ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА» (СПбГУТ)

КАФЕДРА АВТОМАТИЗАЦИИ ПРЕДПРИЯТИЙ СВЯЗИ

ПРОЕКТИРОВАНИЕ ЕДИНОГО ИНФОРМАЦИОННОГО ПРОСТРАНСТВА АВТОМАТИЗИРОВАННЫХ ПРОИЗВОДСТВ

Определение свойств с использованием лямбда-выражений

```
class Point
    private double X;
    ссылок: 7
    public double X { get => _X; set => _X = value; }
    private double Y;
    ссылок: 7
    public double Y { get => Y; set => Y = value; }
    ссылка: 1
    public void MoveBy(double dx, double dy)
       X += dx;
       Y +=dy;
```

Конструктор с параметрами

```
class Point
    ссылок: 8
    public double X { get; set; }
    ссылок: 8
    public double Y { get; set; }
    ссылок: 2
    public Point (double x, double y)
      X = X;
       Y = y;
    ссылка: 1
    public void MoveBy(double dx, double dy)
        X += dx;
        Y += dy;
```

Перегрузка конструктора

```
class Point
    ссылок: 9
    public double X { get; set; }
    ссылок: 9
    public double Y { get; set; }
    ссылок: 2
    public Point()
      X = 0;
      Y = 0;
    ссылок: 0
    public Point (double x, double y)
      X = X;
       Y = y;
    ссылка: 1
    public void MoveBy(double dx, double dy)
       X += dx;
       Y +=dy;
```

Создание объектов с помощью конструктора с параметрами и без параметров

```
static void Main(string[] args)
{
    Point p1 = new Point();
    p1.X = 5;
    p1.Y = 10;
    Console.WriteLine("P1.X = {0}, P1.Y = {1}", p1.X, p1.Y);

    Point p2 = new Point(15, 20);
    Console.WriteLine("P2.X = {0}, P2.Y = {1}", p2.X, p2.Y);

    p1.MoveBy(50, 100);
    Console.WriteLine("P1.X = {0}, P1.Y = {1}", p1.X, p1.Y);
    Console.WriteLine("P2.X = {0}, P2.Y = {1}", p2.X, p2.Y);
}
```

```
class Program
{
     ccылок: 0
     static void Main(string[] args)
     {
          Point p1 = new Point(5, 10);
          Point p2 = new Point(p1.X, p1.Y);
          Console.WriteLine(p1.Position);
     }
}
```

Копирующий конструктор

```
public Point (Point point)
{
    X = point.X;
    Y = point.Y;
}
```

```
static void Main(string[] args)
{
    Point p1 = new Point(5, 10);
    Point p2 = new Point(p1.X, p1.Y);
    Point p3 = new Point(p2);

    Console.WriteLine(p3.Position);
}
```

Метод клонирования объекта

```
public Point Clone()
   return new Point(X, Y);
                              static void Main(string[] args)
                                  Point p1 = new Point(5, 10);
                                  Point p2 = new Point(p1.X, p1.Y);
                                  Point p3 = new Point(p2);
                                  Point p4 = p3.Clone();
                                  Console.WriteLine(p1.Position);
                                  Console.WriteLine(p2.Position);
 X = 5, Y = 10
                                  Console.WriteLine(p3.Position);
X = 5, Y = 10
                                  Console.WriteLine(p4.Position);
```

Код класса Point

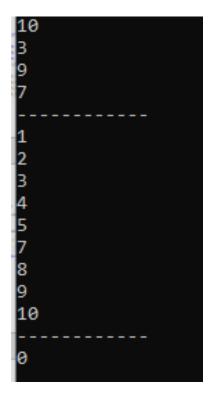
```
class Point
    ссылок: 8
    public double X { get; set; }
    ссылок: 8
    public double Y { get; set; }
    ссылок: 0
    public Point()
        X = 0;
        Y = 0;
    ссылок: 3
    public Point (double x, double y)
        X = X;
        Y = y;
    ссылка: 1
    public Point (Point point)
        X = point.X;
        Y = point.Y;
```

```
ссылка: 1
public Point Clone()
    return new Point(X, Y);
ссылок: 4
public string Position
    get { return "X = " + X + ", Y = " + Y; }
ссылок: 0
public void MoveBy(double dx, double dy)
    X += dx;
    Y +=dy;
```

Классы коллекций

```
static void Main(string[] args)
   List<int> numbers = new List<int>();
   for(int i = 1; i < 11; i++)
       numbers.Add(i);
   Console.WriteLine(numbers.Count);
   Console.WriteLine(numbers[2]);
   numbers.RemoveAt(5);
   Console.WriteLine(numbers.Count);
   Console.WriteLine(numbers[5]);
   Console.WriteLine("----");
   foreach(int number in numbers)
       Console.WriteLine(number);
   Console.WriteLine("----");
   numbers.Clear();
   Console.WriteLine(numbers.Count);
```

```
static void Main(string[] args)
   List<int> numbers = new List<int>();
   for(int i = 1; i < 11; i++)
       numbers.Add(i);
   Console.WriteLine(numbers.Count);
   Console.WriteLine(numbers[2]);
   numbers.RemoveAt(5);
   Console.WriteLine(numbers.Count);
   Console.WriteLine(numbers[5]);
   Console.WriteLine("----");
   foreach(int number in numbers)
       Console.WriteLine(number);
   Console.WriteLine("----");
   numbers.Clear();
   Console.WriteLine(numbers.Count);
```



Работа со списком точек

```
static void Main(string[] args)
   List<Point> points = new List<Point>();
   for(int i = 0; i < 11; i++)
        points.Add(new Point(i, i));
   foreach(Point point in points)
       point.MoveBy(10, 10);
   foreach(Point point in points)
       Console.WriteLine(point.Position);
```