**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:

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, как треугольник";

}

public override string ShapeName()

{

return "Треугольник";

}

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"; }

try

{

Figure figure1 = new Triangle(10, 10, 10);

figure1.ShowInfo();

}

catch (Exception)

{

Console.WriteLine("Error");

}

RectangleFigure figure3 = new Rectangle(5, 5);

figure3.ShowInfo();

Таблица 17.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 10, 10, 10, 5, 5 | 43.3012, 30, 25, 20, равносторонний, квадрат |

Источник: собственная разработка

Анализ результатов представлен на рисунке 17.1.

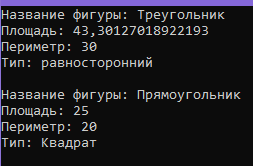


Рисунок 17.1 – Результат работы программы

Источник: собственная разработка

Задание №2. Создать свою библиотеку на индивидуальную тему «Расписание» и продемонстрировать ее функциональность.

Листинг программы:

public class Schedule

{

private DateTime \_startTime;

private DateTime \_endTime;

private string \_name;

private string \_description;

public Schedule(DateTime startTime, DateTime endTime, string name, string description)

{

\_startTime = startTime;

\_endTime = endTime;

\_name = name;

\_description = description;

}

public DateTime StartTime

{

get { return \_startTime; }

set { \_startTime = value; }

}

public DateTime EndTime

{

get { return \_endTime; }

set { \_endTime = value; }

}

public string Name { get { return Name; } set { Name = value; } }

public string Description { get { return Description; } set { Description = value; } }

public string Info()

{

return $"Start time - {\_startTime}, end time - {\_endTime}, name - {\_name}, description - {\_description}";

}

}

Schedule schedal = new Schedule(DateTime.Now, DateTime.Now, "Test", "test schedal");

Console.WriteLine(schedal.Info());

Таблица 17.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 01.06.2022 | 01.06.2022 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 17.2.



Рисунок 17.2 – Результат работы программы

Источник: собственная разработка