

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»
ІНСТИТУТ ЕКОНОМІКИ, УПРАВЛІННЯ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
ФАКУЛЬТЕТ ЕКОНОМІКИ І МЕНЕДЖМЕНТУ
ФОРМА НАВЧАННЯ ДЕННА
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ
ІНФОРМАТИКИ**

Допускається до захисту

Завідувач кафедри _____ О.О. Ємець
(підпис)

« _____ » _____ 2020 р.

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО ДИПЛОМНОЇ РОБОТИ**

на тему

**Тренажер з теми «Побудова блок-схем алгоритмів циклічної структури на прикладі
циклу «do...while» дистанційного навчального курсу «Інформатика» та розробка
його програмного забезпечення
зі спеціальності 122 «Комп'ютерні науки»**

Виконавець роботи Недбайло Ярослав Ігорович

_____ «__» _____ 2020р.
(підпис)

Науковий керівник проф. к.ф. – м.н. Є. М. Ємець

_____ «__» _____ 2020р.
(підпис).

ПОЛТАВА 2020 р.

Зміст

Вступ.....	3
1. Постановка задачі.....	Ошибка! Закладка не определена.
2. Інформаційний огляд.....	6
.....	
.....	

Вступ

Для реалізації своїх функцій у роботі, студент має під час навчання набути певних практичних навичок та теоретичних знань. Важливими умовами підвищення розвитку теорії і практики є створення системи завдань, їх послідовність, різноманітність та складність.

Комп'ютерний тренажер – це навчально-тренувальний пристрій, який симулює ситуацію, створює обставини наближені до реальних. Ця комп'ютерна програма створена щоб надати студенту певні уміння та навички.

Метою роботи є побудова тренажеру з теми «Побудова блок-схем алгоритмів циклічної структури на прикладі циклу «do...while» дистанційного навчального курсу «Інформатика» та розробка його програмного забезпечення.

Об'єктом розробки є процес дистанційного навчання.

Предмет розробки – програмний продукт, що реалізує тренажер для закріплення знань із використання циклу do...while.

Головне завдання – розробити алгоритм роботи тренажера, по якому потрібно розробити навчальний тренажер циклічної структури з використанням циклу «do...while», як складову дистанційного курсу «Інформатика».

При реалізації комп'ютерної програми використано мову програмування C++ і середовище розробки Visual Studio 2013.

Робота складається з чотирьох розділів. У першому розділі розглянуто постановку задачі для реалізації тренажера. У другому розділі описано огляд циклів мови програмування C++, їх відмінності та особливості. У третьому розділі представлено теоретичний матеріал з циклу do...while, наведено приклади застосування та описано алгоритм роботи тренажеру. У четвертому розділі – описано обґрунтування вибору програмних засобів, процес програмної реалізації, саму програму і інструкцію по використанню тренажера.

1. ПОСТАНОВКА ЗАДАЧІ

Оскільки потрібно розробити навчальний тренажер циклу `do...while`, як складову дистанційного курсу «Інформатика», то потрібно виконати наступні умови

- розглянути цикли у мові програмування C++;
- розглянути теоретичні відомості про цикли;
- розробити алгоритм тренажера, що дозволить закріпити знання з теми «Побудова блок-схем алгоритмів циклічної структури на прикладі циклу `do...while`»;
- обґрунтувати вибір програмного забезпечення;
- описати процес програмної реалізації;
- описати роботу тренажера.

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

Інколи необхідно щоб программа повторювала одну і ту ж дію декілька разів. Для цього використовують цикли.

Цикл for

Найуніверсальніший з усіх трьох операторів циклу мови програмування C++. Якщо потрібно виконати точку кількість дій(ітерацій), то цикл for підходить якнайкраще.

Загальний формат запису циклу for для багатократного виконання одного оператора має такий вигляд:

```
for(ініціалізація; вираз; інкремент) оператор;
```

Якщо цикл for призначений для багатократного виконання не одного оператора, а операторного блоку, то його загальний формат має такий вигляд:

```
for(ініціалізація; вираз; інкремент)
```

```
{
```

```
    послідовність операторів
```

```
}
```

Елемент ініціалізація зазвичай є оператором присвоєння, який встановлює керівній змінній циклу початкове значення, що дорівнює нулю. Ця змінна діє як лічильник, який керує роботою циклу. Елемент вираз є умовним оператором, у якому тестується значення керівної змінної циклу. Результат цього тестування визначає, виконається цикл for ще раз чи ні. Елемент інкремент – вираз, який визначає, як змінюється значення керівної змінної циклу після

кожної ітерації. Всі ці елементи циклу `for` повинні відділятися крапкою з комою.

Цикл `for` виконуватиметься доти, доки обчислення елемента виразу дає істинний результат. Як тільки цей умовний вираз стане помилковим, цикл завершиться, а виконання програми продовжиться з оператора, що є наступним за циклом `for`.

Цикл `while`

Ще одною формою оператора циклу є оператор `while`. Його синтаксис у загальному вигляді наступний:

`while (умова)`

`оператор`

Умова циклу – це вираз, що приймає логічне значення "істина" або "фальш". Виконання оператора повторюється доти, доки значенням умови є `true` (істина). Умова обчислюється заново перед кожною ітерацією.

Цикл `do while`

Третьою формою оператора циклу є цикл з післяумовою `do while`. Він має синтаксис:

`do { оператори } while (умова);`

Єдина і головна відмінність від попередньої форми циклу `while` полягає у тому, що умова перевіряється після виконання тіла циклу.

Оператор `break`

За допомогою оператора `break` можна організувати негайний вихід з тіла циклу, знехтувавши виконанням коду програми, що залишився в його тілі і перевіркою умовного виразу. Завдяки виявленню усередині циклу оператора `break` цикл завершується, а керування передається оператору, що є наступним після циклу.

