

## Вариант № 1

Создать класс Point, разработав следующие элементы класса:

- Поля:
  - `int x, y;`
- Конструкторы, позволяющие создать экземпляр класса:
  - с нулевыми координатами;
  - с заданными координатами.
- Методы, позволяющие:
  - вывести координаты точки на экран;
  - рассчитать расстояние от начала координат до точки;
  - переместить точку на плоскости на вектор (a, b).
- Свойства:
  - получить-установить координаты точки (доступное для чтений и записи);
  - позволяющие умножить координаты точки на скаляр (доступное для чтений и записи).

SHLb\_4 Project, Form1.cs File

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using PointSpace;

namespace SHLb_4
{
    public partial class Form1 : Form
    {
        PointClass point;

        public Form1()
        {
            InitializeComponent();
        }

        private void CreateClass_Click(object sender, EventArgs e)
        {
            if (create_x.Text != String.Empty && create_y.Text != String.Empty)
            {
                point = new PointClass(int.Parse(create_x.Text),
int.Parse(create_y.Text));
                OutBox.AppendText("Точка создана.\n");
            } else {
                point = new PointClass();
                OutBox.AppendText("Точка создана, с нулевыми координатами.\n");
            }

            PrintCord.Enabled = true;
            SetCord.Enabled = true;
            MovePoint.Enabled = true;
            PrintDistance.Enabled = true;
        }
    }
}
```

```

        CreateClass.Enabled = false;
    }

    private void PrintCord_Click(object sender, EventArgs e)
    {
        OutBox.AppendText(point.PrintCord() + "\n");
    }

    private void SetCord_Click(object sender, EventArgs e)
    {
        if (set_x.Text != String.Empty && set_y.Text != String.Empty)
        {
            point.x = int.Parse(set_x.Text);
            point.y = int.Parse(set_y.Text);
            OutBox.AppendText("Установлены новые координаты точки.\n");
        }
    }

    private void MovePoint_Click(object sender, EventArgs e)
    {
        if (vector_a.Text != String.Empty && vector_b.Text != String.Empty)
        {
            point.MultiplyX = int.Parse(vector_a.Text);
            point.MultiplyY = int.Parse(vector_b.Text);
            OutBox.AppendText("Точка перемещена на вектор (a,b).\n");
        }
    }

    private void PrintDistance_Click(object sender, EventArgs e)
    {
        OutBox.AppendText("Расстояние до начала координат: " + point.CalcDistance() +
"\n");
    }
}

```

#### PointClass Project, PointClass.cs File

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace PointSpace
{
    /// <summary>
    /// Класс для работы с координатами точки
    /// </summary>
    public class PointClass
    {
        protected int _x;
        protected int _y;

        /// <summary>
        /// Установить т. X
        /// </summary>
        public int x
        {
            get { return this._x; }
            set { this._x = value; }
        }
    }
}

```

```

/// <summary>
/// Установить т. Y
/// </summary>
public int y
{
    get { return this._y; }
    set { this._y = value; }
}

/// <summary>
/// Конструктор по умолчанию
/// </summary>
public PointClass()
{
    this.x = 0;
    this.y = 0;
}

/// <summary>
/// Конструктор с параметрами
/// </summary>
/// <param name="x">Координаты по X.</param>
/// <param name="y">Координаты по Y.</param>
public PointClass(int x, int y)
{
    this.x = x;
    this.y = y;
}

/// <summary>
/// Вывести координаты точки на экран
/// </summary>
/// <returns>Строка с текущими координатами.</returns>
public string PrintCord()
{
    if (x != null && y != null)
        return String.Format("x = {0}, y = {1}", x.ToString(), y.ToString());
    else
        return String.Format("Данные не заполнены.");
}

/// <summary>
/// Рассчитать расстояние от начала координат до точки
/// </summary>
/// <returns>Результаты вычисления расстояния от точки до начала
координат.</returns>
public int CalcDistance()
{
    return (int)Math.Sqrt(Math.Pow(0 - this.x, 2) + Math.Pow(0 - this.y, 2));
}

/// <summary>
/// Переместить точку на плоскости на вектор (a, b)
/// </summary>
/// <param name="x">Параметр a, вектора (a, b).</param>
/// <param name="y">Параметр b, вектора (a, b).</param>
public void MovePoint(int x, int y)
{
    this.x += x;
    this.y += y;
}

/* Свойства позволяющие умножить координаты точки на скаляр (доступное для чтений
и записи) */
/// <summary>

