способами:

змінна = об'єкт . метод (аргумент)

об'єкт . метод аргумент

Перший спосіб – функціональний (аргумент в дужках після назви методу).

Другий спосіб – операторний (аргумент пишеться через пробіл після назви методу).

Якщо необхідно вказати кілька аргументів, то вони записуються через кому.

## РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

### 3.1. Оператори мови VBA

Оператор – це символ, або вираз, який дає вказівку програмі виконати певну дію над конкретними даними. VBA містить декілька категорій операторів: оператор присвоєння; числові і текстові оператори; логічні оператори; оператори порівняння.

Оператор присвоєння. Символ дорівнює (=) в VBA є оператором присвоювання і використовується для при-своєння значення змінній. Правила запису оператора присвоєння наступні: ліворуч від оператора присвоєння повинна стояти змінна, якій передається значення; праворуч від оператора присвоєння повинен стояти вираз.

Операція присвоєння має дві синтаксичні форми:

1. Let varname = expression;

2. varname = expression;

varname - будь-яка змінна VBA

expression - будь-який вираз VBA

Перший варіант операції присвоєння використовувався в ранніх мовах програмування Basic. Другий варіант використовується в сучасній версії VBA.

При виконанні оператора присвоєння VBA спочатку обчислює вираз праворуч від оператора присвоєння, а потім зберігає результат виразу в змінній, ім'я якої знаходиться зліва від знаку рівності.

$X = 5 + 7$ ;  $Y = X + 5$ ;  $Z = X - Y$ ;  $A = B$ ;  $I = I + 1$ .

Початкуючим користувачам іноді незрозумілий сенс останньої операції присвоєння, коли і в лівій, і в правій частинах операції є одна і та ж змінна. В цьому випадку спочатку в проміжній елемент пам'яті поміщається результат обчислення виразу правій частині оператора присвоєння, а потім цей результат привласнюється змінною в лівій частині.

Наприклад, якщо в операторі присвоєння  $A = A + 5$ , змінна  $A$  до операції присвоєння містила значення 7, то після операції вона міститиме значення 12 (7+5).

Слід запам'ятати:

- Можна присвоювати будь-яку чисельну змінну (або вираз) будь-якої іншої змінної чисельного типу (або змінній типу Variant);
- Якщо привласнюється чисельний вираз змінній, що типізується, з меншою точністю (наприклад, Double - Long), VBA округляє значення виразу для збігу з точністю змінної, що набуває нового значення;
- Якщо змінною типу String привласнюється змінна типу Variant, що містить число, VBA автоматично перетворить це число в рядок.

Оператор додавання (+)

Оператора додавання виконує просте додавання. Обидва операнди мають бути чисельними виразами або рядками, які VBA може перетворити в число. Оператора додавання можна також використовувати для виконання арифметичних операцій з даними типу Date.

Оператор віднімання (-)

Оператор віднімання виконує два завдання: використовується для віднімання одного числа з іншого; позначає унарний мінус (це знак мінус, який поміщається перед числом для вказівки того, що це негативне число). Помістити унарний мінус перед змінною або виразом означає те ж, що помножити це число -1.

Оператор множення (\*)

Оператора множення перемножує два числа - результатом виразу множення є твір двох операндів. Обидва операнди у виразі множення мають бути чисельними виразами або рядками, які VBA може перетворити в число.

Оператор ділення (/)

Оператор ділення з плаваючою комою виконує звичайне арифметичне ділення своїх операндів.

Цілочисельне ділення (\)

Цілочисельне ділення відрізняється від ділення з плаваючою комою тим, що його результатом завжди є ціле число без дробової частини.

Піднесення до степені (^)

Оператор піднесення до степені підносить число до степені.

$5 ^ 3 =125$ .

Конструкція оператора With мови VBA.

Після ознайомлення з об'єктами, слід розглянути декілька конструкцій мови VBA, які значно полегшують роботу з об'єктами. Для спрощення розуміння використання оператора With, розглянемо приклад форматування діапазону комірок

Процедура форматування діапазону комірок:

```
Public Sub Format()
```

```
Range("B2:D15").NumberFormat = "#,##0.00"
```

```
Range("B2:D15").Font.Name = "MS San Serif"
```

```
Range("B2:D15").Font.FontStyle = Regular
```

```
Range("B2:D15").Font.Size = 13
```

```
Range("B2:D15").Font.Strikethrough = False
```

```
Range("B2:D15").Font.Superscript = False
```

```
Range("B2:D15").Font.Subscript = True
```

```
Range("B2:D15").Font.OutlineFont = True
```

```
Range("B2:D15").Font.Shadow = True
```

```
Range("B2:D15").Font.Underline = xlUnderlineStyleNone
```

```
Range("B2:D15").Font.ColorIndex = xlAutomatic
```

```
End Sub
```

При введенні коду процедури ми декілька разів набираємо Range("B2:D15"). Із подібних ситуацій VBA пропонує інший вихід – використання оператора With, який “прив’язує” властивості або методи до об’єкта.

Оператор With має наступний синтаксис:

With об'єкт

Оператори

End With

Лістинг наведений нижче демонструє той же код процедури, але з використанням оператора With, який спочатку “прив’язує” до об’єкта Range("B2:D15") властивість NumberFormat = "#,##0.00", а потім властивість Font через другий оператор With “прив’язується” до об’єкта Range("B2:D15").

Оператори With можна вкладати один в одного (допускається до семи вкладень).

Процедура форматування діапазону комірок із використанням оператора With:

```
Public Sub Format()
With Range("B2:D15")
.NumberFormat = "#,##0.00"
With .Font
.Name = "MS San Serif"
.FontStyle = Regular
.Size = 13
.Strikethrough = False
.Superscript = False
.Subscript = True
.OutlineFont = True
.Shadow = True
.Underline = xlUnderlineStyleNone
.ColorIndex = xlAutomatic
End With
End With
End Sub
```

Конструкція оператора For Each мови VBA.

Якщо оператор With використовується для виконання декількох дій для одного об'єкта, то оператор For Each використовується для виконання одного і того ж набору дій для декількох об'єктів. Оператор For Each, як правило, використовувати для перегляду елементів масиву або колекції.

Оператор For Each має наступний формат:

For Each елемент In колекція(масив)

....Next елемент

Елемент – це ідентифікатор змінної-інтерпретатора; колекція – ідентифікатор колекції або масиву.

У кодї процедури Колекції переглядаються аркуші активної книги і на екран по черзі виводиться № аркуша та його ім'я.

Приклад використання оператора For Each.

Public Sub Колекції()

Dim Аркуш As Worksheet

Dim Nomer As Integer

Nomer = 0

For Each Аркуш In ActiveWorkbook.Worksheets

Nomer = Nomer + 1

MsgBox ("Ім'я аркуша: ") & Nomer & " " & Аркуш.Name

Next Аркуш

End Sub

### 3.2. Алгоритм роботи тренажера

**Перший крок.** Користувач отримає наступну умову завдання, вибирається лише 1 варіант відповіді: «Навіщо призначені оператори присвоювання?»

1. Всі варіанти правильні; (вірна відповідь)
2. Запис результату обчислень до змінної;
3. Редагування значень.

Під час проходження тестування, при позитивній відповіді користувачу відкриється наступна умова завдання. При негативній відповіді, користувач отримає вікно з текстом: «Ваша відповідь невірна! Спробуйте ще раз!»

**Другий крок.** Користувач отримає наступну умову завдання, вибирається лише 1 варіант відповіді: «Що називається ітерацією?»

1. Процес виконання всіх операторів, ув'язнених у структуру циклу, кілька разів;
2. Процес виконання всіх операторів, укладених у структуру циклу, один раз; (вірна відповідь)
3. Процес виконання всіх операторів, укладених у структуру циклу, нескінченну кількість разів.

Під час проходження тестування, при позитивній відповіді користувачу відкриється наступна умова завдання. При негативній відповіді, користувач отримає вікно з текстом: «Ваша відповідь невірна! Спробуйте ще раз!»

**Третій крок.** Користувач отримає наступну умову завдання, вибирається лише 1 варіант відповіді: «Ключове слово VBA, що означає початок циклу: »

1. Const;
2. Case;
3. For (вірна відповідь)

Під час проходження тестування, при позитивній відповіді користувачу відкриється наступна умова завдання. При негативній відповіді, користувач отримає вікно з текстом: «Ваша відповідь невірна! Спробуйте ще раз!»

**Четвертий крок.** Користувач отримає наступну умову завдання, вибирається лише 1 варіант відповіді: «Виберіть із запропонованих варіантів оператор цілочислового ділення: »

1. \ ; (вірна відповідь)
2. / ;
3. | .

