class Program

{

static void Main(string[] args)

{

Console.WriteLine("Введите длины сторон треугольника:");

double side1 = GetValidSide("Первая сторона: ");

double side2 = GetValidSide("Вторая сторона: ");

double side3 = GetValidSide("Третья сторона: ");

if (!IsTriangleValid(side1, side2, side3))

{

Console.WriteLine("Треугольник с такими сторонами не существует.");

return;

}

string triangleType = GetTriangleType(side1, side2, side3);

double area = CalculateArea(side1, side2, side3);

string triagnleTypeDegree = GetTriangleTypeDegree(side1, side2, side3);

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

Console.WriteLine($"Тип треугольника по градусам: {triagnleTypeDegree}");

Console.WriteLine($"Площадь треугольника: {area:F2}");

}

static double GetValidSide(string prompt)

{

double side;

while (true)

{

Console.WriteLine(prompt);

if (double.TryParse(Console.ReadLine(), out side) && side > 0)

{

return side;

}

Console.WriteLine("Некорректный ввод. Введите положительное число.");

}

}

static bool IsTriangleValid(double side1, double side2, double side3)

{

return (side1 + side2 > side3) && (side1 + side3 > side2) && (side2 + side3 > side1);

}

static string GetTriangleType(double side1, double side2, double side3)

{

if (side1 == side2 && side2 == side3)

{

return "Равносторонний треугольник";

}

else if (side1 == side2 || side1 == side3 || side2 == side3)

{

return "Равнобедренный треугольник";

}

else

{

return "Разносторонний треугольник";

}

}

static string GetTriangleTypeDegree(double side1, double side2, double side3)

{

double maxSide = Math.Max(side1, Math.Max(side2, side3));

if (Math.Pow(maxSide, 2) == Math.Pow(side1, 2) + Math.Pow(side2, 2) ||

Math.Pow(maxSide, 2) == Math.Pow(side1, 2) + Math.Pow(side3, 2) ||

Math.Pow(maxSide, 2) == Math.Pow(side2, 2) + Math.Pow(side3, 2))

{

return "Прямоугольный треугольник";

}

else if (Math.Pow(maxSide, 2) > Math.Pow(side1, 2) + Math.Pow(side2, 2) ||

Math.Pow(maxSide, 2) > Math.Pow(side1, 2) + Math.Pow(side3, 2) ||

Math.Pow(maxSide, 2) > Math.Pow(side2, 2) + Math.Pow(side3, 2))

{

return "Тупоугольный треугольник";

}

else

{

return "Остроугольный треугольник";

}

}

static double CalculateArea(double side1, double side2, double side3)

{

double p = (side1 + side2 + side3) / 2;

return Math.Sqrt(p \* (p - side1) \* (p - side2) \* (p - side3));

}

}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| А | В | С | Ожидаемый  результат | Результат проверки |
| 3 | 4 | 5 | прямоугольный 6,0 | прямоугольный 6,00 |
| 5 | 5 | 5 | Равносторонний треугольник 10.83 | Равносторонний треугольник 10.83 |
| -1 | 4 | 5 | Не коректный ввод | неккоректный ввод |
| 0 | 1 | 1 | Не коректный ввод | Не коректный ввод |
| ы | 423 | 1 | Не коректный ввод | Не коректный ввод |
| 3 | 5 | 21 | Треугольник с такими сторонами не существует. | Треугольник с такими сторонами не существует. |
| № | 21 | 4 | Треугольник с такими сторонами не существует. | Треугольник с такими сторонами не существует. |

Задание 4

сделать в цикле вайл,чтобы не прекращалось