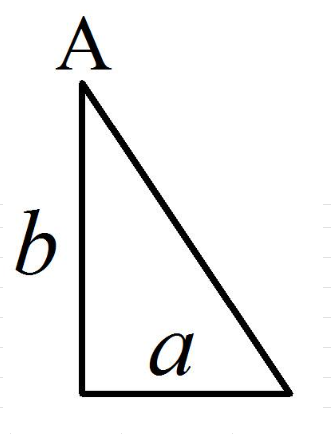
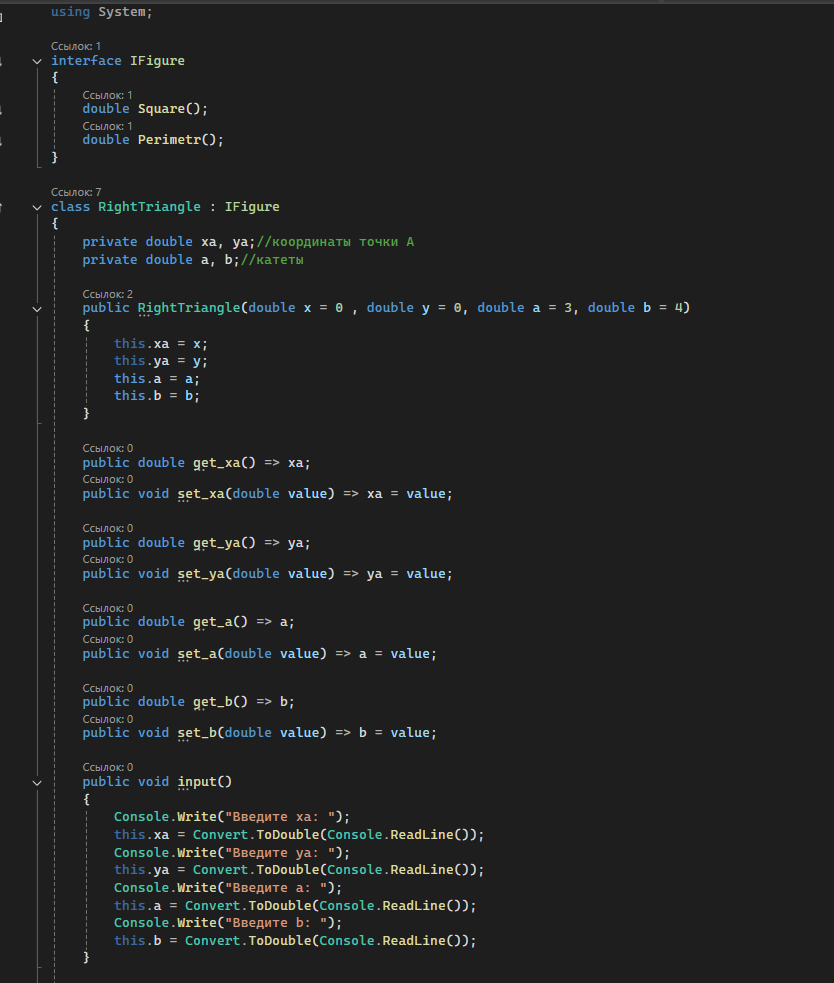
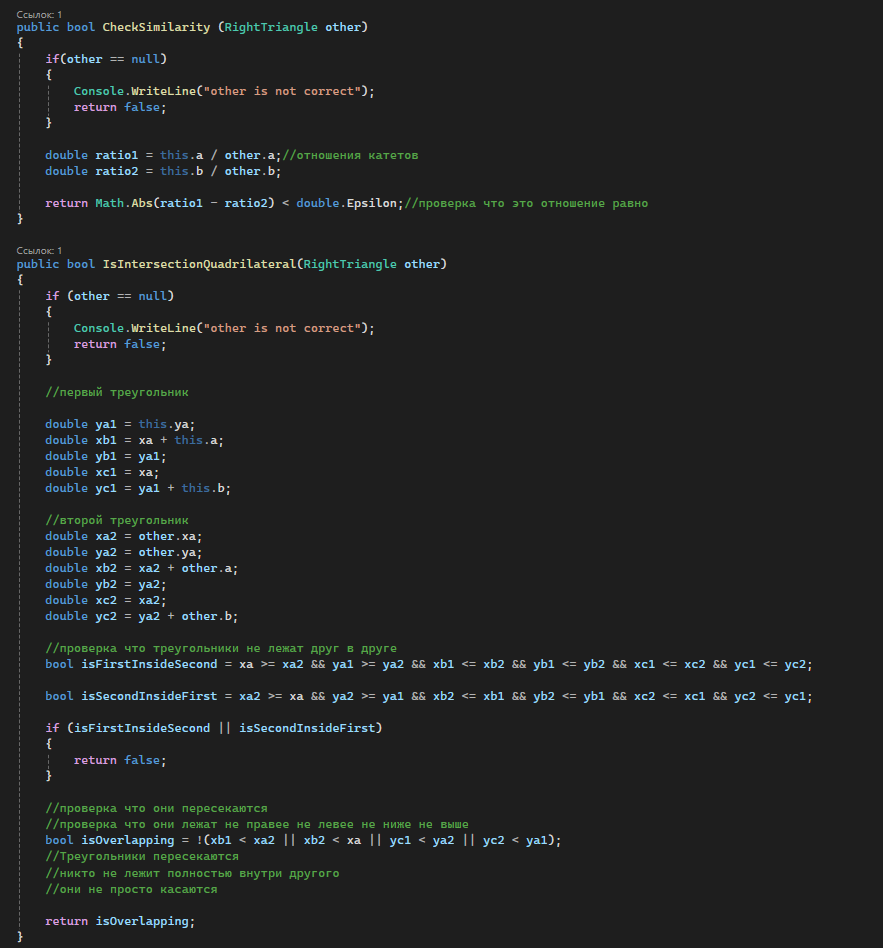
|  |
| --- |
| **2 вариант Класс «Прямоугольный треугольник, со сторонами, параллельными координатным осям» (вариант 2)** |
| Член-данные класса |
| 1) координаты точки A; |
| 2) длины катетов a (горизонтального) и b (вертикального). |
| Методы класса |
| 1) конструкторы: по умолчанию, с аргументами (или с аргументами по умолчанию); |
| 2) ввод/вывод треугольника; |
| 3) нахождение гипотенузы; |
| 4) нахождение периметра; |
| 5) нахождение радиуса описанной окружности; |
| 6) нахождение медианы, проведенной из точки A; |
| 7) нахождение большего из острых углов треугольника; |
| 8) проверка на равнобедренность; |
| 9) нахождение произведения треугольника на число (координаты точки A не меняются, длины катетов умножаются на число); |
| 10) проверка двух треугольников на подобие; |
| 11\*) проверить, имеется ли у двух треугольников пересечение, являющееся четырехугольником. |