Під час проходження тестування, при позитивній відповіді користувачу відкриється наступна умова завдання. При негативній відповіді, користувач отримає вікно з текстом: «Ваша відповідь невірна! Спробуйте ще раз!»

**П'ятий крок.** Користувач отримає наступну умову завдання, вибирається лише 1 варіант відповіді: «Запис оператора множинного вибору до VBA починається зі службових слів?»

1. Case Else;
2. Select Case; (вірна відповідь)
3. Case of.

Під час проходження тестування, при позитивній відповіді користувачу відкриється наступна умова завдання. При негативній відповіді, користувач отримає вікно з текстом: «Ваша відповідь невірна! Спробуйте ще раз!»

**Шостий крок.** Користувач отримає наступну умову завдання, вибирається лише 1 варіант відповіді: «Який тип операцій дозволяє виконувати оператор Like?»

1. Цілий поділ;
2. Зведення в степінь;
3. Операції порівняння рядків. (вірна відповідь)

Під час проходження тестування, при позитивній відповіді користувачу відкриється наступна умова завдання. При негативній відповіді, користувач отримає вікно з текстом: «Ваша відповідь невірна! Спробуйте ще раз!»

**Сьомий крок.** Користувач отримає наступну умову завдання, вибирається лише 1 варіант відповіді: «Синтаксична конструкція лінійного безальтернативного оператора?»

1. If <умова> Then <оператор 1>; (вірна відповідь)
2. Case <умова>Else;
3. Case Else.

Під час проходження тестування, при позитивній відповіді користувачу відкриється наступна умова завдання. При негативній відповіді, користувач отримає вікно з текстом: «Ваша відповідь невірна! Спробуйте ще раз!»

**Восьмий крок.** Користувач отримає наступну умову завдання, вибирається лише 1 варіант відповіді: «Логічні оператори використовуються для: »

1. Створення складних критеріїв для прийняття рішень у процедурі;
2. Всі варіанти вірні; (вірна відповідь)

3. Створення умов, за яких група операторів має повторюватися.

Під час проходження тестування, при позитивній відповіді користувачу відкриється наступна умова завдання. При негативній відповіді, користувач отримає вікно з текстом: «Ваша відповідь невірна! Спробуйте ще раз!»

**Дев'ятий крок.** Користувач отримає наступну умову завдання, вибирається лише 1 варіант відповіді: «Синтаксис циклу з передумовою має вигляд: »

1. Do

<Тіло циклу>

Loop While <умова>;

2. Do

<Тіло циклу>

Loop Until <умова>;

3. Do While <умова>

<Тіло циклу>

Loop. (вірна відповідь)

Під час проходження тестування, при позитивній відповіді користувачу відкриється наступна умова завдання. При негативній відповіді, користувач отримає вікно з текстом: «Ваша відповідь невірна! Спробуйте ще раз!»

**Десятий крок.** Користувач отримає наступну умову завдання, вибирається лише 1 варіант відповіді: «Синтаксис циклу з постумовою має вигляд:»

1. Do

<Тіло циклу>

Loop Until <умова>; (вірна відповідь)

2. Do Until <умова>

<Тіло циклу>

Loop;

3. Do While <умова>

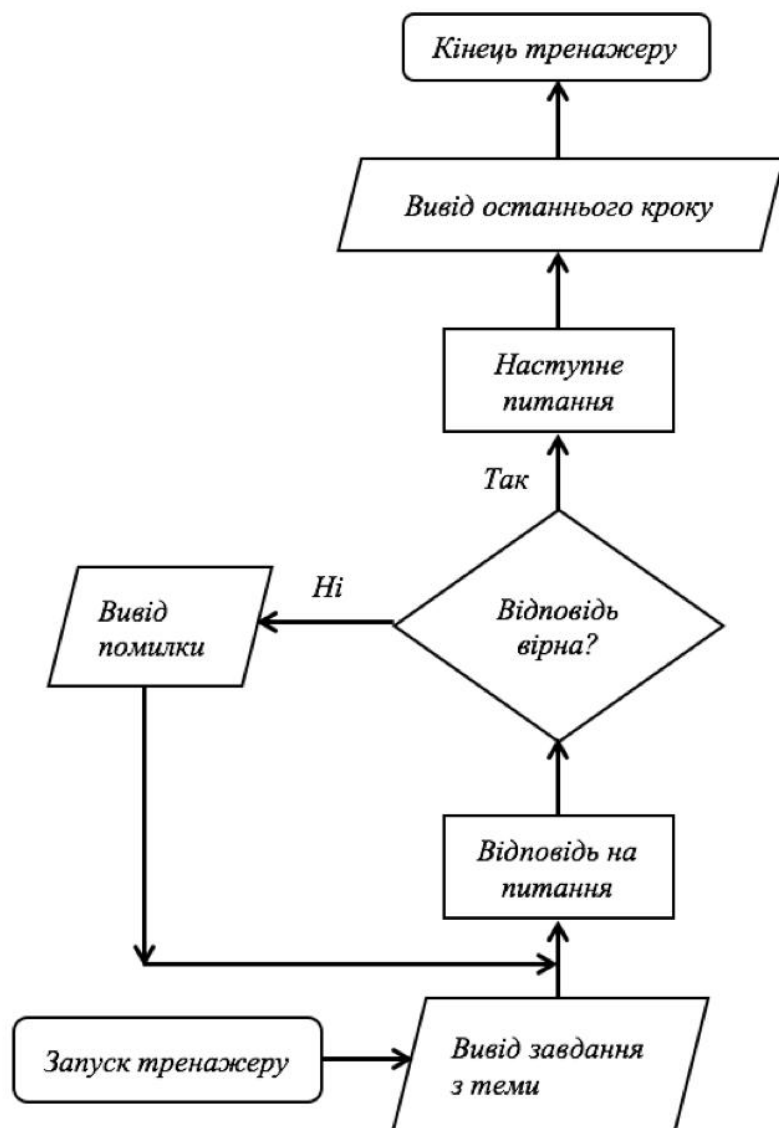
<Тіло циклу>

Loop.

Під час проходження тестування, при позитивній відповіді користувачу відкриється наступна умова завдання. При негативній відповіді, користувач отримає вікно з текстом: «Ваша відповідь невірна! Спробуйте ще раз!»

В кінці користувач переходить на форму з підсумками, на якій написано привітання: «Вітаємо! Ви успішно пройшли тренажер з теми «Оператори Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації».

### 3.3. Блок-схема тренажера



## РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

### 4.1. Обґрунтування вибору програмних засобів

Одним із засобів розробки вищезгаданих Windows-програм є Visual Basic (VB), застосування якого: по-перше, дозволяє у максимальному ступені абстрагуватися від мовних проблем; по-друге, знайомить із середовищем програмування, характерним для усіх засобів розроблення Microsoft. Освоєння ж мови реально можливе тільки в ході практичної роботи й самонавчання. В 1991 році під гаслом тепер і програмісти-початківці можуть легко створювати додатки для Windows" з'явилася перша версія нового інструментального засобу Microsoft Visual Basic. У той момент Microsoft досить скромно оцінювала можливості цієї системи, орієнтуючи її, насамперед, на категорію початківців і непрофесійних програмістів. Основне завдання тоді полягало в тому, щоб випустити на ринок простий і зручний інструмент розробки в новому середовищі Windows, програмування в якому було проблемним навіть для досвідчених фахівців. Тому VB версії 1.0 був схожий скоріше на діючий макет майбутнього середовища розробки, ніж на робочий інструмент. Однак уже тоді принципове нововведення VB полягало в реалізації ідей подійно керованого й візуального програмування в середовищі Window, які радикально відрізнялися від класичних схем розробки програм.