Оператор continue

Цей оператор слугує для пропуску решти виконуваної частини операторів циклу, що міститься безпосередньо після його виклику. Якщо умовою циклу допускається нова ітерація, то вона виконується, в іншому випадку цикл завершується.

Оператор goto

Послідовність виконання операторів у програмі можна змінити за допомогою оператора goto.

Він має синтаксис:

```
goto мітка;
```

Оператор goto вимагає наявності у програмі мітки.

Мітка – дійсний у мові програмування C++ ідентифікатор, за яким поставлено двокрапку.

У процесі виконання оператора goto керування програмою передається оператору, вказаному за допомогою мітки.

Оператор return

Оператор return завершує виконання функції й повертає керування в ту точку програми, звідки вона була викликана.

Його синтаксис:

```
return вираз;
```

У даному випадку вираз – це результат функції. Якщо функція не повертає жодного значення, то оператор повернення має форму:

```
return;
```

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1 Приклад застосування циклу do while

Отже цикл do while – цикл з постумовою, і якщо умова циклу завідомо хибна, то хоча б один раз блок операторів в тілі циклу do while виконається. Розглянемо приклад з використанням цього циклу:

Знайти значення суми: $S = 1 + 3 + \dots + 99$

Фрагмент коду, що розв'язує задачу.

```
// s = 1 + 3 + ... + 99
```

```
int t;
```

```
int s;
```

```
s = 0;
```

```
t = 1;
```

```
do
```

```
{
```

```
    s = s + t;
```

```
    t = t + 2;
```

```
}
```

```
while (t<=99);
```

```
// s = 2500
```


3.2 Алгоритм роботи тренажеру

4. ПРАКТИЧНА ЧАСТИНА

4.1 Обґрунтування вибору програмних засобів

C++ (Сі-плюс-плюс) — мова програмування високого рівня з підтримкою кількох парадигм програмування: об'єктно-орієнтованої, узагальненої та процедурної. Розроблена Б'ярном Страуструпом в AT&T Bell Laboratories 1979 року та початково отримала назву «Сі з класами». Згодом Страуструп перейменував мову на C++ у 1983 р. Базується на мові програмування С. Вперше описана стандартом ISO/IEC 14882:1998, найбільш актуальним же є стандарт ISO/IEC 14882:2014.

У 1990-х роках C++ стала однією з найуживаніших мов програмування загального призначення, має з широкій інструментарій та функції. На ній створені такі проекти як Microsoft Windows і Google Chrome. Мову використовують для системного програмування, розробки програмного забезпечення, написання драйверів, потужних серверних та клієнтських програм, у банківській сфері, торгівлі, а також для розробки розважальних програм, наприклад, відеоігор. C++ суттєво вплинула на інші популярні сьогодні мови програмування: C# та Java.

Нижче наведена таблиця з порівняннями одних із найпопулярніших мов програмування

Мова	Імперативне	Об'єктно-орієнтоване	Функційне	Процедурне	Узагальнене	Рефлексивне	Подійно-орієнтоване
C	+			+			
C++	+	+	+	+	+		
C#	+	+	+		+	+	+
Fortran	+	+		+	+		
Haskell			+		+		
Java	+	+			+	+	
JavaScript	+	+	+			+	
Lisp			+				
PHP	+	+		+		+	
Python	+	+	+			+	

4.2 Опис процесу програмної реалізації

Опис теоретичної частини

```
#pragma once
#include "Form1.h"
#include "MyForm1.h"
```

Підключувані бібліотеки.

```
namespace Project {
```

```
    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;
```

```
    /// <summary>
```

```
    /// Сводка для MyForm
```

```
    /// </summary>
```

```
    public ref class MyForm : public System::Windows::Forms::Form
```

```
    {
```

```
    public:
```

```
        MyForm(void)
```

```
        {
```

```
            InitializeComponent();
```

```
            //
```

```
            //TODO: додайте код конструктора
```

```
            //
```

```
        }
```

```
    protected:
```

```
        /// <summary>
```

```
        /// Освободить все используемые ресурсы.
```

```
        /// </summary>
```

```
        ~MyForm()
```

```
        {
```

```
            if (components)
```

```
            {
```

```
                delete components;
```

```
            }
```

```
        }
```

Компоненти форми.

```
    private: System::Windows::Forms::Button^ button2;
```

```
protected:
private: System::Windows::Forms::Button^ button1;
private: System::Windows::Forms::Label^ label2;
private: System::Windows::Forms::PictureBox^ pictureBox1;
public: System::Windows::Forms::TextBox^ textBox1;
private:
private: System::Windows::Forms::Label^ label1;
private: System::Windows::Forms::PictureBox^ pictureBox2;
private: System::Windows::Forms::Button^ button3;
private: System::Windows::Forms::Button^ button4;
public:
```

```
private:
    /// <summary>
    /// Обязательная переменная конструктора.
    /// </summary>
    System::ComponentModel::Container ^components;
```

#pragma region Windows Form Designer generated code

```
/// <summary>
/// Требуемый метод для поддержки конструктора — не изменяйте
/// содержимое этого метода с помощью редактора кода.
/// </summary>
```