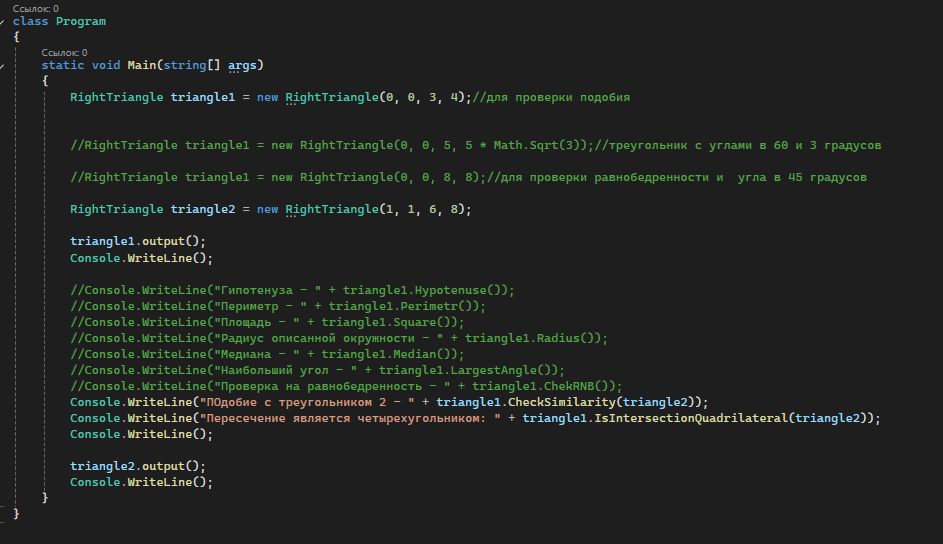


**Код программы**



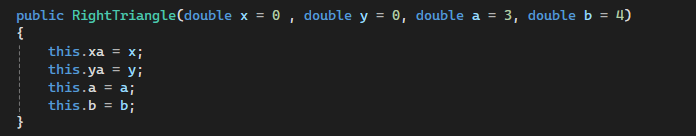






**Тестирование:**

1. Конструктор:



public RightTriangle(double x = 0 , double y = 0, double a = 3, double b = 4)