## 4.2. Опис програмної реалізації

Програмний продукт додатку тренажеру написано за допомогою програмного середовища Visual Studio (рис. 4.1)

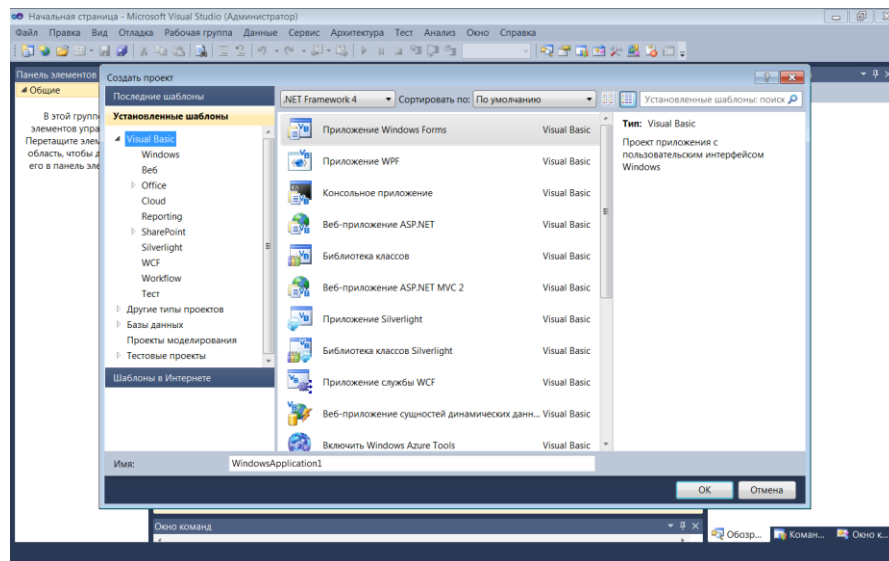


Рисунок 4.1 – огляд програмного середовища.

На початку створюється головне вікно з привітанням з однією кнопкою (рис. 4.2)

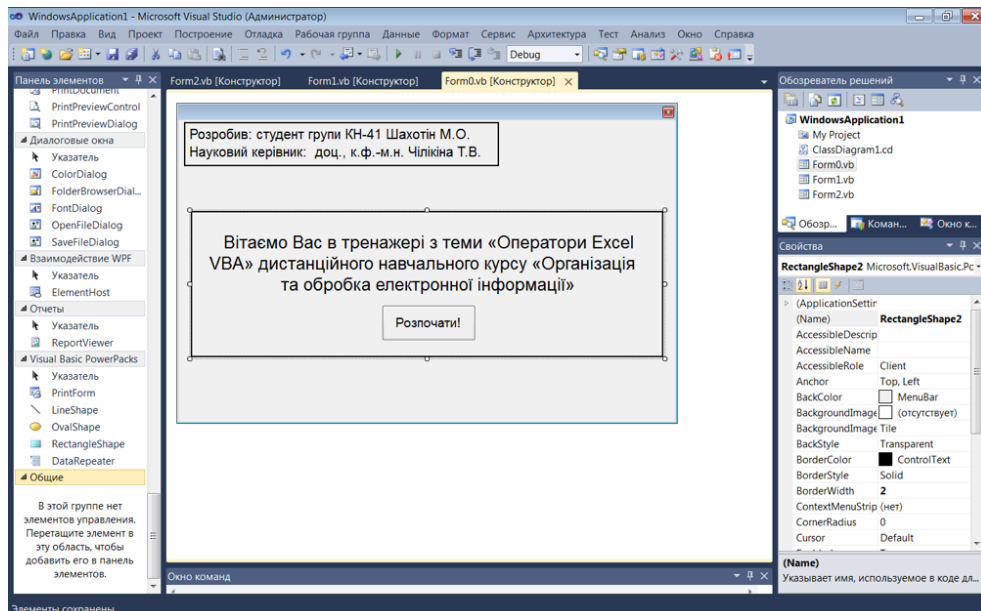


Рисунок 4.2 – головне вікно з привітанням.

Далі, після початкового вікна додатку розробляється наступне вікно з умовою першого завдання. (рис. 4.3)

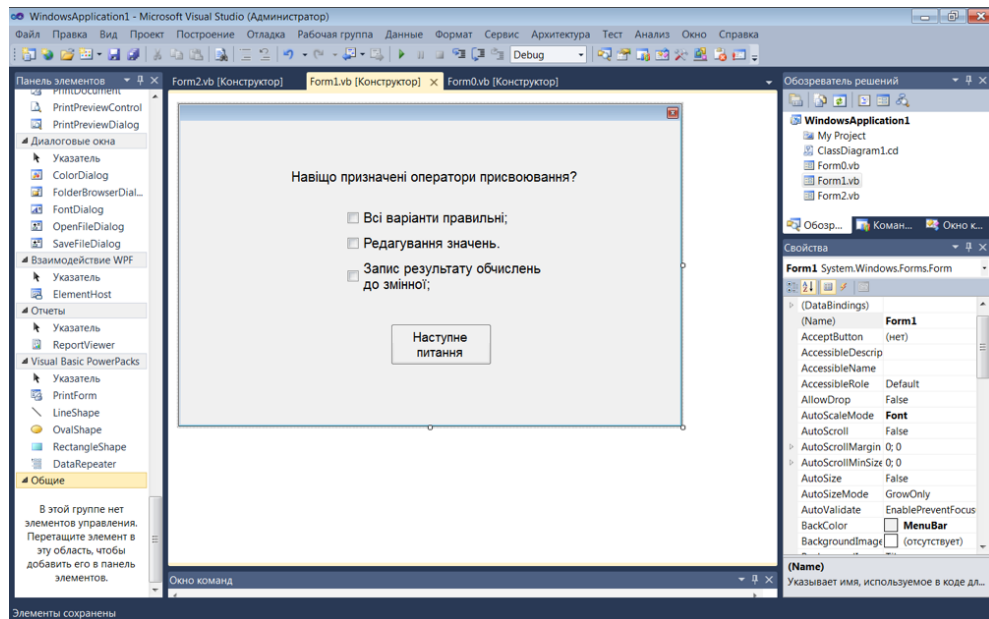


Рисунок 4.3 – умова першого завдання додатку.

Далі, треба програмно реалізувати умову наступного завдання. (рис. 4.4)

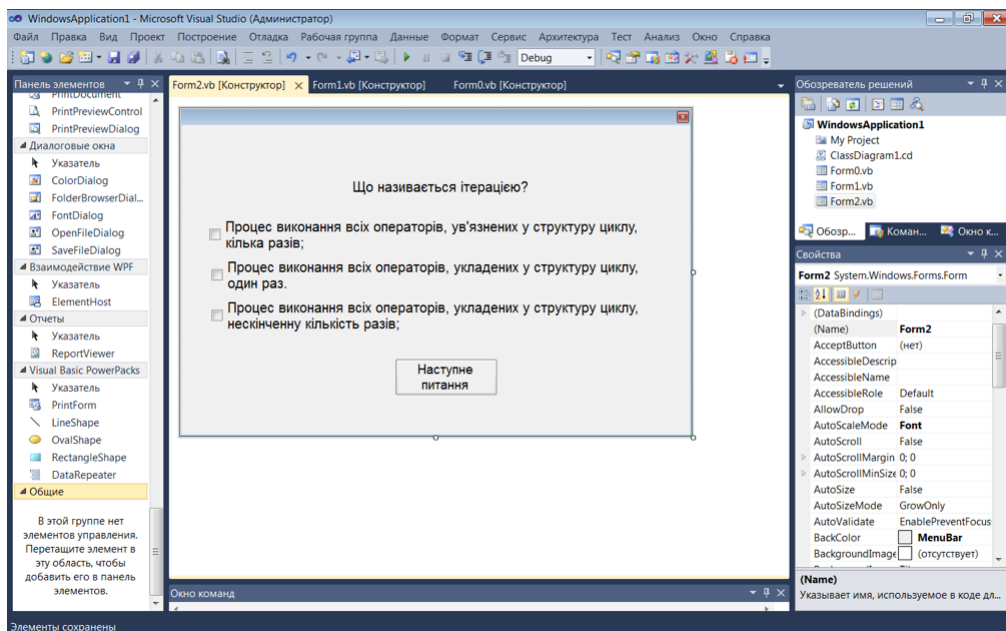


Рисунок 4.4 – умова наступного завдання.