Опис компонентів.

```
void InitializeComponent(void)
{
    System::ComponentModel::ComponentResourceManager^ resources
= (gcnew System::ComponentModel::ComponentResourceManager(MyForm::typeid));
    this->button2 = (gcnew System::Windows::Forms::Button());
    this->button1 = (gcnew System::Windows::Forms::Button());
    this->label2 = (gcnew System::Windows::Forms::Label());
    this->pictureBox1 = (gcnew
System::Windows::Forms::PictureBox());
    this->textBox1 = (gcnew System::Windows::Forms::TextBox());
    this->label1 = (gcnew System::Windows::Forms::Label());
    this->pictureBox2 = (gcnew
System::Windows::Forms::PictureBox());
    this->button3 = (gcnew System::Windows::Forms::Button());
    this->button4 = (gcnew System::Windows::Forms::Button());
    (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox1))->BeginInit();
    (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox2))->BeginInit();
    this->SuspendLayout();
```

```

//
// button2
//
this->button2->BackColor =
System::Drawing::SystemColors::HotTrack;
this->button2->ForeColor =
System::Drawing::SystemColors::ButtonFace;
this->button2->Location = System::Drawing::Point(225, 450);
this->button2->Name = L"button2";
this->button2->Size = System::Drawing::Size(133, 53);
this->button2->TabIndex = 11;
this->button2->Text = L"Повернутися";
this->button2->UseVisualStyleBackColor = false;
this->button2->Click += gcnew System::EventHandler(this,
&MyForm::button2_Click);
//
// button1
//
this->button1->BackColor =
System::Drawing::SystemColors::HotTrack;
this->button1->ForeColor =
System::Drawing::SystemColors::ButtonFace;
this->button1->Location = System::Drawing::Point(428, 450);
this->button1->Name = L"button1 ";
this->button1->Size = System::Drawing::Size(116, 53);
this->button1->TabIndex = 10;
this->button1->Text = L"Продовжити";
this->button1->UseVisualStyleBackColor = false;
this->button1->Click += gcnew System::EventHandler(this,
&MyForm::button1_Click);
//
// label2
//
this->label2->AutoSize = true;
this->label2->Location = System::Drawing::Point(237, 322);
this->label2->Name = L"label2";
this->label2->Size = System::Drawing::Size(0, 13);
this->label2->TabIndex = 9;
//
// pictureBox1
//
this->pictureBox1->Image =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"pictureBox1.Image")));

```

```

this->pictureBox1->Location = System::Drawing::Point(-1, 231);
this->pictureBox1->Name = L"pictureBox1";
this->pictureBox1->Size = System::Drawing::Size(350, 92);
this->pictureBox1->TabIndex = 8;
this->pictureBox1->TabStop = false;
//
// textBox1
//
this->textBox1->Enabled = false;
this->textBox1->Font = (gcnew System::Drawing::Font(L"Times
New Roman", 12.25F));
this->textBox1->Location = System::Drawing::Point(20, 61);
this->textBox1->Multiline = true;
this->textBox1->Name = L"textBox1";
this->textBox1->Size = System::Drawing::Size(728, 164);
this->textBox1->TabIndex = 7;
//
// label1
//
this->label1->AutoSize = true;
this->label1->Font = (gcnew System::Drawing::Font(L"Microsoft
Sans Serif", 25.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
    static_cast<System::Byte>(204)));
this->label1->Location = System::Drawing::Point(270, 9);
this->label1->Name = L"label1";
this->label1->Size = System::Drawing::Size(185, 39);
this->label1->TabIndex = 6;
this->label1->Text = L"Цикл while";
//
// pictureBox2
//
this->pictureBox2->Image =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"pictureBox2.Image")));
this->pictureBox2->Location = System::Drawing::Point(434, 231);
this->pictureBox2->Name = L"pictureBox2";
this->pictureBox2->Size = System::Drawing::Size(314, 92);
this->pictureBox2->TabIndex = 12;
this->pictureBox2->TabStop = false;
//
// button3
//
this->button3->BackColor =

```

```

System::Drawing::SystemColors::HotTrack;
    this->button3->ForeColor =
System::Drawing::SystemColors::ButtonFace;
    this->button3->Location = System::Drawing::Point(623, 450);
    this->button3->Name = L"button3";
    this->button3->Size = System::Drawing::Size(125, 53);
    this->button3->TabIndex = 13;
    this->button3->Text = L"Завершити теорію";
    this->button3->UseVisualStyleBackColor = false;
    this->button3->Click += gcnew System::EventHandler(this,
&MyForm::button3_Click);
    //
    // button4
    //
    this->button4->BackColor =
System::Drawing::SystemColors::HotTrack;
    this->button4->ForeColor =
System::Drawing::SystemColors::ButtonFace;
    this->button4->Location = System::Drawing::Point(12, 450);
    this->button4->Name = L"button4";
    this->button4->Size = System::Drawing::Size(110, 53);
    this->button4->TabIndex = 14;
    this->button4->Text = L"Головне меню";
    this->button4->UseVisualStyleBackColor = false;
    this->button4->Click += gcnew System::EventHandler(this,
&MyForm::button4_Click);
    //
    // MyForm
    //
    this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
    this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
    this->ClientSize = System::Drawing::Size(760, 515);
    this->Controls->Add(this->button4);
    this->Controls->Add(this->button3);
    this->Controls->Add(this->pictureBox2);
    this->Controls->Add(this->button2);
    this->Controls->Add(this->button1);
    this->Controls->Add(this->label2);
    this->Controls->Add(this->pictureBox1);
    this->Controls->Add(this->textBox1);
    this->Controls->Add(this->label1);
    this->Name = L"MyForm";
    this->StartPosition =

```



```

System::Windows::Forms::FormStartPosition::CenterScreen;
        this->Text = L"Теорія";
        this->Load += gnew System::EventHandler(this,
&MyForm::MyForm_Load);
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox1))->EndInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox2))->EndInit();
        this->ResumeLayout(false);
        this->PerformLayout();

    }
#pragma endregion