{

this.xa = x;

this.ya = y;

this.a = a;

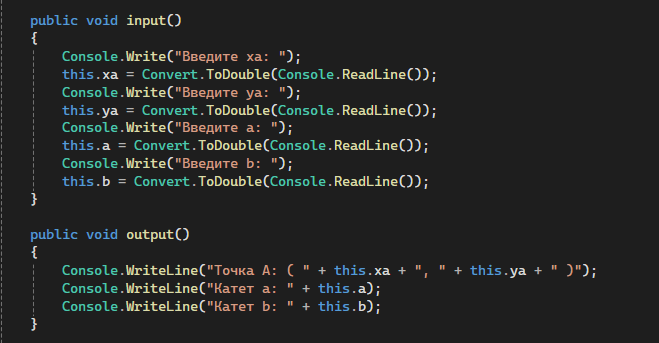
this.b = b;  
}

по умолчанию создаст треугольник с точкой координатами которой являются (0, 0)

и длинами катетов 3 и 4

Так же можно создовать произвольный треугольник

1. Вывод/Ввод треугольника



public void input ()

{

Console.Write("Введите xa: ");

this.xa = Convert.ToDouble(Console.ReadLine());

Console.Write("Введите ya: ");

this.ya = Convert.ToDouble(Console.ReadLine());

Console.Write("Введите a: ");

this.a = Convert.ToDouble(Console.ReadLine());

Console.Write("Введите b: ");

this.b = Convert.ToDouble(Console.ReadLine());

}

public void output ()

{

Console.WriteLine("Точка A: ( " + this.xa + ", " + this.ya + " )");

Console.WriteLine("Катет a: " + this.a);

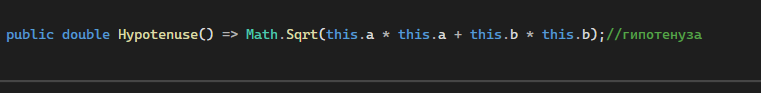
Console.WriteLine("Катет b: " + this.b);

}

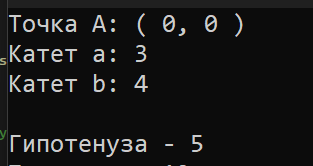
Это метод, который позволяет вводить информацию о треугольнике и выводить в консоль информацию о треугольнике  
Точка A

Катеты a, b

1. Гипотенуза

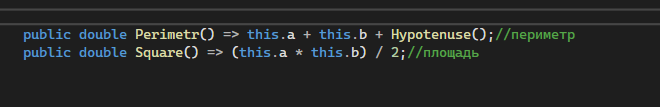


public double Hypotenuse () => Math.Sqrt(this.a \* this.a + this.b \* this.b);



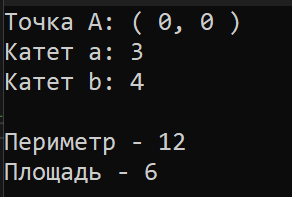
Для стандартного треугольника 3^2 + 4^2 = 25 = 5^2 => гипотенуза = 5

4)Периметр/Площадь



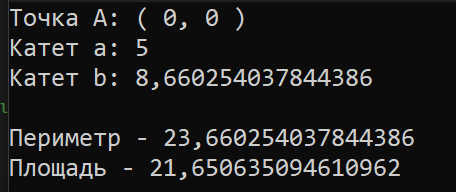
public double Perimetr() => this.a + this.b + Hypotenuse();//периметр

public double Square() => (this.a \* this.b) / 2;//площадь

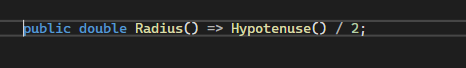


для стандартного треугольника периметр 3 + 4 + гипотенуза(=5) = 12

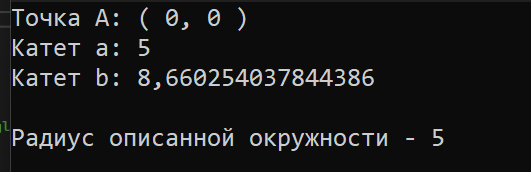
площадь по формуле ½ \* a \* b = ½ \* 3 \* 4 = 6