Користувачу далі потрібно вибрати один варіанти відповіді та натиснути кнопку «Наступне питання», після вибору варіанту відповіді інші два варіанти переходять в неактивний стан, потім користувач отримує повідомлення, яке його інформує чи правильно він відповів на запитання, реалізовано відповідь на запитання наступним кодом:

```

1 Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated() _
2 Partial Class Form3
3     Inherits System.Windows.Forms.Form
4
5     'Форма переопределяет dispose для очистки списка компонентов.
6     <System.Diagnostics.DebuggerNonUserCode()> _
7     Protected Overrides Sub Dispose(ByVal disposing As Boolean)
8         Try
9             If disposing AndAlso components IsNot Nothing Then
10                components.Dispose()
11            End If
12        Finally
13            MyBase.Dispose(disposing)
14        End Try
15    End Sub
16
17    'Является обязательной для конструктора форм Windows Forms
18    Private components As System.ComponentModel.IContainer
19
20    'Примечание: следующая процедура является обязательной для конструктора форм Windows Forms
21    'Для ее изменения используйте конструктор форм Windows Form.
22    'Не изменяйте ее в редакторе исходного кода.
23    <System.Diagnostics.DebuggerStepThrough()> _
24    Private Sub InitializeComponent()
25        Me.Button1 = New System.Windows.Forms.Button()
26        Me.CheckBox1 = New System.Windows.Forms.CheckBox()
27        Me.CheckBox2 = New System.Windows.Forms.CheckBox()
28        Me.CheckBox3 = New System.Windows.Forms.CheckBox()
29        Me.Label2 = New System.Windows.Forms.Label()
30        Me.SuspendLayout()
31
32        'Button1
33
34        Me.Button1.BackColor = System.Drawing.SystemColors.MenuBar
35        Me.Button1.Cursor = System.Windows.Forms.Cursors.Hand
36        Me.Button1.Font = New System.Drawing.Font("Arial", 10.8!, System.Drawing.FontStyle.Regu

```

### 4.3. Опис роботи тренажеру

На початку роботи додатку користувачеві виводиться головне вікно (рис. 4.5)

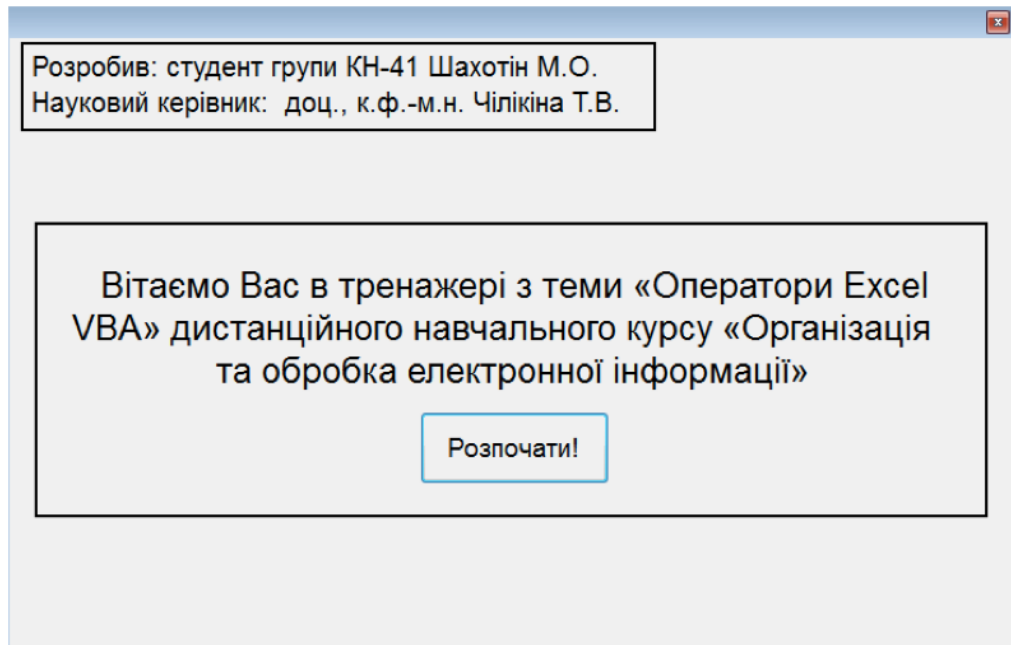


Рисунок 4.5 – головне вікно.

На головному вікнові необхідно клацнути кнопку початку. Потім виведеться інше вікно з умовою та завданням (рис. 4.6)

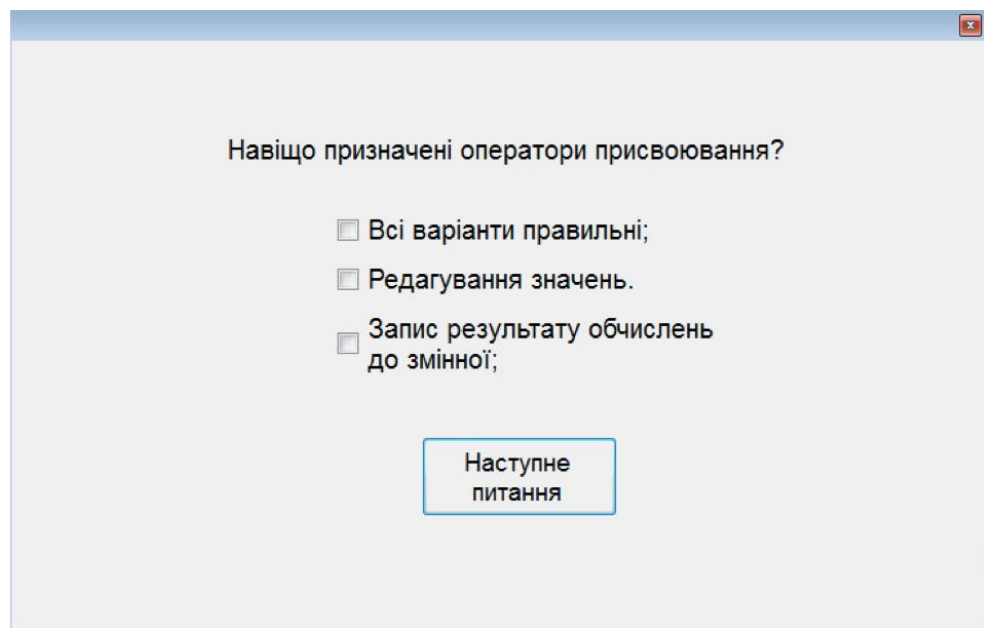


Рисунок 4.6 – перше вікно з умовою та завданням.

Користувачеві пропонується обрати варіант відповіді, яка на його думку правильна, зробити це можна клацнувши по будь-якому варіанту. Після вибору відповіді, виведеться вікно про правильність вибору відповіді (рис. 4.7)

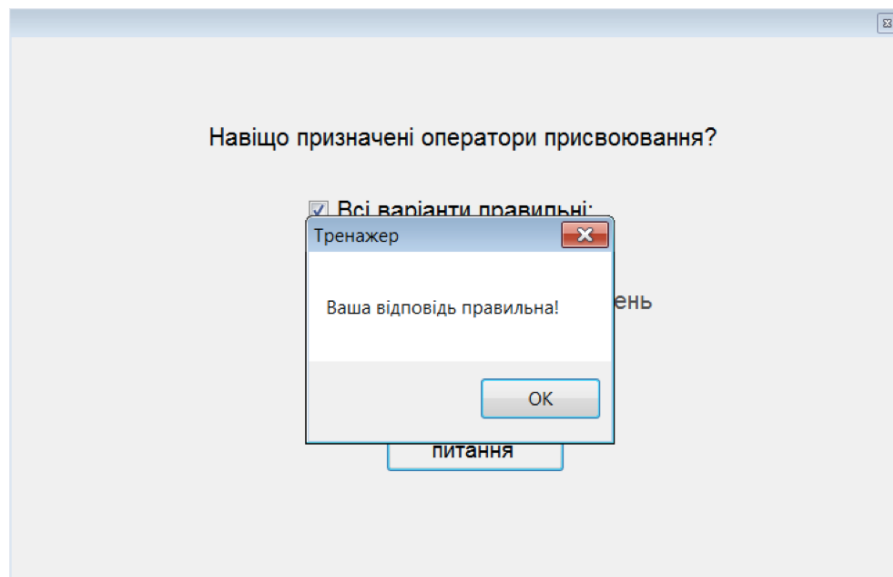


Рисунок 4.7 – вікно з інформацією про те що вибрана відповідь правильна.

