**ПРАКТИЧЕСКОЕ ЗАДАНИЕ 17**

**СБОРКИ, БИБЛИОТЕКИ, АТРИБУТЫ, ДИРЕКТИВЫ**

Задание №1. Треугольник (методы ввода сторон, проверки на существование, вычисления периметра, вычисления площади, определения вида треугольника; Прямоугольник.

Листинг программы:

namespace FigureLib

{

public abstract class Figure

{

public abstract string Area();

public abstract string Perimeter();

public abstract string ShapeName();

public abstract string GetFigureType();

}

public class Triangle : Figure

{

double sideA, sideB, sideC;

public Triangle(double triangleSideA, double triangleSideB, double triangleSideC)

{

if (!isExist(triangleSideA, triangleSideB, triangleSideC))

{

throw new Exception($"Error");

}

SideA = triangleSideA;

SideB = triangleSideB;

SideC = triangleSideC;

}

public double SideA

{

get { return sideA; }

set { sideA = value < 0 ? -value : value; }

}

public double SideB

{

get { return sideB; }

set { sideB = value < 0 ? -value : value; }

}

public double SideC

{

get { return sideC; }

set { sideC = value < 0 ? -value : value; }

}

public bool isExist(double sideA, double sideB, double sideC)

{

if (sideA > 0 && sideB > 0 && sideC > 0 && (sideA + sideB > sideC && sideA + sideC > sideB && sideB + sideC > sideA))

{

return true;

}

return false;

}

public override string Area()

{

double semPer = (sideA + sideB + sideC) / 2;

return Math.Sqrt(semPer \* (semPer - sideA) \* (semPer - sideB) \* (semPer - sideC)).ToString();

}

public override string Perimeter()

{

return (sideA + sideB + sideC).ToString();

}

public override string GetFigureType()

{

if (sideA == SideB && sideA == sideC) return "равносторонний";

if (sideA == SideB || sideA == sideC || SideB == sideC) return "равнобедренный";

return "треугольник";

}

public override string ShapeName()

{

return "Треугольник";

}

}

}

namespace ClassLibrary1

{

public abstract class RectangleFigure

{

public abstract string Area();

public abstract string Perimeter();

public abstract string ShapeName();

public abstract string GetFigureType();

}

public class Rectangle : RectangleFigure

{

double width;

double height;

public Rectangle(double rectangleWidth, double rectangleHeight)

{

Width = rectangleWidth;

Height = rectangleHeight;

}

public double Width

{

get { return width; }

set { width = value < 0 ? -value : value; }

}

public double Height

{

get { return height; }

set { height = value < 0 ? -value : value; }

}

public override string Area()

{

return (width \* height).ToString();

}

public override string Perimeter()

{

return (width \* 2 + height \* 2).ToString();

}

public override string GetFigureType()

{

if (width == height) return "Квадрат";

return "Не квадратный прямоугольник";

}

public override string ShapeName()

{

return "Прямоугольник";

}

}

}

using FigureLib;

using ClassLibrary1;

using Otdyx;

try

{

Figure figuretrianle = new Triangle(7, 5, 3);

Console.WriteLine("Фигура треугольник создана с заданными координатами");

RectangleFigure figurerectangle= new Rectangle(10, 10);

Console.WriteLine("Фигура прямоугольник создана с заданными координатами");

OtdyxVip otdyxVip = new OtdyxVip("Грецию", 700, "Бизнес-класс");

Console.WriteLine(otdyxVip.ShowInfo());

A Console.WriteLine("Введите количество денег для оплаты:");

decimal price = decimal.Parse(Console.ReadLine());

Console.WriteLine(otdyxVip.BuyOtdyx(price));

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Console.ReadLine();

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Фигура треугольник создана  Фигура прямоугольник создана |

Анализ результатов представлен на рисунке 1.1.



Рисунок 1.1 – Результат работы программы

Источник: собственная разработка

Задание №2. Создать свою библиотеку на индивидуальную тему и продемонстрировать ее функциональность. (Отдых)

Листинг программы:

namespace Otdyx

{

public class OtdyxVip

{

public string nameOfOtdyx;

public decimal priceOfOtdyx;

public string typeOfOtdyx;

public OtdyxVip(string nameOfOtdyx, decimal priceOfOtdyx, string typeOfOtdyx)

{

this.nameOfOtdyx = nameOfOtdyx;

this.priceOfOtdyx = priceOfOtdyx;

this.typeOfOtdyx = typeOfOtdyx;

}

public string BuyOtdyx(decimal priceOfOtdyx)

{

if (priceOfOtdyx < this.priceOfOtdyx)

{

return "Недостаточно средств, накопите больше денюжек";

}

else

{

return $"Вы купили тур в {nameOfOtdyx}";

}

}

public string ShowInfo()

{

return $"Тур - {nameOfOtdyx}; Цена - {priceOfOtdyx}; Тип отдыха - {typeOfOtdyx};";

}

}

}

using FigureLib;

using ClassLibrary1;

using Otdyx;

try

{

Figure figuretrianle = new Triangle(7, 5, 3);

Console.WriteLine("Фигура треугольник создана с заданными координатами");

RectangleFigure figurerectangle= new Rectangle(10, 10);

Console.WriteLine("Фигура прямоугольник создана с заданными координатами");

OtdyxVip otdyxVip = new OtdyxVip("Грецию", 700, "Бизнес-класс");

Console.WriteLine(otdyxVip.ShowInfo());

Console.WriteLine("Введите количество денег для оплаты:");

decimal price = decimal.Parse(Console.ReadLine());

Console.WriteLine(otdyxVip.BuyOtdyx(price));

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Console.ReadLine();

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 140 | Недостаточно средств, накопите больше денюжек |

Анализ результатов представлен на рисунке 1.2.

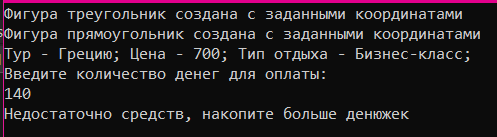


Рисунок 1.2 – Результат работы программы

Источник: собственная разработка