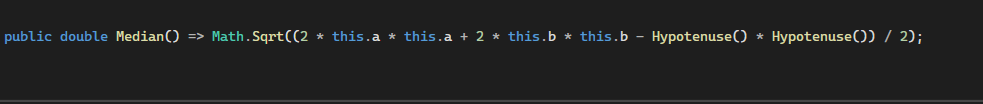
5)Радиус описанной окружности



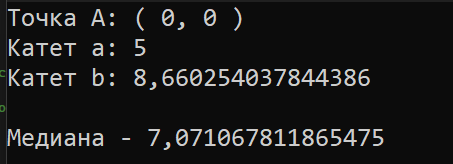
public double Radius() => Hypotenuse() / 2;



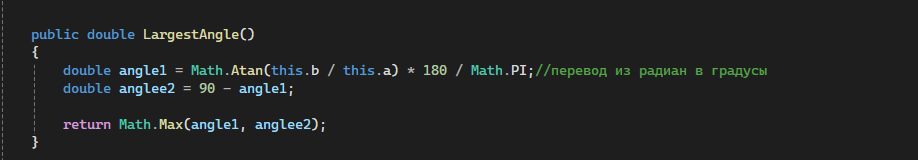
5)Медиана



public double Median() => Math.Sqrt((2 \* this.a \* this.a + 2 \* this.b \* this.b - Hypotenuse() \* Hypotenuse()) / 2);



6)Наибольший угол



public double LargestAngle()

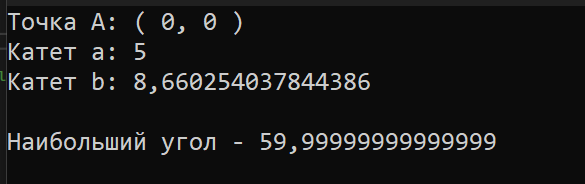
{

double angle1 = Math.Atan(this.b / this.a) \* 180 / Math.PI;//перевод из радиан в градусы

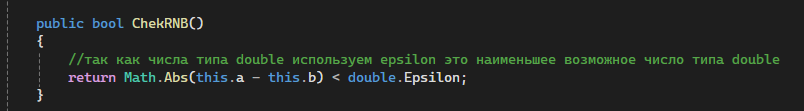
double anglee2 = 90 - angle1;

return Math.Max(angle1, anglee2);

}



7)Проверка на равнобедренность



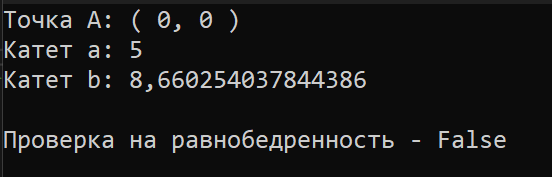
public bool ChekRNB()

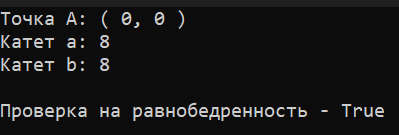
{

//так как числа типа double используем epsilon это наименьшее возможное число типа double

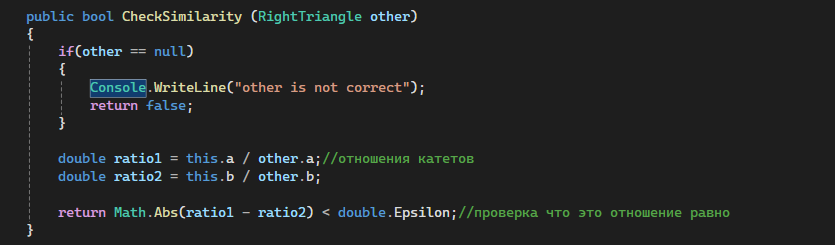
return Math.Abs(this.a - this.b) < double.Epsilon;

}





8)Проверка на подобие



public bool CheckSimilarity (RightTriangle other)

{

if(other == null)