```

Код зміни тексту на формі цикл «While» при натисканні на кнопку

```

private: System::Void button2_Click(System::Object^ sender,
System::EventArgs^ e) {
    label1->Text = "Цикл while";
    textBox1->Text = "Оператор циклу while або цикл while – цикл, що
повторює одну і ту ж дію, поки умова продовження циклу while залишається
істиною.";
    textBox1->Text += Environment::NewLine + "Умова продовження циклу
має бути істиною, як тільки умова стала хибною, виконується вихід з циклу. Також
як і в умовних операторах вибору, фігурні дужки можуть опускатися в тому
випадку, якщо тіло циклу – це один оператор. Але як правило в циклі виконується
декілька операторів, так як крім виконання корисної дії необхідно робити умову
циклу while хибною, інакше цикл буде нескінченним, а це, в свою чергу, приведе
до зависання програми.";
    textBox1->Text += Environment::NewLine + "Нижче наведено
синтаксис циклу while.";
    pictureBox2->Visible = false;
    pictureBox1->Visible = true;
}

```

Інформація, яка запускається при завантаженні форми.

```

private: System::Void MyForm_Load(System::Object^ sender, System::EventArgs^ e) {
    textBox1->Text = "Оператор циклу while або цикл while – цикл, що повторює
одну і ту ж дію, поки умова продовження циклу while залишається істиною.";
    textBox1->Text += Environment::NewLine + "Умова продовження циклу має
бути істиною, як тільки умова стала хибною, виконується вихід з циклу. Також як і
в умовних операторах вибору, фігурні дужки можуть опускатися в тому випадку,
якщо тіло циклу – це один оператор. Але як правило в циклі виконується декілька
операторів, так як крім виконання корисної дії необхідно робити умову циклу
while хибною, інакше цикл буде нескінченним, а це, в свою чергу, приведе до
зависання програми.";
    textBox1->Text += Environment::NewLine + "Нижче наведено синтаксис

```

```
циклу while";
    pictureBox2->Visible = false;
}
```

Код зміни тексту на формі цикл «Do while», при натисканні на кнопку

```
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
```

```
    label1->Text = "Цикл do while";
```

```
    pictureBox1->Visible = false;
```

textBox1->Text = "Цикл do while відрізняється від циклу while тим, що в do while спочатку виконується тіло циклу, а потім перевіряється умова продовження циклу. Через таку особливість do while називають циклом з постумовою. Таким чином, якщо умова do while завідомо хибна, то хоча б один раз блок операторів в тілі циклу do while виконається. В підсумку do while відрізняється від циклу while структурою.";

```
    textBox1->Text += Environment::NewLine + "На початку циклу do while пишеться зарезервоване слово do, після йдуть фігурні дужки, які можна опускати, у разі використання одного оператора в тілі циклу do while. Після закривання фігурної дужки, позначаючої кінець тіла циклу do while, стоїть умова циклу do while, після якої обов'язково потрібно ставити крапку з комою.";
```

```
    pictureBox2->Visible = true;
```

```
}
```

Завершення теорії при натисканні на кнопку.

```
private: System::Void button3_Click(System::Object^ sender, System::EventArgs^ e) {
```

```
    MessageBox::Show("Теорія пройдена!");
```

```
    MyForm1^ form = gcnew MyForm1();
```

```
    form->Show();
```

```
    this->Close();
```

```
}
```

Повернення в меню при натисканні на кнопку.

```
private: System::Void button4_Click(System::Object^ sender, System::EventArgs^ e) {
```

```
    MessageBox::Show("Повернення до головного меню!");
```

```
    this->Close();
```

```
}
```

```
};
```

```
}
```

Опис практичної частини

```
#pragma once
```

Підключення бібліотек.

```
namespace Project {
```

```
    using namespace System;
```

```
    using namespace System::ComponentModel;
```

```
    using namespace System::Collections;
```

```
    using namespace System::Windows::Forms;
```

```
    using namespace System::Data;
```

```

using namespace System::Drawing;

/// <summary>
/// Сводка для MyForm1
/// </summary>
public ref class MyForm1 : public System::Windows::Forms::Form
{
public:
    MyForm1(void)
    {
        InitializeComponent();
        //
        //TODO: добавьте код конструктора
        //
    }

protected:
    /// <summary>
    /// Освободить все используемые ресурсы.
    /// </summary>
    ~MyForm1()
    {
        if (components)
        {
            delete components;
        }
    }
}

```

Компоненты формы.

```

private: System::Windows::Forms::GroupBox^ groupBox1;
protected:
private: System::Windows::Forms::RadioButton^ radioButton3;
private: System::Windows::Forms::RadioButton^ radioButton1;
private: System::Windows::Forms::RadioButton^ radioButton4;
private: System::Windows::Forms::RadioButton^ radioButton2;
private: System::Windows::Forms::Label^ label1;
private: System::Windows::Forms::GroupBox^ groupBox2;
private: System::Windows::Forms::RadioButton^ radioButton5;
private: System::Windows::Forms::RadioButton^ radioButton6;
private: System::Windows::Forms::RadioButton^ radioButton7;
private: System::Windows::Forms::RadioButton^ radioButton8;
private: System::Windows::Forms::GroupBox^ groupBox3;
private: System::Windows::Forms::RadioButton^ radioButton9;
private: System::Windows::Forms::RadioButton^ radioButton10;
private: System::Windows::Forms::RadioButton^ radioButton11;
private: System::Windows::Forms::RadioButton^ radioButton12;
private: System::Windows::Forms::GroupBox^ groupBox4;
private: System::Windows::Forms::RadioButton^ radioButton13;
private: System::Windows::Forms::RadioButton^ radioButton14;
private: System::Windows::Forms::RadioButton^ radioButton15;
private: System::Windows::Forms::RadioButton^ radioButton16;
private: System::Windows::Forms::PictureBox^ pictureBox1;
private: System::Windows::Forms::PictureBox^ pictureBox2;

```

```

private: System::Windows::Forms::GroupBox^ groupBox5;
private: System::Windows::Forms::RadioButton^ radioButton17;
private: System::Windows::Forms::RadioButton^ radioButton18;
private: System::Windows::Forms::RadioButton^ radioButton19;
private: System::Windows::Forms::RadioButton^ radioButton20;
private: System::Windows::Forms::PictureBox^ pictureBox3;
private: System::Windows::Forms::Button^ button1;

