**17 СБОРКИ, БИБЛИОТЕКИ, АТРИБУТЫ, ДИРЕКТИВЫ**

Задание 1. 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:

class Program

{

static void Main(string[] args)

{

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

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

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

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

Triangle triag = new Triangle();

Console.WriteLine("Периметр треугльника равен " + triag.TriPerimetr(a, b, c));

Console.WriteLine(triag.TriangleForm(a, b, c));

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

int a1 = int.Parse(Console.ReadLine());

int b1 = int.Parse(Console.ReadLine());

Rectangle rec = new Rectangle();

Console.WriteLine(rec.RecPerimetr(a1, b1));

Console.WriteLine(rec.RecSquare(a1, b1));

}

}

}

public class Triangle

{

public double SideA { get; set; }

public double SideB { get; set; }

public double SideC { get; set; }

public double TriPerimetr(double SideA, double SideB, double SideC)

{

return SideA + SideB + SideC;

}

public string TriangleForm(double one, double two, double three)

{

if (one <= 0 || two <= 0 || three <= 0 || one + two <= three || one + three <= two || two + three <= one)

{

return "Треугольника с такими сторонами не существует!!!";

}

if (one == two && one == three && two == three)

{

double s = (Math.Pow(one, 2) \* Math.Sqrt(3)) / 4;

return "Треугольник равносторонний!\nПлощадь треугольника равна " + Convert.ToString(s);

}

else if ((one == two && one != three) || (two == three && two != one) || (one == three && one != two))

{

double s = (three \* Math.Sqrt(4 \* Math.Pow(one, 2) - Math.Pow(two, 2))) / 4;

return "Треугольник равнобедренный!\nПлощадь треугольника равна " + Convert.ToString(s);

}

else

{

double p = TriPerimetr(one, two, three) / 2;

double s = Math.Sqrt(p \* (p - one) \* (one - two) \* (p - three));

return "Треугольник разносторонний!\nПлощадь треугольника равна " + Convert.ToString(s);

}

}

}

}

public class Rectangle

{

public double SideA { get; set; }

public double SideB { get; set; }

public string RecPerimetr(double SideA, double SideB)

{

double p = (SideA + SideB) \* 2;

return "Периметр прямоугольника равен " + p;

}

public string RecSquare(double SideA, double SideB)

{

double s = SideA \* SideB;

return "Площадь прямоугольника равна " + s;

}

}

}

Таблица 17.1 – Выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Введите стороны треугольника:  14  15  16  Введите стороны прямоугольника:  14  16 | Периметр треугльника равен 45  Треугольник разносторонний!  Периметр прямоугольника равен 60  Площадь прямоугольника равна 224 |

Анализ результатов:

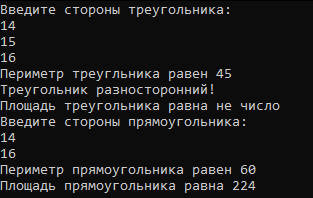


Рисунок 17.1 – Результат работы программы

Источник: собственная разработка

Задание 2. Создать библиотеку flowers и продемонстрировать ее функциональность.

Листинг программы:

public abstract class flowers

{

public abstract void Opisanie();

public abstract void Sr();

public class Roses : flowers

{

public Roses(int cost, string color, int col, string type)

{

this.Cost = cost;

this.color = color;

this.col = col;

this.type = type;

}

public int Cost;

public string color;

public int col;

public string type;

public override void Opisanie()

{

System.Console.WriteLine("Розы {0} Цвет: {1} Количество: {2} Цена: {3}", type, color, col, Cost);

}

public override void Sr()

{

if (col == 0) System.Console.WriteLine("Розы отсутствуют");

if (Cost == 0) System.Console.WriteLine("Розы дорогие");

else System.Console.WriteLine("Розы дешевые");

}

}

public class Tulips : flowers

{

public Tulips(int cost, string color, int col, string type)

{

this.Cost = cost;

this.color = color;

this.col = col;

this.type = type;

}

public int Cost;

public string color;

public int col;

public string type;

public override void Opisanie()

{

System.Console.WriteLine("Тюльпаны {0} Цвет: {1} Количество: {2} Цена: {3}", type, color, col, Cost);

}

public override void Sr()

{

if (col == 0) System.Console.WriteLine("Тюльпаны отсутствует");

if (Cost == 0) System.Console.WriteLine("Тюльпаны дорогая");

else System.Console.WriteLine("Тюльпаны дешевая");

}

}

public class Pionies : flowers

{

public Pionies(int cost, string color, int col, string type)

{

this.Cost = cost;

this.color = color;

this.col = col;

this.type = type;

}

public int Cost;

public string color;

public int col;

public string type;

public override void Opisanie()

{

System.Console.WriteLine("Пионы {0} Цвет: {1} Количество: {2} Цена: {3}", type, color, col, Cost);

}

public override void Sr()

{

if (col == 0) System.Console.WriteLine("Пионы отсутствуют");

if (Cost == 0) System.Console.WriteLine("Пионы дорогие");

else System.Console.WriteLine("Пионы дешевые");

}

}

}

}

class Program

{

static void Main(string[] args)

{

const int n = 3;

flowers[] material = new flowers[n];

material[0] = new flowers.Roses(7, "Красные", 11, "Душистые");

material[1] = new flowers.Tulips(5, "Жёлтые", 3, "Обыкновенные");

material[2] = new flowers.Pionies(6, "Белые", 5, "Японские");

foreach (flowers elem in material)

{

elem.Opisanie();

elem.Sr();

}

Console.ReadLine();

}

}

}

Анализ результатов:

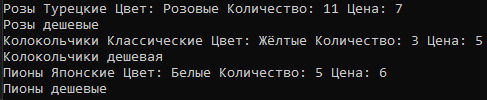


Рисунок 17.2 – Результат работы программы

Источник: собственная разработка