**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:

using System;

public abstract class Shape

{

public abstract double Area();

public abstract double Perimeter();

}

public class Triangle : Shape

{

public double A { get; set; }

public double B { get; set; }

public double C { get; set; }

public Triangle(double a, double b, double c)

{

A = a;

B = b;

C = c;

}

public override double Area()

{

double p = (A + B + C) / 2;

return Math.Sqrt(p \* (p - A) \* (p - B) \* (p - C));

}

public override double Perimeter()

{

return A + B + C;

}

public string Type()

{

if (A == B && B == C)

return "Equilateral";

else if (A == B || B == C || A == C)

return "Isosceles";

else

return "Scalene";

}

}

public class Rectangle : Shape

{

public double A { get; set; }

public double B { get; set; }

public Rectangle(double a, double b)

{

A = a;

B = b;

}

public override double Area()

{

return A \* B;

}

public override double Perimeter()

{

return 2 \* (A + B);

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  |  |

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

Листинг класса:

using System;

public class Phone

{

public string Model { get; private set; }

public string Manufacturer { get; private set; }

public double Price { get; private set; }

public Phone(string model, string manufacturer, double price)

{

Model = model;

Manufacturer = manufacturer;

Price = price;

}

public void ChangeModel(string newModel)

{

Model = newModel;

}

public void ChangeManufacturer(string newManufacturer)

{

Manufacturer = newManufacturer;

}

public void ChangePrice(double newPrice)

{

Price = newPrice;

}

public override string ToString()

{

return $"Model: {Model}, Manufacturer: {Manufacturer}, Price: {Price}";

}

}

Листинг программы:

using System;

class Program

{

static void Main()

{

// Создаем новый объект класса Phone

Phone myPhone = new Phone("iPhone 13", "Apple", 999.99);

// Выводим информацию о телефоне

Console.WriteLine(myPhone.ToString());

// Изменяем модель телефона

myPhone.ChangeModel("iPhone 13 Pro");

Console.WriteLine($"Новая модель: {myPhone.Model}");

// Изменяем производителя телефона

myPhone.ChangeManufacturer("Apple Inc.");

Console.WriteLine($"Новый производитель: {myPhone.Manufacturer}");

// Изменяем цену телефона

myPhone.ChangePrice(1099.99);

Console.WriteLine($"Новая цена: {myPhone.Price}");

}

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Model: iPhone 13, Manufacturer: Apple, Price: 999,99  Новая модель: iPhone 13 Pro  Новый производитель: Apple Inc.  Новая цена: 1099,99 |

Анализ результатов:

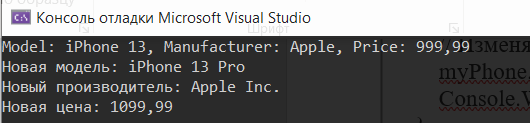


Рисунок 1.2 – Результат работы программы