```

```

private:
    /// <summary>
    /// Обязательная переменная конструктора.
    /// </summary>
    System::ComponentModel::Container ^components;

```

```

#pragma region Windows Form Designer generated code

```

```

    /// <summary>
    /// Требуемый метод для поддержки конструктора — не изменяйте
    /// содержимое этого метода с помощью редактора кода.
    /// </summary>
    void InitializeComponent(void)

```

Опис компонентів форми.

```

    {
        System::ComponentModel::ComponentResourceManager^ resources = (gcnew
System::ComponentModel::ComponentResourceManager(MyForm1::typeid));
        this->groupBox1 = (gcnew System::Windows::Forms::GroupBox());
        this->radioButton3 = (gcnew System::Windows::Forms::RadioButton());
        this->radioButton1 = (gcnew System::Windows::Forms::RadioButton());
        this->radioButton4 = (gcnew System::Windows::Forms::RadioButton());
        this->radioButton2 = (gcnew System::Windows::Forms::RadioButton());
        this->label1 = (gcnew System::Windows::Forms::Label());
        this->groupBox2 = (gcnew System::Windows::Forms::GroupBox());
        this->radioButton5 = (gcnew System::Windows::Forms::RadioButton());
        this->radioButton6 = (gcnew System::Windows::Forms::RadioButton());
        this->radioButton7 = (gcnew System::Windows::Forms::RadioButton());
        this->radioButton8 = (gcnew System::Windows::Forms::RadioButton());
        this->groupBox3 = (gcnew System::Windows::Forms::GroupBox());
        this->radioButton9 = (gcnew System::Windows::Forms::RadioButton());
        this->radioButton10 = (gcnew System::Windows::Forms::RadioButton());
        this->radioButton11 = (gcnew System::Windows::Forms::RadioButton());
        this->radioButton12 = (gcnew System::Windows::Forms::RadioButton());
        this->groupBox4 = (gcnew System::Windows::Forms::GroupBox());
        this->radioButton13 = (gcnew System::Windows::Forms::RadioButton());
        this->radioButton14 = (gcnew System::Windows::Forms::RadioButton());
        this->radioButton15 = (gcnew System::Windows::Forms::RadioButton());
        this->radioButton16 = (gcnew System::Windows::Forms::RadioButton());
        this->pictureBox1 = (gcnew System::Windows::Forms::PictureBox());
        this->pictureBox2 = (gcnew System::Windows::Forms::PictureBox());
        this->groupBox5 = (gcnew System::Windows::Forms::GroupBox());
        this->radioButton17 = (gcnew System::Windows::Forms::RadioButton());
        this->radioButton18 = (gcnew System::Windows::Forms::RadioButton());
        this->radioButton19 = (gcnew System::Windows::Forms::RadioButton());
        this->radioButton20 = (gcnew System::Windows::Forms::RadioButton());

```

```

        this->pictureBox3 = (gcnew System::Windows::Forms::PictureBox());
        this->button1 = (gcnew System::Windows::Forms::Button());
        this->groupBox1->SuspendLayout();
        this->groupBox2->SuspendLayout();
        this->groupBox3->SuspendLayout();
        this->groupBox4->SuspendLayout();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox1))->BeginInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox2))->BeginInit();
        this->groupBox5->SuspendLayout();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox3))->BeginInit();
        this->SuspendLayout();
        //
        // groupBox1
        //
        this->groupBox1->Controls->Add(this->radioButton3);
        this->groupBox1->Controls->Add(this->radioButton1);
        this->groupBox1->Controls->Add(this->radioButton4);
        this->groupBox1->Controls->Add(this->radioButton2);
        this->groupBox1->Location = System::Drawing::Point(12, 56);
        this->groupBox1->Name = L"groupBox1";
        this->groupBox1->Size = System::Drawing::Size(200, 129);
        this->groupBox1->TabIndex = 0;
        this->groupBox1->TabStop = false;
        this->groupBox1->Text = L"1) Цикл, який повторюється, поки умова
істинна.";
        //
        // radioButton3
        //
        this->radioButton3->AutoSize = true;
        this->radioButton3->Location = System::Drawing::Point(6, 80);
        this->radioButton3->Name = L"radioButton3";
        this->radioButton3->Size = System::Drawing::Size(52, 17);
        this->radioButton3->TabIndex = 4;
        this->radioButton3->TabStop = true;
        this->radioButton3->Text = L"с) For";
        this->radioButton3->UseVisualStyleBackColor = true;
        //
        // radioButton1
        //
        this->radioButton1->AutoSize = true;
        this->radioButton1->Location = System::Drawing::Point(6, 34);
        this->radioButton1->Name = L"radioButton1";
        this->radioButton1->Size = System::Drawing::Size(93, 17);
        this->radioButton1->TabIndex = 1;
        this->radioButton1->TabStop = true;
        this->radioButton1->Text = L"а) Цикл While";
        this->radioButton1->UseVisualStyleBackColor = true;
        //
        // radioButton4