{

Console.WriteLine("other is not correct");

return false;

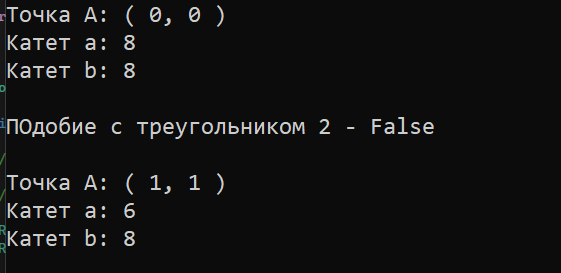
}

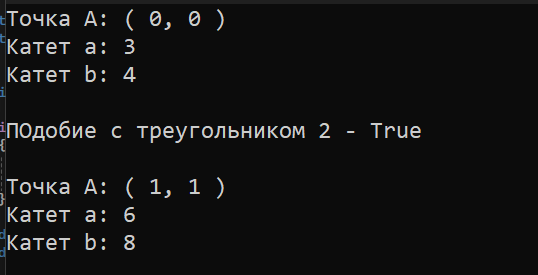
double ratio1 = this.a / other.a;//отношения катетов

double ratio2 = this.b / other.b;

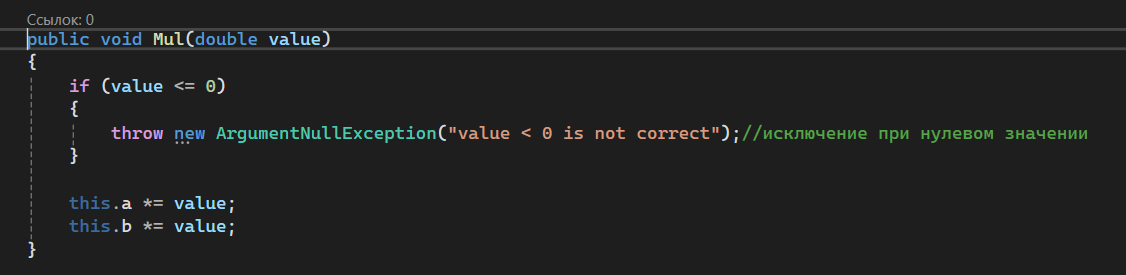
return Math.Abs(ratio1 - ratio2) < double.Epsilon;//проверка что это отношение равно

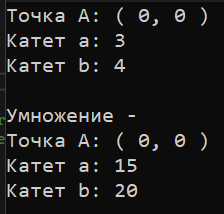
}



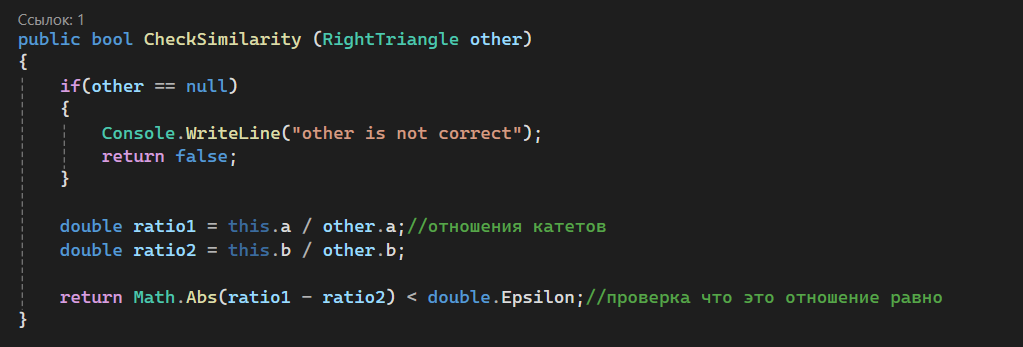


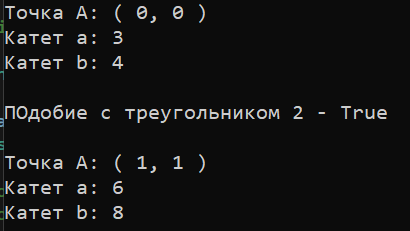
9) нахождение произведения треугольника на число (координаты точки A не меняются, длины катетов умножаются на число)





10) проверка двух треугольников на подобие





11\*) проверить, имеется ли у двух треугольников пересечение, являющееся четырехугольником.  
