

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

**ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ
ПРОГРАММИРОВАНИЕ В УПРАВЛЕНИИ
ТЕХНИЧЕСКИМИ СИСТЕМАМИ**

2020

Классы и объекты

Класс точек

```
using System;
using System.Collections.Generic;
using System.Text;

namespace ConsoleApp1
{
    ссылка: 0
    class Point
    {
        public double X;
        public double Y;

        ссылка: 0
        public void MoveBy(double dx, double dy)
        {
            X += dx;
            Y += dy;
        }
    }
}
```

Объекты класса Point

```
using System;
```

```
namespace ConsoleApp1
{
    ссылка: 0
    class Program
    {
        ссылка: 0
        static void Main(string[] args)
        {
            Point p1 = new Point();
            p1.X = 5.0;
            p1.Y = 10.0;
            Console.WriteLine("P1.X = {0}, P1.Y = {1}", p1.X, p1.Y);

            Point p2 = new Point();
            p2.X = 15.0;
            p2.Y = 20.0;
            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);
        }
    }
}
```

```
P1.X = 5, P1.Y = 10
P2.X = 15, P2.Y = 20
P1.X = 55, P1.Y = 110
P2.X = 15, P2.Y = 20
```

```
class Point
{
    private double _X;
    ссылка: 7
    public double X
    {
        get { return _X; }
        set { _X = value; }
    }

    private double _Y;
    ссылка: 7
    public double Y
    {
        get { return _Y; }
        set { _Y = value; }
    }

    ссылка: 1
    public void MoveBy(double dx, double dy)
    {
        X += dx;
        Y += dy;
    }
}
```

Автоматические свойства

```
class Point
{
    ссылка: 7
    public double X { get; set; }
    ссылка: 7
    public double Y { get; set; }

    ссылка: 1
    public void MoveBy(double dx, double dy)
    {
        X += dx;
        Y += dy;
    }
}
```

Определение свойств с использованием лямбда-выражений

```
class Point
{
    private double _X;
    ссылка: 7
    public double X { get => _X; set => _X = value; }

    private double _Y;
    ссылка: 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
{
    ссылка: 8
    public double X { get; set; }
    ссылка: 8
    public double Y { get; set; }

    ссылка: 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
{
    ссылка: 9
    public double X { get; set; }
    ссылка: 9
    public double Y { get; set; }

    ссылка: 2
    public Point()
    {
        X = 0;
        Y = 0;
    }

    ссылка: 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
{
    ссылка: 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);  
}
```

```
X = 5, Y = 10  
X = 5, Y = 10  
X = 5, Y = 10  
X = 5, Y = 10
```

```
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);  
    Console.WriteLine(p3.Position);  
    Console.WriteLine(p4.Position);  
}
```

Код класса Point

```
class Point
{
    ссылка: 8
    public double X { get; set; }
    ссылка: 8
    public double Y { get; set; }

    ссылка: 0
    public Point()
    {
        X = 0;
        Y = 0;
    }

    ссылка: 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);
}

ссылка: 4
public string Position
{
    get { return "X = " + X + ", Y = " + Y; }
}

ссылка: 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);
}

```

```

10
3
9
7
-----
1
2
3
4
5
7
8
9
10
-----
0

```

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);
    }
}
```