```

```

//
this->radioButton4->AutoSize = true;
this->radioButton4->Location = System::Drawing::Point(6, 103);
this->radioButton4->Name = L"radioButton4";
this->radioButton4->Size = System::Drawing::Size(43, 17);
this->radioButton4->TabIndex = 3;
this->radioButton4->TabStop = true;
this->radioButton4->Text = L"d) If";
this->radioButton4->UseVisualStyleBackColor = true;
//
// radioButton2
//
this->radioButton2->AutoSize = true;
this->radioButton2->Location = System::Drawing::Point(6, 57);
this->radioButton2->Name = L"radioButton2";
this->radioButton2->Size = System::Drawing::Size(110, 17);
this->radioButton2->TabIndex = 2;
this->radioButton2->TabStop = true;
this->radioButton2->Text = L"b) Цикл Do While";
this->radioButton2->UseVisualStyleBackColor = true;
//
// label1
//
this->label1->AutoSize = true;
this->label1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif",
15.25F));
this->label1->Location = System::Drawing::Point(13, 9);
this->label1->Name = L"label1";
this->label1->Size = System::Drawing::Size(599, 25);
this->label1->TabIndex = 1;
this->label1->Text = L"Обрати необхідно лише одну відповідь у кожному
питанні!";
//
// groupBox2
//
this->groupBox2->Controls->Add(this->radioButton5);
this->groupBox2->Controls->Add(this->radioButton6);
this->groupBox2->Controls->Add(this->radioButton7);
this->groupBox2->Controls->Add(this->radioButton8);
this->groupBox2->Location = System::Drawing::Point(12, 200);
this->groupBox2->Name = L"groupBox2";
this->groupBox2->Size = System::Drawing::Size(200, 154);
this->groupBox2->TabIndex = 5;
this->groupBox2->TabStop = false;
this->groupBox2->Text = L"2) Цикл, у якого спочатку виконується тіло
циклу, а потім перевіряється умова.";
//
// radioButton5
//
this->radioButton5->AutoSize = true;
this->radioButton5->Location = System::Drawing::Point(6, 102);
this->radioButton5->Name = L"radioButton5";

```

```

this->radioButton5->Size = System::Drawing::Size(43, 17);
this->radioButton5->TabIndex = 4;
this->radioButton5->TabStop = true;
this->radioButton5->Text = L"c) If";
this->radioButton5->UseVisualStyleBackColor = true;
//
// radioButton6
//
this->radioButton6->AutoSize = true;
this->radioButton6->Location = System::Drawing::Point(6, 56);
this->radioButton6->Name = L"radioButton6";
this->radioButton6->Size = System::Drawing::Size(93, 17);
this->radioButton6->TabIndex = 1;
this->radioButton6->TabStop = true;
this->radioButton6->Text = L"а) Цикл While";
this->radioButton6->UseVisualStyleBackColor = true;
//
// radioButton7
//
this->radioButton7->AutoSize = true;
this->radioButton7->Location = System::Drawing::Point(6, 125);
this->radioButton7->Name = L"radioButton7";
this->radioButton7->Size = System::Drawing::Size(110, 17);
this->radioButton7->TabIndex = 3;
this->radioButton7->TabStop = true;
this->radioButton7->Text = L"d) Цикл Do While";
this->radioButton7->UseVisualStyleBackColor = true;
//
// radioButton8
//
this->radioButton8->AutoSize = true;
this->radioButton8->Location = System::Drawing::Point(6, 79);
this->radioButton8->Name = L"radioButton8";
this->radioButton8->Size = System::Drawing::Size(52, 17);
this->radioButton8->TabIndex = 2;
this->radioButton8->TabStop = true;
this->radioButton8->Text = L"b) For";
this->radioButton8->UseVisualStyleBackColor = true;
//
// groupBox3
//
this->groupBox3->Controls->Add(this->radioButton9);
this->groupBox3->Controls->Add(this->radioButton10);
this->groupBox3->Controls->Add(this->radioButton11);
this->groupBox3->Controls->Add(this->radioButton12);
this->groupBox3->Location = System::Drawing::Point(294, 200);
this->groupBox3->Name = L"groupBox3";
this->groupBox3->Size = System::Drawing::Size(200, 154);
this->groupBox3->TabIndex = 7;
this->groupBox3->TabStop = false;
this->groupBox3->Text = L"4) Який цикл потрібно написати замість

```

кряпок\?";

```

//
// radioButton9
//
this->radioButton9->AutoSize = true;
this->radioButton9->Location = System::Drawing::Point(6, 80);
this->radioButton9->Name = L"radioButton9";
this->radioButton9->Size = System::Drawing::Size(93, 17);
this->radioButton9->TabIndex = 4;
this->radioButton9->TabStop = true;
this->radioButton9->Text = L"с) Цикл While";
this->radioButton9->UseVisualStyleBackColor = true;
//
// radioButton10
//
this->radioButton10->AutoSize = true;
this->radioButton10->Location = System::Drawing::Point(6, 34);
this->radioButton10->Name = L"radioButton10";
this->radioButton10->Size = System::Drawing::Size(52, 17);
this->radioButton10->TabIndex = 1;
this->radioButton10->TabStop = true;
this->radioButton10->Text = L"а) For";
this->radioButton10->UseVisualStyleBackColor = true;
//
// radioButton11
//
this->radioButton11->AutoSize = true;
this->radioButton11->Location = System::Drawing::Point(6, 103);
this->radioButton11->Name = L"radioButton11";
this->radioButton11->Size = System::Drawing::Size(43, 17);
this->radioButton11->TabIndex = 3;
this->radioButton11->TabStop = true;
this->radioButton11->Text = L"д) If";
this->radioButton11->UseVisualStyleBackColor = true;
//
// radioButton12
//
this->radioButton12->AutoSize = true;
this->radioButton12->Location = System::Drawing::Point(6, 57);
this->radioButton12->Name = L"radioButton12";
this->radioButton12->Size = System::Drawing::Size(110, 17);
this->radioButton12->TabIndex = 2;
this->radioButton12->TabStop = true;
this->radioButton12->Text = L"б) Цикл Do While";
this->radioButton12->UseVisualStyleBackColor = true;
//
// groupBox4
//
this->groupBox4->Controls->Add(this->radioButton13);
this->groupBox4->Controls->Add(this->radioButton14);
this->groupBox4->Controls->Add(this->radioButton15);
this->groupBox4->Controls->Add(this->radioButton16);
this->groupBox4->Location = System::Drawing::Point(288, 56);