Якщо користувач помилився, виведеться наступне вікно (рис. 4.8)

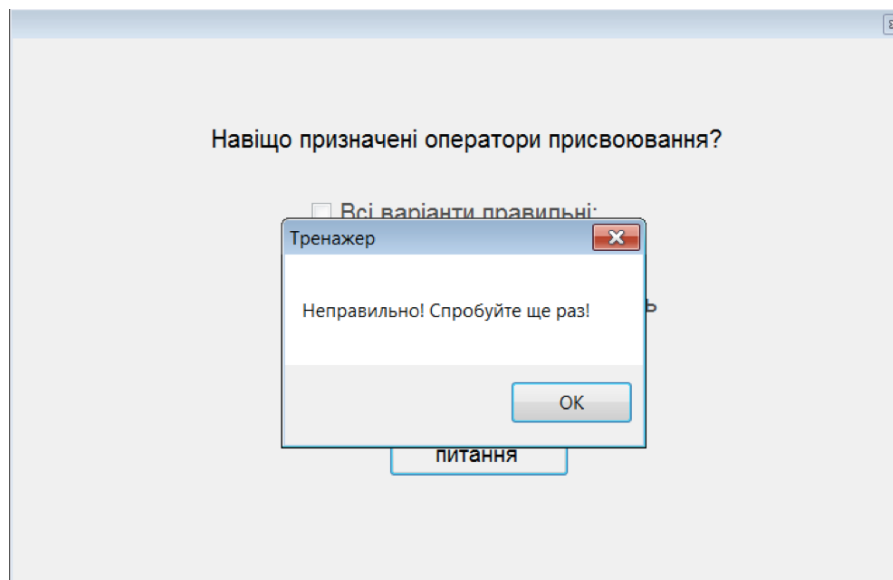
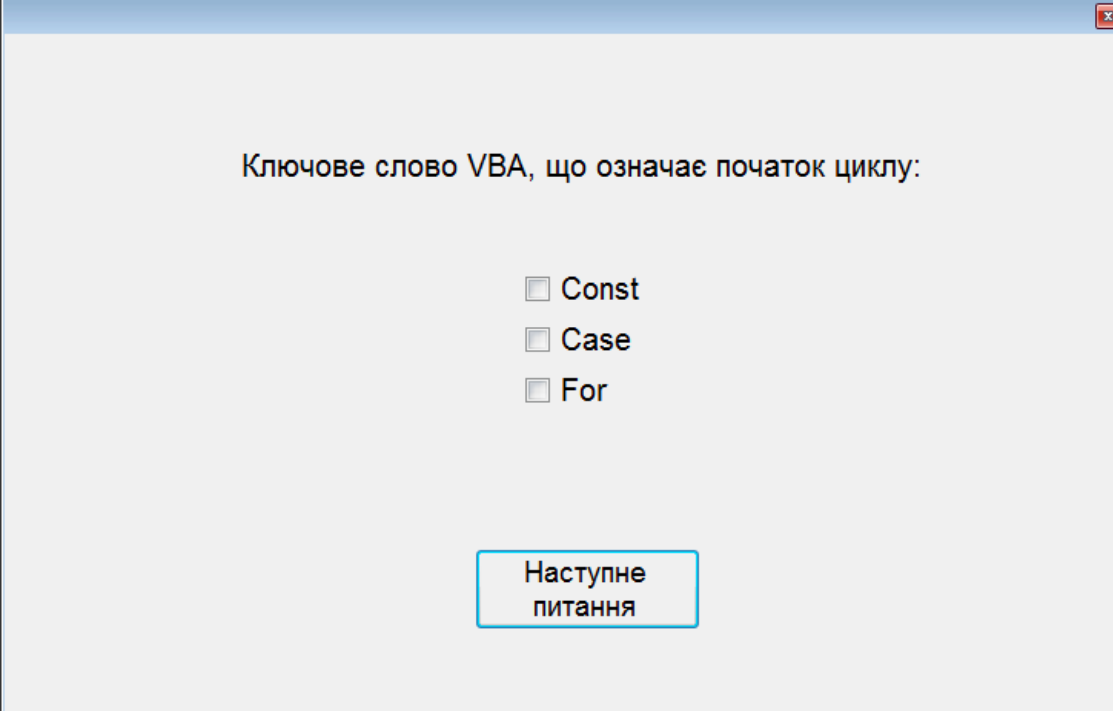


Рисунок 4.8 – вікно з інформацією про те що вибрана відповідь неправильна.

## Вивід завдання 3 (рис. 4.9)



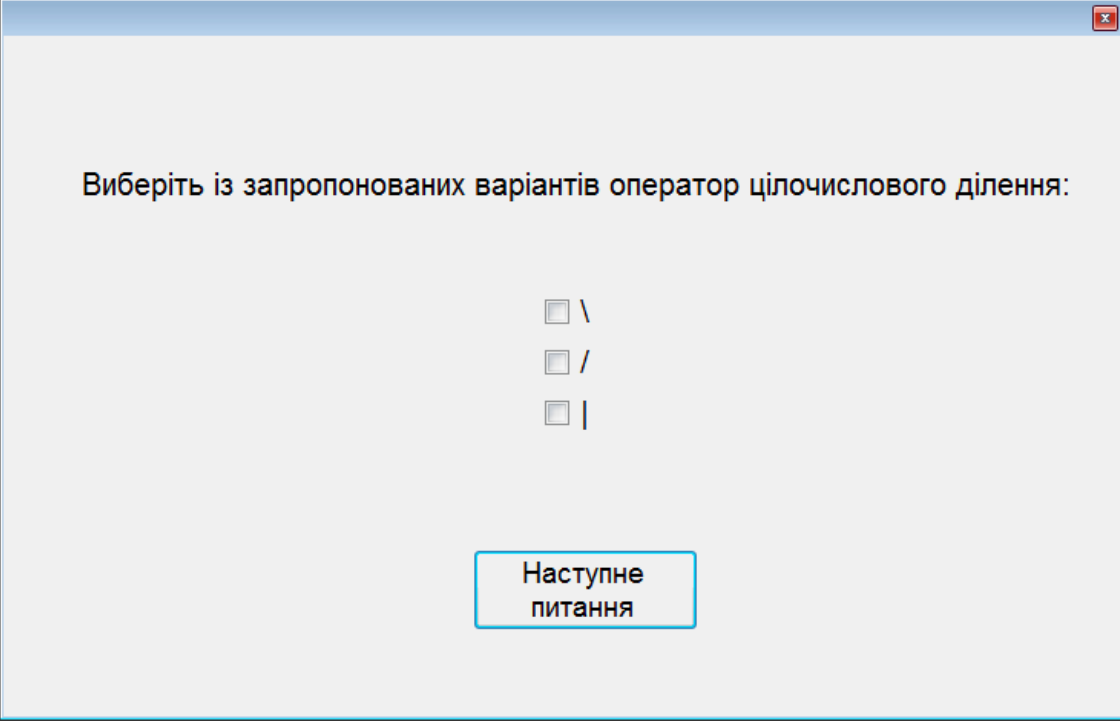
Ключове слово VBA, що означає початок циклу:

- Const
- Case
- For

Наступне питання

Рисунок 4.9 – форма з умовою завдання номер 3.

## Вивід завдання 4 (рис. 5.0)



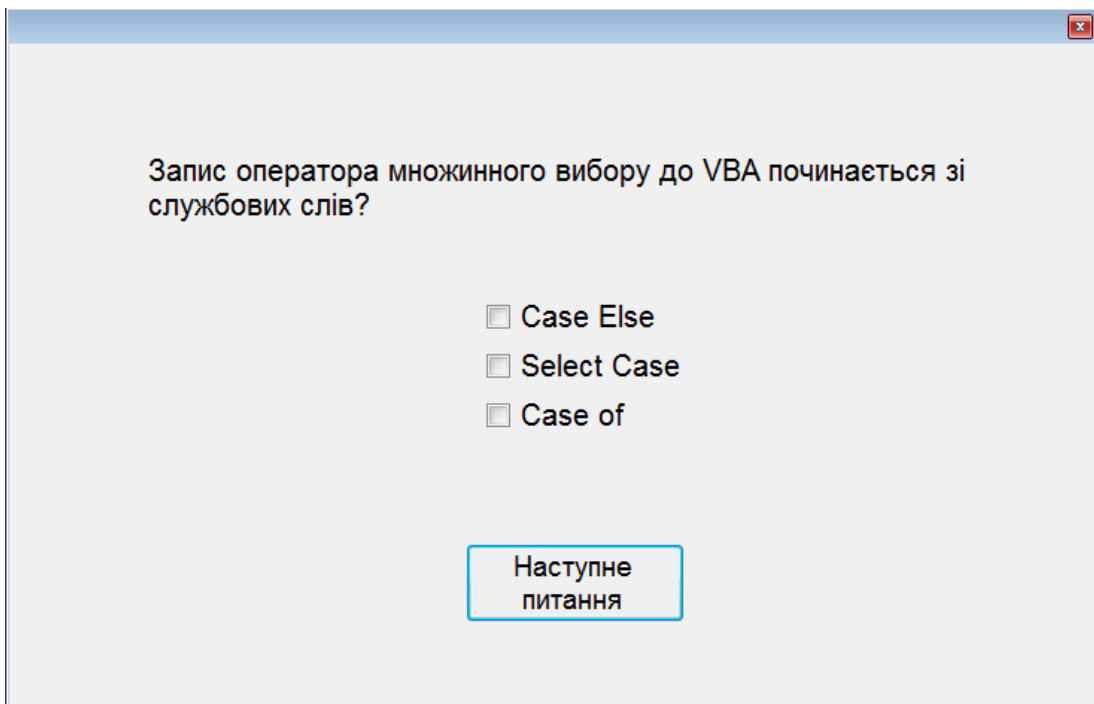
Виберіть із запропонованих варіантів оператор цілочислового ділення:

- \
- /
- |

Наступне питання

Рисунок 5.0 – форма з умовою завдання номер 4.

