

Лабораторная работа №2

Цель работы

Приобретение навыков работы со ссылочными и значимыми типами данных, а также логическими и математическими операторами.

В ходе работы необходимо рассчитать все параметры треугольника. Требуется создать класс, соответствующий треугольнику, запросить у пользователя длины его сторон. В одном из полей класса должна содержаться структура, хранящая рассчитанные параметры треугольника, такие как:

- Площадь треугольника;
- значения углов;
- периметр;
- является ли треугольник
прямоугольным/равносторонним/равнобедренным;
- радиус описанной окружности;
- радиус вписанной окружности.

Упражнение 1. Создание класса треугольника, структуры рассчитываемых параметров и перечисления типа треугольника

Создайте класс треугольника (**Triangle**) со следующими полями:

- Длины сторон.
- Структура (**Struct**), содержащая рассчитанные параметры треугольника.

В свою очередь, структура **TriangleParams** должна содержать перечисление **TriangleType**, указывающее на тип треугольника, образованного по введенным пользователем сторонам.

```
class Triangle
{
    private double _sideAb;
    private double _sideBc;
    private double _sideCa;

    public double SideAb
    {
        get { return _sideAb; }
        set { _sideAb = value; }
    }

    public double SideBc
    {
        get { return _sideBc; }
        set { _sideBc = value; }
    }

    public double SideCa
    {
        get { return _sideCa; }
        set { _sideCa = value; }
    }
}

struct TriangleParams
{
    public double Square;
    public double AngleA;
    public double AngleB;
    public double AngleC;
    public double Perimeter;
    public TriangleType Type;
    public double CircumCircleRadius;
    public double InCircleRadius;
}

public enum TriangleType
{
    EquilateralTriangle,
    IsoscelesTriangle,
    RightTriangle
}
```

Поля класса **SideAb**, **SideBc**, **SideCa** будут запрашиваться у пользователя.

Упражнение 2. Реализация проверки вводимых значений

При объявлении класса мы использовали структуру **get{...}set{...}**, таким образом можно выполнить некоторый код на этапе инициализации переменной. Реализуем проверку вводимых значений следующим образом:

```

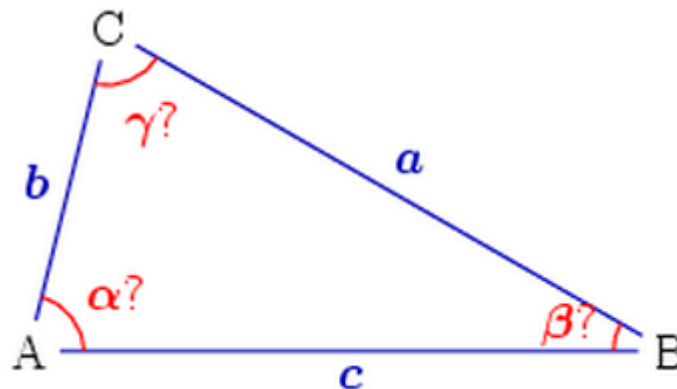
public double SideAb
{
    get { return _sideAb; }
    set
    {
        if (value >= 0)
        {
            throw new Exception("Variable should be greater then zero");
        }
        _sideAb = value;
    }
}

```

Прodelайте то же самое для остальных полей класса.

Самостоятельно реализуйте конструктор класса **Triangle**, в котором выполните проверку на соответствие введенных пользователем значений неравенству треугольника, согласно которому сумма **любых** двух сторон должна быть больше третьей стороны.

Упражнение 3. Расчет параметров треугольника



Согласно указанным далее выражениям рассчитайте все необходимые параметры треугольника:

Площадь треугольника	$S = \sqrt{p \times (p-a) \times (p-b) \times (p-c)}$ <p>где p - полупериметр</p>
Периметр	$(a+b+c)$
Значения углов	$\alpha = \arccos \frac{b^2 + c^2 - a^2}{2bc}$ $\beta = \arccos \frac{a^2 + c^2 - b^2}{2ac}$ <p>Третий угол находится из правила, что сумма всех трёх углов должна быть равна 180°</p>
Радиус описанной окружности	$(a*b*c)/(4*S)$
Радиус вписанной окружности	(S/p) , где p - полупериметр

Примечание

Платформа .Net позволяет рассчитывать арккосинус и арксинус с помощью метода **Acos** и **Asin** класса **Math**, например:

```
double a = Math.Asin(1.5708);
```

При этом значения угла в метод нужно передавать в радианах, формула для перевода градусов в радианы:

Значение в радианах = ((значение в градусах)*Pi/180)

Возведение в степень производится с помощью метода **Pow** класса **Math**:

```
double a = Math.Pow(2,2);
```