**17 Сборки, библиотеки, атрибуты, директивы**

Задание 1. Создать библиотеку с двумя классами: Треугольник (методы ввода сторон, проверки на существование, вычисления периметра, вычисления площади, определения вида треугольника (разносторонний, равнобедренный, равносторонний)); Прямоугольник (методы ввода сторон, вычисления периметра, вычисления площади).

Листинг программы:

Rectange.cs:

public class Rectangle

{public double Width { get; private set; } public double Height {get; private set; }

public Rectangle(double width, double height)

{ Width = width; Height = height;}

public double GetPerimeter(){return 2 \* (Width + Height);}

public double GetArea(){return Width \* Height;}}

Triangle.cs:

public class Triangle

{public double SideA { get; private set; }public double SideB {get; private set; }

public double SideC { get; private set; }

public Triangle(double sideA, double sideB, double sideC)

{ SideA = sideA; SideB = sideB; SideC = sideC;}

public bool IsValid()

{return SideA+ SideB>SideC&&SideA+SideC>SideB&&SideB+SideC>SideA;}

public double GetPerimeter(){return SideA + SideB + SideC;}

public double GetArea(){double s = GetPerimeter() / 2;

return Math.Sqrt(s \* (s - SideA) \* (s - SideB) \* (s - SideC));}

public string GetTriangleType(){if (!IsValid()){

return "Треугольник не существует";}

else if (SideA == SideB && SideB == SideC){

return "Равносторонний треугольник";}

else if (SideA == SideB || SideA == SideC || SideB == SideC)

{return "Равнобедренный треугольник";}

else{return "Разносторонний треугольник";}}}

Program.cs:

class Program

{static void Main(string[] args){Console.WriteLine("Введите стороны треугольника:"); double sideA = double.Parse(Console.ReadLine());

double sideB = double.Parse(Console.ReadLine());

double sideC = double.Parse(Console.ReadLine());

Triangle triangle = new Triangle(sideA, sideB, sideC);

Console.WriteLine($"Периметр треугольника: {triangle.GetPerimeter()}");

Console.WriteLine($"Площадь треугольника: {triangle.GetArea()}");

Console.WriteLine($"Треугольник {triangle.GetTriangleType()}");

Console.WriteLine();Console.WriteLine("Введите стороны прямоугольника:");

double Width = double.Parse(Console.ReadLine());

double Height = double.Parse(Console.ReadLine());

Rectangle rectangle = new Rectangle(Width, Height);

Console.WriteLine($"Периметр прямоугольника: {rectangle.GetPerimeter()}");

Console.WriteLine($"Площадь прямоугольника: {rectangle.GetArea()}");}}}

Таблица 17.1 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 3, 4, 5; 5,10; | Периметр треугольника: 12  Площадь треугольника: 6  Разносторонний треугольник  Периметр прямоугольника: 30  Площадь прямоугольника: 50 |

Источник: собственная разработка

Анализ результатов:

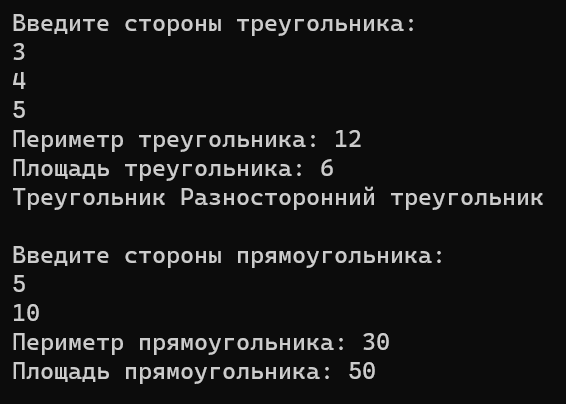


Рисунок 17.1 – Результат работы программы

Источник: собственная разработка

Задание 2. Создать свою библиотеку на индивидуальную тему «Погода» и продемонстрировать ее функциональность.

Листинг программы:

FlowersLibrary.cs:

namespace FlowersLibrary{public enum FlowerType{

Холодостойкие, Теплолюбивые, Теплостойкие}

public class Flower{public string Name{get;set;}

public int Temperature{get;set; }

public Flower(string name, int temperature){

Name = name; Temperature = temperature;}

public FlowerType GetFlowerType(){

if (Temperature >= 0 && Temperature <= 10){

return FlowerType.Холодостойкие;}

else if (Temperature > 10 && Temperature <= 27)

{return FlowerType.Теплолюбивые;}

else if (Temperature > 27) {return FlowerType.Теплостойкие;}

else{throw new ArgumentException("Некорректное значение температуры"); }}}}

Program.cs:

Console.Write("Введите название цветка:");

string name = Console.ReadLine();

Console.Write("Введите температуру, при которой нормально растет цветок:"); int temperature = int.Parse(Console.ReadLine());

Flower flower = new Flower(name, temperature);

FlowerType flowerType = flower.GetFlowerType();

Console.WriteLine($"Цветок {flower.Name} относится к типу {flowerType}");

Таблица 17.2 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Розы; 22 | Цветок Розы относится к типу Теплолюбивые |

Источник: собственная разработка

Анализ результатов:

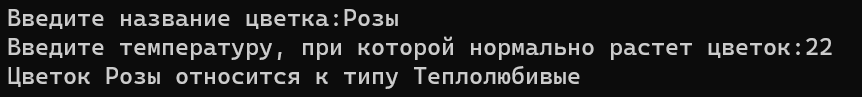


Рисунок 17.2 – Результат работы программы

Источник: собственная разработка