```

```

    /// Умножить координаты т. X на скаляр
    /// </summary>
    /// <value>Возвращает текущее значение точки, по оси X. Записывает в точку ее
значение умноженное на скаляр.</value>
    public double MultiplyX
    {
        get { return x; }
        set { x = (int) (value * x); }
    }

    /// <summary>
    /// Умножить координаты т. Y на скаляр
    /// </summary>
    /// <value>Возвращает текущее значение точки, по оси Y. Записывает в точку ее
значение умноженное на скаляр.</value>
    public double MultiplyY
    {
        get { return y; }
        set { y = (int)(value * y); }
    }
}

```

PointClass.XML

```

<?xml version="1.0"?>
<doc>
    <assembly>
        <name>PointClass</name>
    </assembly>
    <members>
        <member name="T:PointSpace.PointClass">
            <summary>
                Класс для работы с координатами точки
            </summary>
        </member>
        <member name="M:PointSpace.PointClass.#ctor">
            <summary>
                Конструктор по умолчанию
            </summary>
        </member>
        <member name="M:PointSpace.PointClass.#ctor(System.Int32,System.Int32)">
            <summary>
                Конструктор с параметрами
            </summary>
            <param name="x">Координаты по X.</param>
            <param name="y">Координаты по Y.</param>
        </member>
        <member name="M:PointSpace.PointClass.PrintCord">
            <summary>
                Вывести координаты точки на экран
            </summary>
            <returns>Строка с текущими координатами.</returns>
        </member>
        <member name="M:PointSpace.PointClass.CalcDistance">
            <summary>
                Рассчитать расстояние от начала координат до точки
            </summary>
            <returns>Результаты вычисления расстояния от точки до начала
координат.</returns>
        </member>
        <member name="M:PointSpace.PointClass.MovePoint(System.Int32,System.Int32)">
            <summary>
                Переместить точку на плоскости на вектор (a, b)
            </summary>
            <param name="x">Параметр a, вектора (a, b).</param>

```

```

        <param name="y">Параметр b, вектора (a, b).</param>
    </member>
    <member name="P:PointSpace.PointClass.x">
        <summary>
            Установить т. X
        </summary>
    </member>
    <member name="P:PointSpace.PointClass.y">
        <summary>
            Установить т. Y
        </summary>
    </member>
    <member name="P:PointSpace.PointClass.MultiplyX">
        <summary>
            Умножить координаты т. X на скаляр
        </summary>
        <value>Возвращает текущее значение точки, по оси X. Записывает в точку ее
значение умноженное на скаляр.</value>
    </member>
    <member name="P:PointSpace.PointClass.MultiplyY">
        <summary>
            Умножить координаты т. Y на скаляр
        </summary>
        <value>Возвращает текущее значение точки, по оси Y. Записывает в точку ее
значение умноженное на скаляр.</value>
    </member>
</members>
</doc>

```

Лабораторная работа №4

Создать класс

Начальные координаты  
X: 1 Y: 2

Переместить точку на вектор

Вектор  
a: 4 b: 1

Установить координаты точки

Координаты точки  
X: 3 Y: 4

Вывести текущие координаты

Вывести расстояние до начала координат

Точка создана.  
Точка перемещена на вектор (a,b).  
x = 4, y = 12  
Установлены новые координаты точки.  
x = 3, y = 4  
Расстояние до начала координат: 5  
Точка перемещена на вектор (a,b).  
x = 12, y = 4