**національний університет БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ україни**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**ЛАБОРАТОРНА РОБОТА №3**

Виконав:

студент групи ІПЗ-2006б

Купчик В. Г.

Київ – 2023

**Завдання до лабораторної роботи**

**Мета роботи:** навчитися програмувати прості класи об'єктів на прикладах геометричних фігур.

**Завдання до лабораторної роботи:**

1. Створити новий проект типу «Додаток Windows Form» - графічну форму для відображення об'єктів створюваного класу.

2. У проекті створити окремий файл, і в ньому запрограмувати клас, що описує геометричну

фігуру відповідно до варіанта завдання.

3. В клас ввести статичне поле для підрахунку кількості примірників об'єктів цього класу.

4. В клас ввести поля, що задають геометричні розміри фігури, їх колір і інші параметри.

5. Запрограмувати конструктор з параметрами для об'єктів цього класу.

6. На формі розмістити кнопку і запрограмувати її на створення чергового екземпляру об'єкта цього класу. При кожному натисканні кнопки повинен створюватися і відображатися на екрані новий екземпляр геометричної фігури. Розмір, розташування та інші параметри фігури повинні обчислюватися відповідно до підраховують автоматично порядковим номером примірника.

7. Перевірити працездатність створеної програми.

Варіант - 5 (15)

5. Геометрічна фігура «чотирикутний зірка»:

a) його екземпляр містить розміри a і b.

b) його конструктор без параметра створює екземпляр зі значенням 0, а конструктор з параметрами створює екземпляр з відповідним значенням a і b.

Изображение выглядит как линия, зарисовка

Автоматически созданное описание

Код:

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Windows.Forms;

namespace WindowsFormsApp2

{

public class Star

{

public static int StarCount = 0;

private Point center;

private int radius;

private Form1 form;

private double angle = 0;

private double spikeLength;

private int spikeCount;

private int lastRadius = 10;

public double Angle { get { return angle; } }

public Star(Point c, double r, int spikes, Form1 form1)

{

++StarCount;

center = c;

radius = form1.scale \* (int)r;

form = form1;

spikeLength = radius \* 0.2;

spikeCount = spikes;

}

public void Draw(Pen pen, Graphics graph)

{

if (graph != null)

{

Point[] starPoints = new Point[spikeCount \* 2];

for (int i = 0; i < spikeCount \* 2; i++)

{

double angleRad = Math.PI \* 2 \* i / (spikeCount \* 2) + angle;

double length = i % 2 == 0 ? radius : spikeLength;

double x = center.X + length \* Math.Cos(angleRad);

double y = center.Y + length \* Math.Sin(angleRad);

starPoints[i] = new Point((int)x, (int)y);

}

graph.DrawPolygon(pen, starPoints);

}

}

public void Show() { Draw(form.pen1, form.graph); }

public void Hide() { Draw(form.pen2, form.graph); }

public void IncreaseSize(Star lastStar)

{

radius = (int)(lastStar.radius \* 1.5); // Збільшуємо розмір зірки

spikeLength = radius \* 0.2;

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Windows.Forms.AxHost;

using WindowsFormsApp2;

namespace WindowsFormsApp2

{

public partial class Form1 : Form

{

public Graphics graph;

public Pen pen1, pen2, pen3;

private Star lastStar; // Змінна для зберігання останньої створеної зірки

public int scale = 5;

public Form1()

{

InitializeComponent();

graph = CreateGraphics();

pen1 = new Pen(Color.Black);

pen2 = new Pen(Color.White);

pen3 = new Pen(Color.Red);

}

private void button1\_Click(object sender, EventArgs e)

{

Point p1 = new Point(scale \* (Star.StarCount \* 10 + 10), scale \* 30);

Star star = new Star(p1, 10, 4, this);

if (lastStar != null)

{

star.IncreaseSize(lastStar); // Збільшення розміру з урахуванням останньої зірки

}

star.Show();

lastStar = star; // Зберігаємо останню створену зірку

}

}

}

Результат виконання:

Изображение выглядит как линия

Автоматически созданное описание

Висновок

Ця лабораторна робота допомогла розібратися з основами створення графічних об'єктів в середовищі Windows Forms, а також з роботою з класами та методами для малювання та збільшення їх розміру.

Контрольні питання

1. Що таке наслідування?

Наслідування - це механізм об'єктно-орієнтованого програмування, в якому новий клас (підклас або нащадок) може успадковувати властивості і методи іншого класу (базового класу або батька).

1. Що таке інкапсуляція?

Інкапсуляція - це принцип об'єктно-орієнтованого програмування, який полягає в об'єднанні даних (полів чи змінних) та методів (функцій) для їх операцій в одному об'єкті або класі, і прихованні деталей реалізації від зовнішнього світу.

1. Що таке поліморфізм?

Поліморфізм - це концепція об'єктно-орієнтованого програмування, яка дозволяє об'єктам різних класів реагувати на однаковий запит або метод специфічним для свого класу способом. Це дозволяє використовувати загальні інтерфейси для роботи з об'єктами різних типів.

1. Що таке абстрактний клас?

Абстрактний клас - це клас у програмуванні, який не може бути інстанційованим (створеним як об'єкт), і він містить один або декілька абстрактних методів, які мають бути реалізовані в підкласах. Використовується для створення загальних інтерфейсів і обов'язкових методів для підкласів.

1. Розкрийте призначення DLL-бібліотеки.

DLL (Dynamic Link Library) - це файл бібліотеки у середовищі операційної системи Windows, який містить функції і ресурси, доступні для інших програм. Вони дозволяють спільно використовувати функціональність між різними програмами та зменшують дублювання коду.

1. Наведіть переваги застосування DLL-бібліотек.

Переваги використання DLL-бібліотек включають:

Можливість спільного використання функціональності між декількома програмами.

Зменшення обсягу коду в окремих програмах.

Можливість оновлення і виправлення бібліотек без зміни основного програмного коду.

Зменшення використання системних ресурсів завдяки перевикористанню коду.

1. Опишіть механізм створення модуля DLL-бібліотеки.

Для створення DLL-бібліотеки в середовищі Windows, ви виконуєте наступні кроки:

Створіть новий проект у вашому обраному середовищі розробки (наприклад, Visual Studio).

Виберіть тип проекту, який відповідає бібліотеці DLL.

Розробіть функції або класи, які мають бути доступні з інших програм.

Скомпілюйте проект у форматі DLL.

Ваша бібліотека DLL готова до використання і може бути викликана іншими програмами, які імпортують її функціональність.