

Міністерство освіти і науки, молоді та спорту України
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«НАЦІОНАЛЬНИЙ ГІРНИЧИЙ УНІВЕРСИТЕТ»



Звіт
З лабораторної роботи №4
З дисципліни
«Об'єктно-орієнтоване програмування»
Варіант № 5

Виконав:
студент гр. :122-19-2
Вареник М. О

Перевірили:
доцент каф. ПЗКС
Приходченко С.Д

асистент каф. ПЗКС
Шевцова О.С

Дніпро

2020

Тема роботи: вивчення поліморфізму в класах C#.

Цілі роботи:

- навчитися створювати віртуальні методи;
- навчитися перевантажувати арифметичні операції та операції порівняння.

Основне завдання:

Створити класову модель, яка включає в себе абстрактний клас CGraphicsObject, його спадкоємця - базовий клас CPoint, и класа-спадкоємця от CPoint, що являє собою графічний об'єкт згідно з заданим варіантом. Всі класи, за виключенням абстрактного, повинні мати конструктори, деструктори, відповідні set- и get-методи, а також метод *Show*, який може мати різну сигнатуру для різних класів и повинен виводити на екран інформацію у вигляді тексту про об'єкт та його базові класи. Абстрактний клас повинен мати декларацію полей та методів.

Обов'язково перевантажувати вказану в варіанті операцію всередині класу.

5	Коло	<u>Еліпс</u>	<u>Операція "+", яка додає периметри графічних об'єктів</u>
---	------	--------------	---

Код програми:

CGraphicsObject.cs

```
using System.Drawing;
using System.Windows.Forms;

namespace homework_4
{
    public abstract class CGraphicsObject
    {
        protected Point point;

        public abstract double Perimeter();

        public virtual string GetClassName()
        {
            return "CGraphicsObject";
        }

        public virtual void Show(Label label)
        {
        }

        public virtual void Draw(Graphics graphics)
        {
        }

        public static double operator +(CGraphicsObject graphicsObject1, CGraphicsObject
graphicsObject2)
        {
            return graphicsObject1.Perimeter() + graphicsObject2.Perimeter();
        }
    }
}
```

```
}
```

Circle.cs

```
using System.Windows.Forms;
using static System.Math;
using System.Drawing;

namespace homework_4
{
    public class Circle : CGraphicsObject
    {
        private double radius;

        public Circle(double radius, int x, int y)
        {
            this.radius = radius;
            point = new Point(x, y);
        }

        ~Circle()
        {
        }

        public double Radius // радиус окружности
        {
            get { return radius; }

            set { radius = value; }
        }

        public override string GetClassName()
        {
            return "Circle";
        }

        public override double Perimeter()
        {
            return 2 * PI * Radius;
        }

        public override void Show(Label label)
        {
            label.Text = GetClassName() + ": \nRadius=" + Radius + "\nPerimeter=" +
Perimeter() + "\n";
        }

        public override void Draw(Graphics graphics)
        {
            graphics.DrawEllipse(new Pen(Color.Black), new Rectangle(point.X, point.Y,
(int)Radius, (int)Radius));
        }
    }
}
```

Ellipse.cs

```
using System.Drawing;
using System.Windows.Forms;
using static System.Math;
```

```

namespace homework_4
{
    public class Ellipse : CGraphicsObject
    {
        private double a; //первая полуось эллипса

        private double b; //вторая полуось эллипса

        public Ellipse(double a, double b, int x, int y)
        {
            this.a = a;

            this.b = b;

            point = new Point(x, y);
        }

        ~Ellipse()
        {
        }

        public double A //первая полуось эллипса
        {
            get { return a; }

            set { a = value; }
        }

        public override void Show(Label label)
        {
            label.Text = GetClassName() + ": \nA=" + A + "\nB=" + B + "\nPerimeter=" +
Perimeter() + "\n";
        }

        public override void Draw(Graphics graphics)
        {
            graphics.DrawEllipse(new Pen(Color.Black), new Rectangle(point.X, point.Y,
(int)A, (int)B));
        }

        public double B //вторая полуось эллипса
        {
            get { return b; }

            set { b = value; }
        }

        public override string GetClassName()
        {
            return "Ellipse";
        }

        public override double Perimeter()
        {
            return PI * Sqrt(2 * (Pow(A, 2) + Pow(B, 2)));
        }
    }
}

```

Form1.cs

```
using System.Windows.Forms;
```

```

namespace homework_4
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

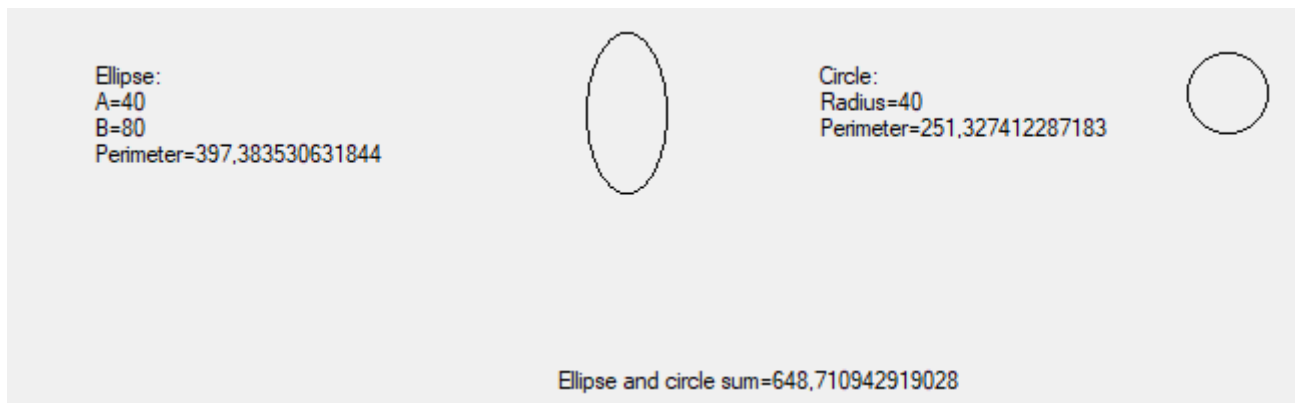
        private void pictureBox1_Paint(object sender, PaintEventArgs e)
        {
            Ellipse ellipse = new Ellipse(40, 80, 400, 20);
            ellipse.Draw(e.Graphics);
            ellipse.Show(label1);

            Circle circle = new Circle(40, 700, 30);
            circle.Draw(e.Graphics);
            circle.Show(label2);

            label3.Text = "Ellipse and circle sum=" + (ellipse + circle);
        }
    }
}

```

Результат:



Висновки: навчився створювати віртуальні методи. Навчився перевантажувати арифметичні операції та операції порівняння.