## Вивід завдання 5 (рис. 5.1)



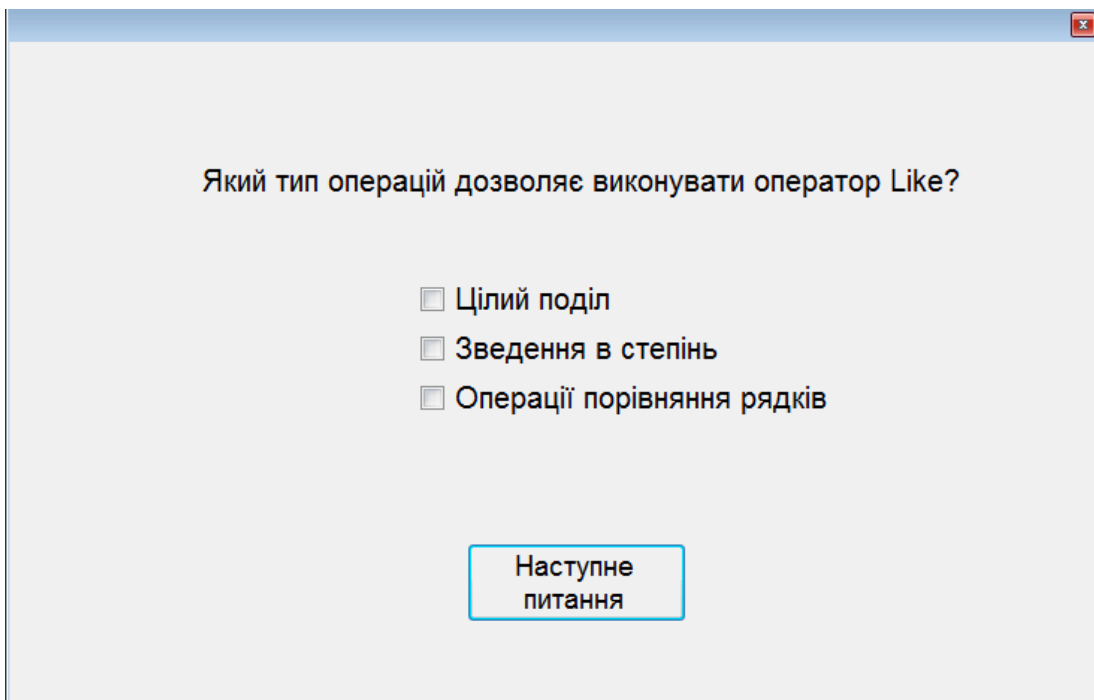
Запис оператора множинного вибору до VBA починається зі службових слів?

- Case Else
- Select Case
- Case of

Наступне питання

Рисунок 5.1 – форма з умовою завдання номер 5.

## Вивід завдання 6 (рис. 5.2)



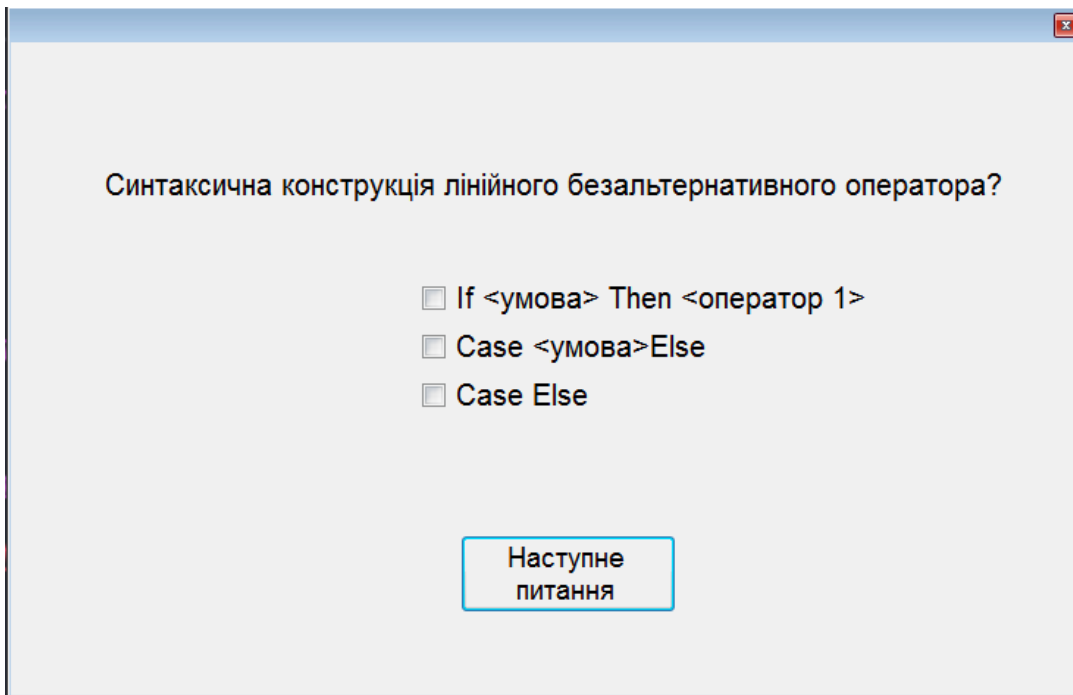
Який тип операцій дозволяє виконувати оператор Like?

- Цілий поділ
- Зведення в степінь
- Операції порівняння рядків

Наступне питання

Рисунок 5.2 – форма з умовою завдання номер 6.

## Вивід завдання 7 (рис. 5.3)



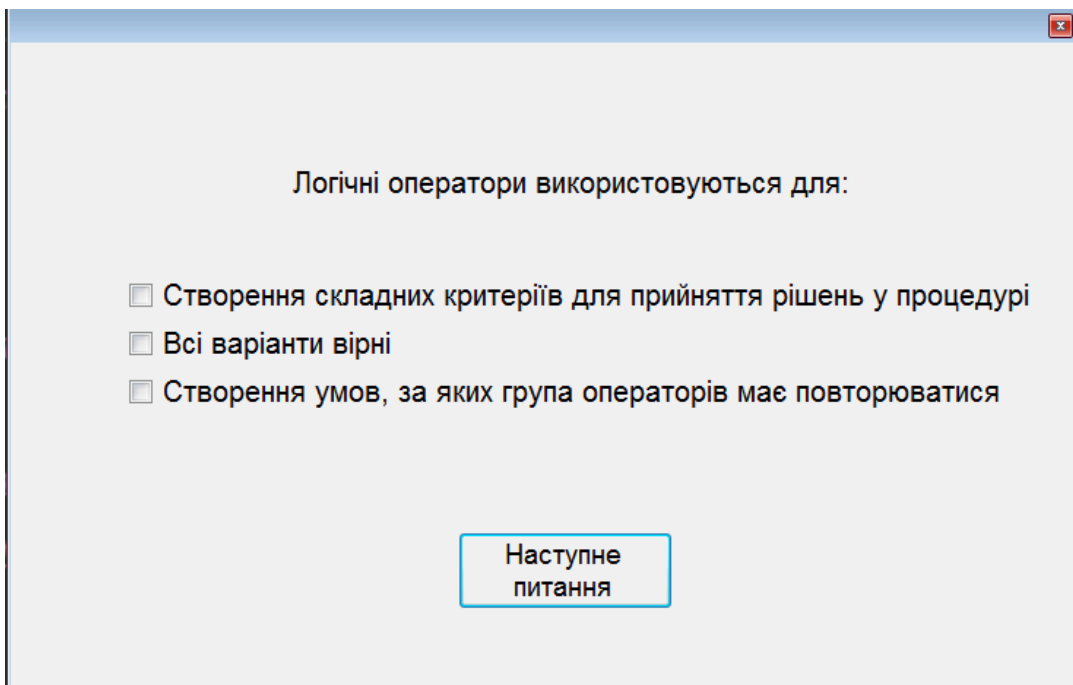
Синтаксична конструкція лінійного безальтернативного оператора?

- If <умова> Then <оператор 1>
- Case <умова>Else
- Case Else

Наступне питання

Рисунок 5.3 – форма з умовою завдання номер 7.