```



```

this->groupBox4->Name = L"groupBox4";
this->groupBox4->Size = System::Drawing::Size(200, 129);
this->groupBox4->TabIndex = 6;
this->groupBox4->TabStop = false;
this->groupBox4->Text = L"3) Який цикл потрібно написати замість
кряпок\?";

//
// radioButton13
//
this->radioButton13->AutoSize = true;
this->radioButton13->Location = System::Drawing::Point(6, 80);
this->radioButton13->Name = L"radioButton13";
this->radioButton13->Size = System::Drawing::Size(93, 17);
this->radioButton13->TabIndex = 4;
this->radioButton13->TabStop = true;
this->radioButton13->Text = L"с) Цикл While";
this->radioButton13->UseVisualStyleBackColor = true;
//
// radioButton14
//
this->radioButton14->AutoSize = true;
this->radioButton14->Location = System::Drawing::Point(6, 34);
this->radioButton14->Name = L"radioButton14";
this->radioButton14->Size = System::Drawing::Size(110, 17);
this->radioButton14->TabIndex = 1;
this->radioButton14->TabStop = true;
this->radioButton14->Text = L"а) Цикл Do While";
this->radioButton14->UseVisualStyleBackColor = true;
//
// radioButton15
//
this->radioButton15->AutoSize = true;
this->radioButton15->Location = System::Drawing::Point(6, 103);
this->radioButton15->Name = L"radioButton15";
this->radioButton15->Size = System::Drawing::Size(52, 17);
this->radioButton15->TabIndex = 3;
this->radioButton15->TabStop = true;
this->radioButton15->Text = L"д) For";
this->radioButton15->UseVisualStyleBackColor = true;
//
// radioButton16
//
this->radioButton16->AutoSize = true;
this->radioButton16->Location = System::Drawing::Point(6, 57);
this->radioButton16->Name = L"radioButton16";
this->radioButton16->Size = System::Drawing::Size(43, 17);
this->radioButton16->TabIndex = 2;
this->radioButton16->TabStop = true;
this->radioButton16->Text = L"б) If";
this->radioButton16->UseVisualStyleBackColor = true;
//
// pictureBox1

```

```

//
this->pictureBox1->Image =
(cli::safe_cast<System::Drawing::Image^(resources->GetObject(L"pictureBox1.Image")));
this->pictureBox1->Location = System::Drawing::Point(503, 65);
this->pictureBox1->Name = L"pictureBox1";
this->pictureBox1->Size = System::Drawing::Size(200, 111);
this->pictureBox1->TabIndex = 8;
this->pictureBox1->TabStop = false;
//
// pictureBox2
//
this->pictureBox2->Image =
(cli::safe_cast<System::Drawing::Image^(resources->GetObject(L"pictureBox2.Image")));
this->pictureBox2->Location = System::Drawing::Point(503, 216);
this->pictureBox2->Name = L"pictureBox2";
this->pictureBox2->Size = System::Drawing::Size(174, 81);
this->pictureBox2->TabIndex = 9;
this->pictureBox2->TabStop = false;
//
// groupBox5
//
this->groupBox5->Controls->Add(this->radioButton17);
this->groupBox5->Controls->Add(this->radioButton18);
this->groupBox5->Controls->Add(this->radioButton19);
this->groupBox5->Controls->Add(this->radioButton20);
this->groupBox5->Location = System::Drawing::Point(12, 373);
this->groupBox5->Name = L"groupBox5";
this->groupBox5->Size = System::Drawing::Size(200, 154);
this->groupBox5->TabIndex = 8;
this->groupBox5->TabStop = false;
this->groupBox5->Text = L"5) Який цикл потрібно написати замість
крапок\?";
//
// radioButton17
//
this->radioButton17->AutoSize = true;
this->radioButton17->Location = System::Drawing::Point(6, 80);
this->radioButton17->Name = L"radioButton17";
this->radioButton17->Size = System::Drawing::Size(93, 17);
this->radioButton17->TabIndex = 4;
this->radioButton17->TabStop = true;
this->radioButton17->Text = L"с) Цикл While";
this->radioButton17->UseVisualStyleBackColor = true;
//
// radioButton18
//
this->radioButton18->AutoSize = true;
this->radioButton18->Location = System::Drawing::Point(6, 103);
this->radioButton18->Name = L"radioButton18";
this->radioButton18->Size = System::Drawing::Size(52, 17);
this->radioButton18->TabIndex = 1;
this->radioButton18->TabStop = true;

```

```

this->radioButton18->Text = L"d) For";
this->radioButton18->UseVisualStyleBackColor = true;
//
// radioButton19
//
this->radioButton19->AutoSize = true;
this->radioButton19->Location = System::Drawing::Point(6, 57);
this->radioButton19->Name = L"radioButton19";
this->radioButton19->Size = System::Drawing::Size(43, 17);
this->radioButton19->TabIndex = 3;
this->radioButton19->TabStop = true;
this->radioButton19->Text = L"b) If";
this->radioButton19->UseVisualStyleBackColor = true;
//
// radioButton20
//
this->radioButton20->AutoSize = true;
this->radioButton20->Location = System::Drawing::Point(6, 32);
this->radioButton20->Name = L"radioButton20";
this->radioButton20->Size = System::Drawing::Size(110, 17);
this->radioButton20->TabIndex = 2;
this->radioButton20->TabStop = true;
this->radioButton20->Text = L"a) ЦИКЛ Do While";
this->radioButton20->UseVisualStyleBackColor = true;
//
// pictureBox3
//
this->pictureBox3->Image =
(cli::safe_cast<System::Drawing::Image^(resources->GetObject(L"pictureBox3.Image")));
this->pictureBox3->Location = System::Drawing::Point(237, 373);
this->pictureBox3->Name = L"pictureBox3";
this->pictureBox3->Size = System::Drawing::Size(477, 120);
this->pictureBox3->TabIndex = 10;
this->pictureBox3->TabStop = false;
//
// button1
//
this->button1->BackColor = System::Drawing::SystemColors::HotTrack;
this->button1->ForeColor = System::Drawing::SystemColors::ButtonFace;
this->button1->Location = System::Drawing::Point(300, 499);
this->button1->Name = L"button1";
this->button1->Size = System::Drawing::Size(209, 53);
this->button1->TabIndex = 11;
this->button1->Text = L"Завершить";
this->button1->UseVisualStyleBackColor = false;
this->button1->Click += gcnew System::EventHandler(this,
&MyForm1::button1_Click);
//
// MyForm1
//
this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

