Министерство науки и ВЫСШЕГО образования   
Российской Федерации

Федеральное автономное образовательное учреждение высшего образования  
«Санкт-Петербургский государственный политехнический   
университет Петра Великого»  
**Институт среднего профессионального образования**

**Лабораторная работа № 5  
Тема:** «Разработка спецификаций структурных единиц класса. Организация иерархии наследования.»

МДК.01.01 Разработка программных модулей

**Вариант 9**

Выполнил: студент группы 32919/8  
Крупина Е.А.

Проверил: преподаватель   
Прокофьев А.А.

Санкт-Петербург  
2024

**Лабораторная работа№5**

**Тема:** Разработка спецификаций структурных единиц класса. Организация

иерархии наследования.

**Цель работы:** получить практические навыки разработки иерархии классов,

методов, свойств классов.

**Задание:**

Создать базовый класс Point (точка), содержащий поля: x, y (координаты точки). Создать производный класс Circle (круг), содержащий дополнительное поле: радиус, а от класса Circle производный класс Ellipse (эллипс), содержащий ещё одно дополнительное поле: второй радиус. Описать методы для вывода на экран сведений об объекте, получения и установки значений полей, методы вычисления площади фигур.

**Диаграмма классов:**



Рис.1 – Диаграмма классов

**Код программы:**

1. Program.cs

using System;

using System.Drawing;

namespace Lab5

{

class Program

{

static void Main()

{

//Проверка функционала конструкторов и методов для вывода

Point point = new Point(1, 2);

point.Print();

//Проверка функционала свойств

point.X = 3;

point.Y = 4; Console.WriteLine($"Координата точки X: {point.X}");

Console.WriteLine();

Console.WriteLine();

Circle circle = new Circle(1, 2, 3);

circle.Print(); Console.WriteLine();

Circle ellipse = new Ellipse(1, 2, 3, 2);

ellipse.Print();

//Проверка функционала методов для вычисления площади

Console.WriteLine();

Console.WriteLine($"Площадь окружности: {circle.Square()}");

Console.WriteLine($"Площадь эллипса: {ellipse.Square()}");

Console.WriteLine();

//Проверка функционала базовых конструкторов

Point point2 = new Point();

point2.Print();

Console.WriteLine();

Ellipse ellipse2 = new Ellipse(); ellipse2.Print();

Console.WriteLine(); Circle circle2 = new Circle();

circle2.Print(); Console.WriteLine();

}

}

}

1. Point.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab5

{

class Point //Базовый класс Точка

{

protected double x { get; set; }

protected double y { get; set; }

public double X

{

get { return x; }

set { x = value; }

}

public double Y

{

get { return y; }

set { y = value; }

}

public Point()

{ //Базовый конструктор

this.X = 0; this.Y = 0;

}

public Point(double x, double y)

{ //Конструктор

this.X = x;

this.Y = y;

}

public virtual void Print()

{ // Метод вывода на экран

Console.Write($"Координаты точки: ({X}, {Y})\n");

}

}

}

1. Circle.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab5

{

class Circle : Point

{ //Порожденный класс Окружность

protected double radius { get; set; }

public double Radius

{

get { return radius; }

set { radius = value; }

}

public Circle()

{

this.Radius = 0;

}

public Circle(double x, double y, double radius) : base(x, y)

{

this.Radius = radius;

}

public override void Print()

{

Console.Write($"Окружность:\n");

Console.Write($"Координаты центра: ({X}, {Y})\n"); Console.Write($"Радиус: {Radius}\n");

}

public virtual double Square()

{

double square = 3.14 \* Math.Pow(Radius, 2); return square;

}

}

}

1. Ellipse.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab5

{

class Ellipse : Circle

{ //Порожденный класс Эллипс

protected double secondRadius;

public double SecondRadius

{

get { return secondRadius; }

set { secondRadius = value; }

}

public Ellipse()

{

this.secondRadius = 0;

}

public Ellipse(double x, double y, double radius, double secondRadius) : base(x, y, radius)

{

this.secondRadius = secondRadius;

}

public override void Print()

{

Console.Write($"Эллипс:\n"); Console.Write($"Координаты центра: ({X}, {Y})\n");

Console.Write($"Первая полуось: {Radius}\n"); Console.Write($"Вторая полуось: {SecondRadius}\n");

}

public override double Square()

{

double square = 3.14 \* radius \* secondRadius; return square;

}

}

}

**Результат испытаний программы:**

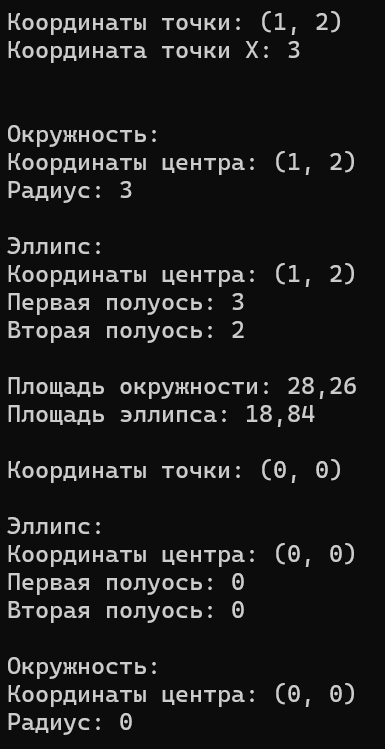


Рис.2 – Результат испытаний программы