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

Задания №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).

Рекомендации: создать простую иерархию классов и оформить ее в виде библиотеки, то есть сборки с расширением dll. Класс с наследниками, у всех классов и методов public (класс наследник, поля, конструкторы, методы, свойства).

Листинг программы:

using System;

using TriangleLibrary;

using RectangleLibrary;

class Program

{

static void Main()

{

Triangle triangle = new Triangle(3, 4, 5);

if (triangle.IsValid())

{

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

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

Console.WriteLine($"Тип треугольника: {triangle.GetTriangleType()}");

}

else

{

Console.WriteLine("Треугольник не существует.");

}

Rectangle rectangle = new Rectangle(5, 10);

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

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

}

}

Анализ результатов:

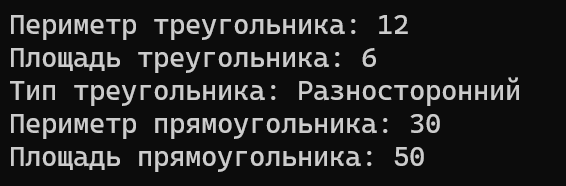


Рисунок 17.1 – Результаты работы программы

Задание №2. Создать свою библиотеку на индивидуальную тему и продемонстрировать ее функциональность. Тема: Игра

Листинг программы:

Using System;

using System.Numerics;

using GameLibrary;

class Program

{

static void Main()

{

Player player = new Player("Анастасия");

player.ShowStats();

Console.WriteLine("\nИгрок получает урон...");

player.TakeDamage(20);

player.ShowStats();

Console.WriteLine("\nИгрок находит предмет...");

Item potion = new Item("Зелье здоровья", 50);

potion.UseItem(player);

player.ShowStats();

}

}

Анализ результатов:

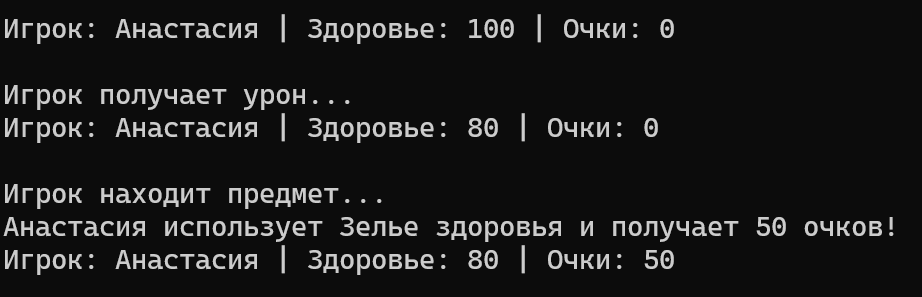


Рисунок 17.2 – Результаты работы программы