```

```

        this->ClientSize = System::Drawing::Size(758, 564);
        this->Controls->Add(this->button1);
        this->Controls->Add(this->pictureBox3);
        this->Controls->Add(this->groupBox5);
        this->Controls->Add(this->pictureBox2);
        this->Controls->Add(this->pictureBox1);
        this->Controls->Add(this->groupBox3);
        this->Controls->Add(this->groupBox4);
        this->Controls->Add(this->groupBox2);
        this->Controls->Add(this->label1);
        this->Controls->Add(this->groupBox1);
        this->Name = L"MyForm1";
        this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterScreen;
        this->Text = L"Тест";
        this->groupBox1->ResumeLayout(false);
        this->groupBox1->PerformLayout();
        this->groupBox2->ResumeLayout(false);
        this->groupBox2->PerformLayout();
        this->groupBox3->ResumeLayout(false);
        this->groupBox3->PerformLayout();
        this->groupBox4->ResumeLayout(false);
        this->groupBox4->PerformLayout();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox1))->EndInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox2))->EndInit();
        this->groupBox5->ResumeLayout(false);
        this->groupBox5->PerformLayout();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox3))->EndInit();
        this->ResumeLayout(false);
        this->PerformLayout();
    }
#pragma endregion

```

Умови для перевірки тесту.

```

private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
    int s = 0;
    if (radioButton1->Checked || radioButton7->Checked || radioButton13->Checked ||
radioButton12->Checked || radioButton20->Checked) { відповіли на 1 питання
        s = s + 1;
        if (radioButton1->Checked && radioButton7->Checked || radioButton1-
>Checked && radioButton13->Checked || radioButton1->Checked && radioButton12->Checked ||
radioButton1->Checked && radioButton20->Checked || radioButton7->Checked && radioButton12-
>Checked || radioButton7->Checked && radioButton13->Checked || radioButton7->Checked &&
radioButton20->Checked || radioButton20->Checked && radioButton12->Checked || radioButton20-
>Checked && radioButton13->Checked) { відповіли на 2 питання
            s = s + 1;
            if (radioButton1->Checked && radioButton7->Checked &&

```

```
radioButton12->Checked || radioButton1->Checked && radioButton12->Checked && radioButton20->Checked || radioButton7->Checked && radioButton12->Checked && radioButton20->Checked || radioButton7->Checked && radioButton13->Checked && radioButton20->Checked || radioButton7->Checked && radioButton12->Checked && radioButton13->Checked || radioButton1->Checked && radioButton7->Checked && radioButton20->Checked || radioButton1->Checked && radioButton7->Checked && radioButton13->Checked || radioButton1->Checked && radioButton12->Checked && radioButton13->Checked || radioButton1->Checked && radioButton12->Checked && radioButton13->Checked || radioButton12->Checked && radioButton13->Checked && radioButton20->Checked) {
```

вiдповiли на 3 питання

```
    s = s + 1;
    if (radioButton1->Checked && radioButton7->Checked && radioButton12->Checked && radioButton13->Checked || radioButton1->Checked && radioButton7->Checked && radioButton12->Checked && radioButton20->Checked || radioButton1->Checked && radioButton7->Checked && radioButton13->Checked && radioButton20->Checked || radioButton7->Checked && radioButton12->Checked && radioButton13->Checked && radioButton20->Checked || radioButton1->Checked && radioButton12->Checked && radioButton13->Checked && radioButton20->Checked || radioButton1->Checked && radioButton12->Checked && radioButton13->Checked && radioButton20->Checked) {
```

вiдповiли на 4 питання

```
    s = s + 1;
    if (radioButton1->Checked && radioButton7->Checked && radioButton12->Checked && radioButton13->Checked && radioButton20->Checked) {
```

вiдповiли на 5 питань

```
    s = s + 1;
    Перевiрка вiдповiдей и вивiд потрiбного тексту.
    if (s == 0) {
        MessageBox::Show("Ви вiдповiли вiрно на нуль питань!");
    }
    else if (s == 1) {
        MessageBox::Show("Ви вiдповiли вiрно на одне питання!");
    }
    else if (s == 2) {
        MessageBox::Show("Ви вiдповiли вiрно на два питання!");
    }
    else if (s == 3) {
        MessageBox::Show("Ви вiдповiли вiрно на три питання!");
    }
    else if (s == 4) {
        MessageBox::Show("Ви вiдповiли вiрно на чотири питання!");
    }
    else if (s == 5) {
        MessageBox::Show("Ви вiдповiли вiрно на п'ять запитань!");
    }
    this->Close();
}
if (s == 4) {
```

```

        MessageBox::Show("Ви відповіли вірно на
чотири питання!");
    }
    }
    if (s == 3) {
        MessageBox::Show("Ви відповіли вірно на три
питання!");
    }
    }
    if (s == 2) {
        MessageBox::Show("Ви відповіли вірно на два питання!");
    }
    }
    if (s == 1) {
        MessageBox::Show("Ви відповіли вірно на одне питання!");
    }
    }
    if (s == 0) {
        MessageBox::Show("Ви відповіли вірно на нуль питань!");
    }
    this->Close();
}
};
}

```