## Вивід завдання 8 (рис. 5.4)



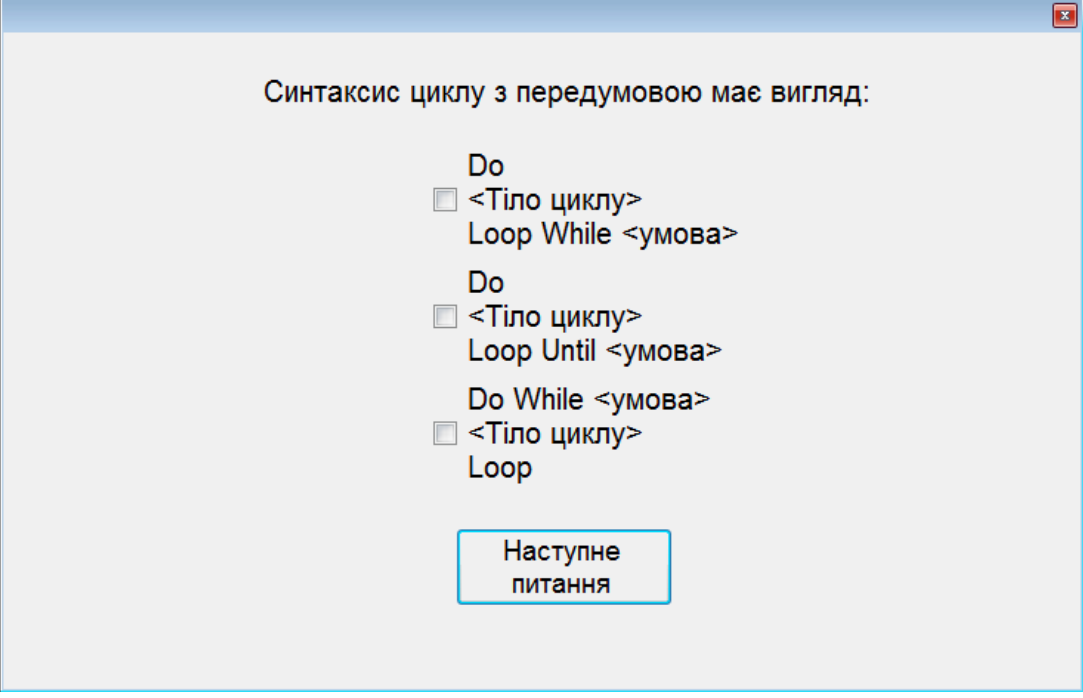
Логічні оператори використовуються для:

- Створення складних критеріїв для прийняття рішень у процедурі
- Всі варіанти вірні
- Створення умов, за яких група операторів має повторюватися

Наступне питання

Рисунок 5.4 – форма з умовою завдання номер 8.

Вивід завдання 9 (рис. 5.5)



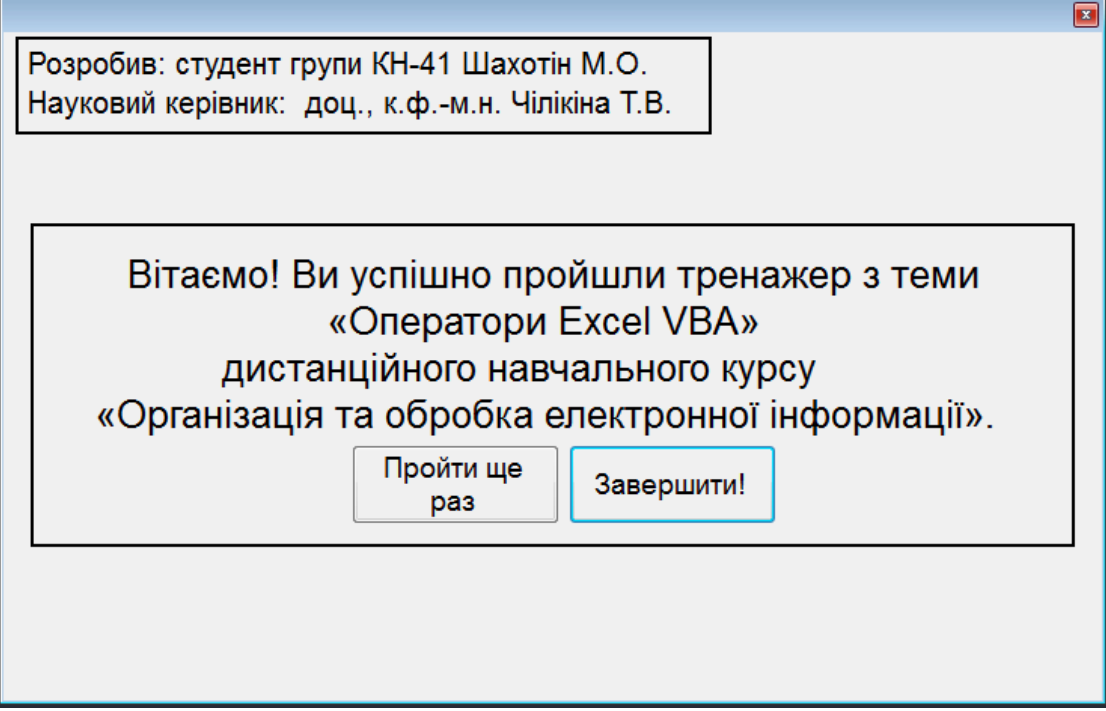
Синтаксис циклу з передумовою має вигляд:

- Do  
<Тіло циклу>  
Loop While <умова>
- Do  
<Тіло циклу>  
Loop Until <умова>
- Do While <умова>  
<Тіло циклу>  
Loop

Наступне питання

Рисунок 5.5 – форма з умовою завдання номер 9.

В підсумку користувач тренажеру отримує наступну форму (рис. 5.6)



Розробив: студент групи КН-41 Шахотін М.О.  
Науковий керівник: доц., к.ф.-м.н. Чілікіна Т.В.

Вітаємо! Ви успішно пройшли тренажер з теми  
«Оператори Excel VBA»  
дистанційного навчального курсу  
«Організація та обробка електронної інформації».

Пройти ще раз      Завершити!

Рисунок 5.6 – Остання форма підсумків.

## ВИСНОВКИ

Отже, розробка навчальних додатків тренажерів дійсно важлива в наш час. Дистанційна освіта грає важливу роль в періоди воєнного часу, коли доступ до традиційних форм освіти може бути обмеженим або неможливим.

Щоб покращити освітній процес рекомендовано використовувати записи відео-уроків викладачами, забезпечувати студентів необхідною технікою з доступом до інтернету, залучати волонтерські організації та програми з підтримки вищої освіти в умовах війни, скорочувати навантаження на онлайн-платформи та залучати спеціалістів ІТ сектору, а також залучати кошти посадових осіб та організацій, зацікавлених в підвищенні якості освіти, змінювати систему мотивації студентів, запроваджувати автоматичний контроль завдань в онлайн-системі.

Поставлене науковим керівником завдання виконано, запрограмовано тренажер з теми «Оператори Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації».

## СПИСОК ЛІТЕРАТУРИ

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. – 1 електрон. опт. диск (CVD-ROM).
2. Особливості дистанційного навчання студентів в умовах воєнного стану [Електронний ресурс]. – Режим доступу: <http://perspectives.pp.ua/index.php/pis/article/download/1394/1391>
3. Оператори мови VBA [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/7621813/page:38/>
4. Microsoft Visual Studio [Електронний ресурс]. – Режим доступу: [https://uk.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio)
5. Microsoft Visual Basic [Електронний ресурс]. – Режим доступу: [https://uk.wikipedia.org/wiki/Visual\\_Basic](https://uk.wikipedia.org/wiki/Visual_Basic)
6. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання: ДСТУ 7.1-2006. – [Чинний від 2007-07-01]. – К. : Держспоживстандарт України, 2007. – 47 с.
7. Програмування на Visual BASIC [Електронний ресурс]. – Режим доступу: [https://kursoviks.com.ua/bd\\_kompyuterni/article\\_post/65-lektsiya-osnovi-programuvannya-movoyu-visual-basic-6-0](https://kursoviks.com.ua/bd_kompyuterni/article_post/65-lektsiya-osnovi-programuvannya-movoyu-visual-basic-6-0)
8. Microsoft Visual Basic 2010 Step by Step [Електронний ресурс]. – Режим доступу: [https://books.google.com.ua/books?hl=uk&lr=&id=Ap9CAwAAQBAJ&oi=fnd&pg=PT18&dq=visual+basic&ots=n5NfMNuX8q&sig=8OFEk2tHiXjaKynsUtD6SgQKabY&redir\\_esc=y#v=onepage&q=visual%20basic&f=false](https://books.google.com.ua/books?hl=uk&lr=&id=Ap9CAwAAQBAJ&oi=fnd&pg=PT18&dq=visual+basic&ots=n5NfMNuX8q&sig=8OFEk2tHiXjaKynsUtD6SgQKabY&redir_esc=y#v=onepage&q=visual%20basic&f=false)
9. Randolph Visual Studio 2010 for professionals // Randolph, Nick, Gardner, David, Minutillo, Michael, Anderson, Chris.: Trans. with English - М.: LLC "I.D. Williams", 2011. - 1184 p.

