Университет Синергия

Лабораторная работа № 3

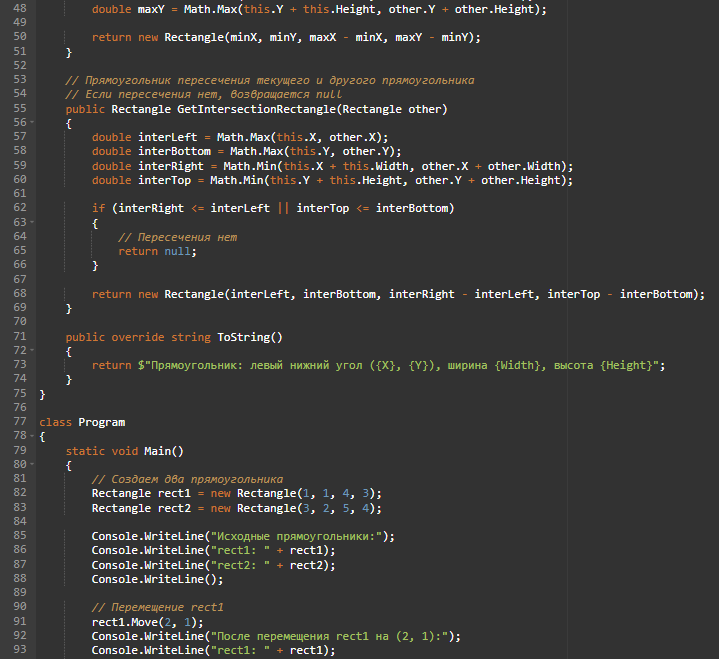
Тема: Простейшие классы

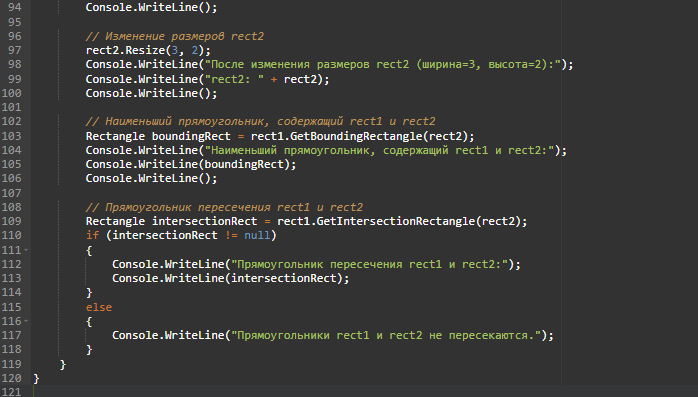
Ошурков Александр ДКИП-203прог

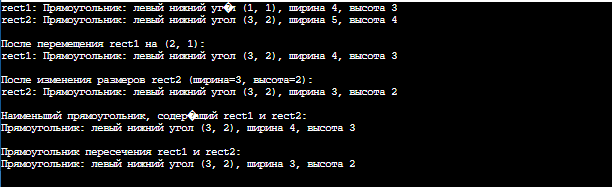
Москва 2024

**Задание 1:**

****

****

****

**using System;**

**class Rectangle**

**{**

**// Координаты левого нижнего угла**

**public double X { get; private set; }**

**public double Y { get; private set; }**

**// Размеры прямоугольника**

**public double Width { get; private set; }**

**public double Height { get; private set; }**

**// Конструктор**

**public Rectangle(double x, double y, double width, double height)**

**{**

**if (width < 0 || height < 0)**

**throw new ArgumentException("Ширина и высота должны быть неотрицательными.");**

**X = x;**

**Y = y;**

**Width = width;**

**Height = height;**

**}**

**// Перемещение прямоугольника на dx, dy**

**public void Move(double dx, double dy)**

**{**

**X += dx;**

**Y += dy;**

**}**

**// Изменение размеров**

**public void Resize(double newWidth, double newHeight)**

**{**

**if (newWidth < 0 || newHeight < 0)**

**throw new ArgumentException("Ширина и высота должны быть неотрицательными.");**

**Width = newWidth;**

**Height = newHeight;**

**}**

**// Наименьший прямоугольник, содержащий текущий и другой прямоугольник**

**public Rectangle GetBoundingRectangle(Rectangle other)**

**{**

**double minX = Math.Min(this.X, other.X);**

**double minY = Math.Min(this.Y, other.Y);**

**double maxX = Math.Max(this.X + this.Width, other.X + other.Width);**

**double maxY = Math.Max(this.Y + this.Height, other.Y + other.Height);**

**return new Rectangle(minX, minY, maxX - minX, maxY - minY);**

**}**

**// Прямоугольник пересечения текущего и другого прямоугольника**

**// Если пересечения нет, возвращается null**

**public Rectangle GetIntersectionRectangle(Rectangle other)**

**{**

**double interLeft = Math.Max(this.X, other.X);**

**double interBottom = Math.Max(this.Y, other.Y);**

**double interRight = Math.Min(this.X + this.Width, other.X + other.Width);**

**double interTop = Math.Min(this.Y + this.Height, other.Y + other.Height);**

**if (interRight <= interLeft || interTop <= interBottom)**

**{**

**// Пересечения нет**

**return null;**

**}**

**return new Rectangle(interLeft, interBottom, interRight - interLeft, interTop - interBottom);**

**}**

**public override string ToString()**

**{**

**return $"Прямоугольник: левый нижний угол ({X}, {Y}), ширина {Width}, высота {Height}";**

**}**

**}**

**class Program**

**{**

**static void Main()**

**{**

**// Создаем два прямоугольника**

**Rectangle rect1 = new Rectangle(1, 1, 4, 3);**

**Rectangle rect2 = new Rectangle(3, 2, 5, 4);**

**Console.WriteLine("Исходные прямоугольники:");**

**Console.WriteLine("rect1: " + rect1);**

**Console.WriteLine("rect2: " + rect2);**

**Console.WriteLine();**

**// Перемещение rect1**

**rect1.Move(2, 1);**

**Console.WriteLine("После перемещения rect1 на (2, 1):");**

**Console.WriteLine("rect1: " + rect1);**

**Console.WriteLine();**

**// Изменение размеров rect2**

**rect2.Resize(3, 2);**

**Console.WriteLine("После изменения размеров rect2 (ширина=3, высота=2):");**

**Console.WriteLine("rect2: " + rect2);**

**Console.WriteLine();**

**// Наименьший прямоугольник, содержащий rect1 и rect2**

**Rectangle boundingRect = rect1.GetBoundingRectangle(rect2);**

**Console.WriteLine("Наименьший прямоугольник, содержащий rect1 и rect2:");**

**Console.WriteLine(boundingRect);**

**Console.WriteLine();**

**// Прямоугольник пересечения rect1 и rect2**

**Rectangle intersectionRect = rect1.GetIntersectionRectangle(rect2);**

**if (intersectionRect != null)**

**{**

**Console.WriteLine("Прямоугольник пересечения rect1 и rect2:");**

**Console.WriteLine(intersectionRect);**

**}**

**else**

**{**

**Console.WriteLine("Прямоугольники rect1 и rect2 не пересекаются.");**

**}**

**}**

**}**