10. Halvorson M. Microsoft Visual Basic 2005. Series "Step by step" / M. Halvorson. - ECOM Publishers. - 640 p.
11. Тестування з теми оператори VBA [Електронний ресурс]. – Режим доступу: <https://quizizz.com/admin/quiz/6379ffe1778b3a001e14415c/%D1%82%D0%B5%D1%81%D1%82-2-%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80%D1%8B-vba>
12. Інструкції (оператори) VBA [Електронний ресурс]. – Режим доступу: [https://sukachoff.ru/uk/ustrojstva/instrukcii-operatory-yazyka-vba-logicheskie-operator-vb-operator-visual-basic/](https://sukachoff.ru/uk/ustrojstva/instrukcii-operatory-yazyka-vba-logicheskie-operatory-vb-operator-visual-basic/)
13. Visual basic умова. Оператори умови в VBA. [Електронний ресурс]. – Режим доступу: <https://gadgets-room.ru/visual-basic-uslovie-operator-usloviya-v-vba-if-uslovie-then-vyrazhenie/>
14. Оператори мови vba та її керуючі конструкції [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/7621813/page:38/>
15. Самовчитель по excel vba [Електронний ресурс]. – Режим доступу: <https://ua.waykun.com/articles/samovchitel-po-excel-vba-operatori-umovi-v-vba.php>

## ДОДАТОК А. КОД ПРОГРАМИ

```
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
```

```
Partial Class Form0
```

```
    Inherits System.Windows.Forms.Form
```

```
    'Форма переопределяет dispose для очистки списка компонентов.
```

```
    <System.Diagnostics.DebuggerNonUserCode()> _
```

```
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
```

```
        Try
```

```
            If disposing AndAlso components IsNot Nothing Then
```

```
                components.Dispose()
```

```
            End If
```

```
        Finally
```

```
            MyBase.Dispose(disposing)
```

```
        End Try
```

```
    End Sub
```

```
    'Является обязательной для конструктора форм Windows Forms
```

```
    Private components As System.ComponentModel.IContainer
```

'Примечание: следующая процедура является обязательной для конструктора форм Windows Forms

```
'Для ее изменения используйте конструктор форм Windows Form.
```

```
'Не изменяйте ее в редакторе исходного кода.
```

```
<System.Diagnostics.DebuggerStepThrough()> _
```

```
    Private Sub InitializeComponent()
```

```
        Me.Button1 = New System.Windows.Forms.Button()
```

```
        Me.Label1 = New System.Windows.Forms.Label()
```

```

Me.Label2 = New System.Windows.Forms.Label()
Me.Label3 = New System.Windows.Forms.Label()
Me.ShapeContainer1 = New
Microsoft.VisualBasic.PowerPacks.ShapeContainer()
Me.RectangleShape1 = New
Microsoft.VisualBasic.PowerPacks.RectangleShape()
Me.RectangleShape2 = New
Microsoft.VisualBasic.PowerPacks.RectangleShape()
Me.SuspendLayout()
'
'Button1
'
Me.Button1.BackColor = System.Drawing.SystemColors.MenuBar
Me.Button1.Cursor = System.Windows.Forms.Cursors.Hand
Me.Button1.Font = New System.Drawing.Font("Arial", 10.8!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(204, Byte))
Me.Button1.ForeColor =
System.Drawing.SystemColors.ActiveCaptionText
Me.Button1.Location = New System.Drawing.Point(298, 272)
Me.Button1.Name = "Button1"
Me.Button1.Size = New System.Drawing.Size(138, 53)
Me.Button1.TabIndex = 0
Me.Button1.Text = "Розпочати!"
Me.Button1.UseVisualStyleBackColor = False
'
'Label1
'
Me.Label1.AutoSize = True

```

```

Me.Label1.Font = New System.Drawing.Font("Arial", 15.2!)
Me.Label1.ForeColor =
System.Drawing.SystemColors.ActiveCaptionText
Me.Label1.Location = New System.Drawing.Point(32, 164)
Me.Label1.Name = "Label1"
Me.Label1.Size = New System.Drawing.Size(653, 93)
Me.Label1.TabIndex = 1
Me.Label1.Text = " Вітаємо Вас в тренажері з теми «Оператори
Excel " & Global.Microsoft.VisualBasic.ChrW(13) &
Global.Microsoft.VisualBasic.ChrW(10) & " VBA» дистанційного навчаль"
& _
"ного курсу «Організація " &
Global.Microsoft.VisualBasic.ChrW(13) &
Global.Microsoft.VisualBasic.ChrW(10) & " та обробка
електронної інформації»"
'
'Label2
'
Me.Label2.AutoSize = True
Me.Label2.Font = New System.Drawing.Font("Arial", 12.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(204, Byte))
Me.Label2.ForeColor =
System.Drawing.SystemColors.ActiveCaptionText
Me.Label2.Location = New System.Drawing.Point(12, 9)
Me.Label2.Name = "Label2"
Me.Label2.Size = New System.Drawing.Size(422, 23)
Me.Label2.TabIndex = 2
Me.Label2.Text = "Розробив: студент групи КН-41 Шахотін М.О."

```

```

',
'Label3
',
Me.Label3.AutoSize = True
Me.Label3.Font = New System.Drawing.Font("Arial", 12.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(204, Byte))
Me.Label3.ForeColor =
System.Drawing.SystemColors.ActiveCaptionText
Me.Label3.Location = New System.Drawing.Point(12, 35)
Me.Label3.Name = "Label3"
Me.Label3.Size = New System.Drawing.Size(438, 23)
Me.Label3.TabIndex = 3
Me.Label3.Text = "Науковий керівник: доц., к.ф.-м.н. Чілікіна Т.В."
',
'ShapeContainer1
',
Me.ShapeContainer1.Location = New System.Drawing.Point(0, 0)
Me.ShapeContainer1.Margin = New System.Windows.Forms.Padding(0)
Me.ShapeContainer1.Name = "ShapeContainer1"
Me.ShapeContainer1.Shapes.AddRange(New
Microsoft.VisualBasic.PowerPacks.Shape() {Me.RectangleShape2,
Me.RectangleShape1 })
Me.ShapeContainer1.Size = New System.Drawing.Size(729, 444)
Me.ShapeContainer1.TabIndex = 4
Me.ShapeContainer1.TabStop = False
',
'RectangleShape1
',

```

```

Me.RectangleShape1.BorderWidth = 2
Me.RectangleShape1.Location = New System.Drawing.Point(9, 4)
Me.RectangleShape1.Name = "RectangleShape1"
Me.RectangleShape1.Size = New System.Drawing.Size(460, 63)
'
'RectangleShape2
'
Me.RectangleShape2.BorderWidth = 2
Me.RectangleShape2.Location = New System.Drawing.Point(19, 135)
Me.RectangleShape2.Name = "RectangleShape2"
Me.RectangleShape2.Size = New System.Drawing.Size(691, 213)
'
'Form0
'
Me.AutoScaleDimensions = New System.Drawing.SizeF(8.0!, 16.0!)
Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
Me.BackColor = System.Drawing.SystemColors.MenuBar
Me.ClientSize = New System.Drawing.Size(729, 444)
Me.Controls.Add(Me.Label3)
Me.Controls.Add(Me.Label2)
Me.Controls.Add(Me.Label1)
Me.Controls.Add(Me.Button1)
Me.Controls.Add(Me.ShapeContainer1)
Me.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.FixedToolWindow
Me.Name = "Form0"
Me.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen
Me.ResumeLayout(False)

```

```
Me.PerformLayout()
```

```
End Sub
```

```
Friend WithEvents Button1 As System.Windows.Forms.Button
```

```
Friend WithEvents Label1 As System.Windows.Forms.Label
```

```
Friend WithEvents Label2 As System.Windows.Forms.Label
```

```
Friend WithEvents Label3 As System.Windows.Forms.Label
```

```
Friend WithEvents ShapeContainer1 As
```

```
Microsoft.VisualBasic.PowerPacks.ShapeContainer
```

```
Friend WithEvents RectangleShape2 As
```

```
Microsoft.VisualBasic.PowerPacks.RectangleShape
```

```
Friend WithEvents RectangleShape1 As
```

```
Microsoft.VisualBasic.PowerPacks.RectangleShape
```

```
End Class
```