



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

О Т Ч Е Т

по лабораторной работе № 11

Название: Интерфейсы, сортировка с компаратором

Дисциплина: Разработка приложений на языке C#

Студент

ИУ6-75Б

(Группа)

(Подпись, дата)

И.Ю. Жосан

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

А.М. Минитаева

(И.О. Фамилия)

Москва, 2023

Задание:

1. Создать класс Point - точка на плоскости с вещественными координатами x, y. Создать конструктор, ToString() и свойства для доступа к координатам точки.

2. Создайте метод, который генерирует набор (List<Point>) случайно расположенных точек в квадрате [0,1]x[0,1].

3. Используя интерфейс IComparer, выведите все точки, упорядочивая их следующими способами:

- по удалению от начала координат (сначала выводится ближайшая к началу координат, порядок равноудалённых точек не важен);
- по удалению от оси абсцисс (сначала выводится ближайшая к оси абсцисс, порядок равноудалённых точек не важен);
- по удалению от оси ординат (сначала выводится ближайшая к оси ординат, порядок равноудалённых точек не важен);
- по удалению от диагонали первой и третьей четвертей (прямая $y=x$, порядок равноудалённых точек не важен).

Выполнение задания:

Был разработан класс для создания объектов типа точка на плоскости:

```
class Point
{
    protected double x;
    protected double y;
    public Point(double a, double b, int n)
    {
        x = a;
        y = b;
        num = n;
    }
    public override string ToString()
    {
        Return $"{num}.({Math.Round(x, 3)},
                                           {Math.Round(y, 3)})";
    }
}
```

```

        public double X
        {
            get { return x; }
            set { x = value; }
        }
        public double Y
        {
            get { return y; }
            set { y = value; }
        }
    }

```

Метод, генерирующий список случайных точек:

```

static List<Point> generateListOfPoints(int n)
{
    List<Point> result = new List<Point>();
    Random random = new Random();
    for (int i = 0; i < n; i++)
        result.Add(new Point(r.NextDouble(), r.NextDouble(), i+1));
    return result;
}

```

Классы, созданные для сравнения точек по разным признакам и унаследованные от IComparer:

```

class PointComparer_by_begin : IComparer<Point>
{
    public int Compare(Point x, Point y)
    {
        double len1 = lengthBetweenPoints(x.X, x.Y, 0.0, 0.0);
        double len2 = lengthBetweenPoints(0.0, 0.0, y.X, y.Y);
        return CompareLengths(len1, len2);
    }
}

class PointComparer_by_abscis : IComparer<Point>
{
    public int Compare(Point x, Point y)
    {
        return CompareLengths(x.Y, y.Y);
    }
}

```

```

        }
    }
class PointComparer_by_ordinate : IComparer<Point>
{
    public int Compare(Point x, Point y)
    {
        return CompareLengths(x.X, y.X);
    }
}
class PointComparer_by_diagonal : IComparer<Point>
{
    public int Compare(Point x, Point y)
    {
        double len1 = Math.Abs(x.X - x.Y) / Math.Sqrt(2);
        double len2 = Math.Abs(y.X - y.Y) / Math.Sqrt(2);
        return CompareLengths(len1, len2);
    }
}
static double lengthBetweenPoints(double x1, double y1, double x2,
                                   double y2)
{
    return Math.Sqrt(Math.Pow(x2-x1, 2)+Math.Pow(y2-y1, 2));
}
static int CompareLengths(double len1, double len2)
{
    if (len1 == len2) return 0;
    else if (len1 < len2) return -1;
    else return 1;
}

```

Основная программа:

```

static void Main(string[] args)
{
    List<Point> points =new List<Point>(generateListOfPoints(5));
    Console.WriteLine("Random list of points:");
    for (int i = 0; i < points.Count; i++)
        Console.WriteLine(points[i]);
}

```

```

        Console.WriteLine("\nSorted list by begin (0, 0):");
        points.Sort(new PointComparer_by_begin());
        for (int i = 0; i < points.Count; i++)
        {
            double len = lengthBetweenPoints(points[i].X,
                                                points[i].Y, 0.0, 0.0);
            Console.WriteLine($"{points[i]} \t{Math.Round(len, 6)}");
        }

        Console.WriteLine("\nSorted list by abscis y=0:");
        points.Sort(new PointComparer_by_abscis());
        for (int i = 0; i < points.Count; i++)
            Console.WriteLine($"{points[i]} \t
                                {Math.Round(points[i].Y, 6)}");

        Console.WriteLine("\nSorted list by ordinate x=0:");
        points.Sort(new PointComparer_by_ordinate());
        for (int i = 0; i < points.Count; i++)
            Console.WriteLine($"{points[i]} \t
                                {Math.Round(points[i].X, 6)}");

        Console.WriteLine("\nSorted list by diagonal y=x:");
        points.Sort(new PointComparer_by_diagonal());
        for (int i = 0; i < points.Count; i++)
        {
            double len = Math.Abs(points[i].X - points[i].Y) /
                                Math.Sqrt(2);
            Console.WriteLine($"{points[i]} \t {Math.Round(len, 6)}");
        }

        Console.ReadLine();
    }

```

Пример выполнения программы приведен на рисунке 1.

```
Random list of points:
1. (0,459, 0,144)
2. (0,013, 0,778)
3. (0,955, 0,147)
4. (0,637, 0,782)
5. (0,684, 0,422)

Sorted list by begin (0, 0):
1. (0,459, 0,144)      0,481301
2. (0,013, 0,778)      0,778086
5. (0,684, 0,422)      0,803923
3. (0,955, 0,147)      0,965912
4. (0,637, 0,782)      1,008595

Sorted list by abscis y=0:
1. (0,459, 0,144)      0,143882
3. (0,955, 0,147)      0,14729
5. (0,684, 0,422)      0,421918
2. (0,013, 0,778)      0,777976
4. (0,637, 0,782)      0,781965

Sorted list by ordinate x=0:
2. (0,013, 0,778)      0,013115
1. (0,459, 0,144)      0,459292
4. (0,637, 0,782)      0,63702
5. (0,684, 0,422)      0,684308
3. (0,955, 0,147)      0,954616

Sorted list by diagonal y=x:
4. (0,637, 0,782)      0,102491
5. (0,684, 0,422)      0,185537
1. (0,459, 0,144)      0,223028
2. (0,013, 0,778)      0,540838
3. (0,955, 0,147)      0,570865
```

Рисунок 1 – Выполнение программы

Проверка результата на координатной плоскости приведена на рисунке 2.

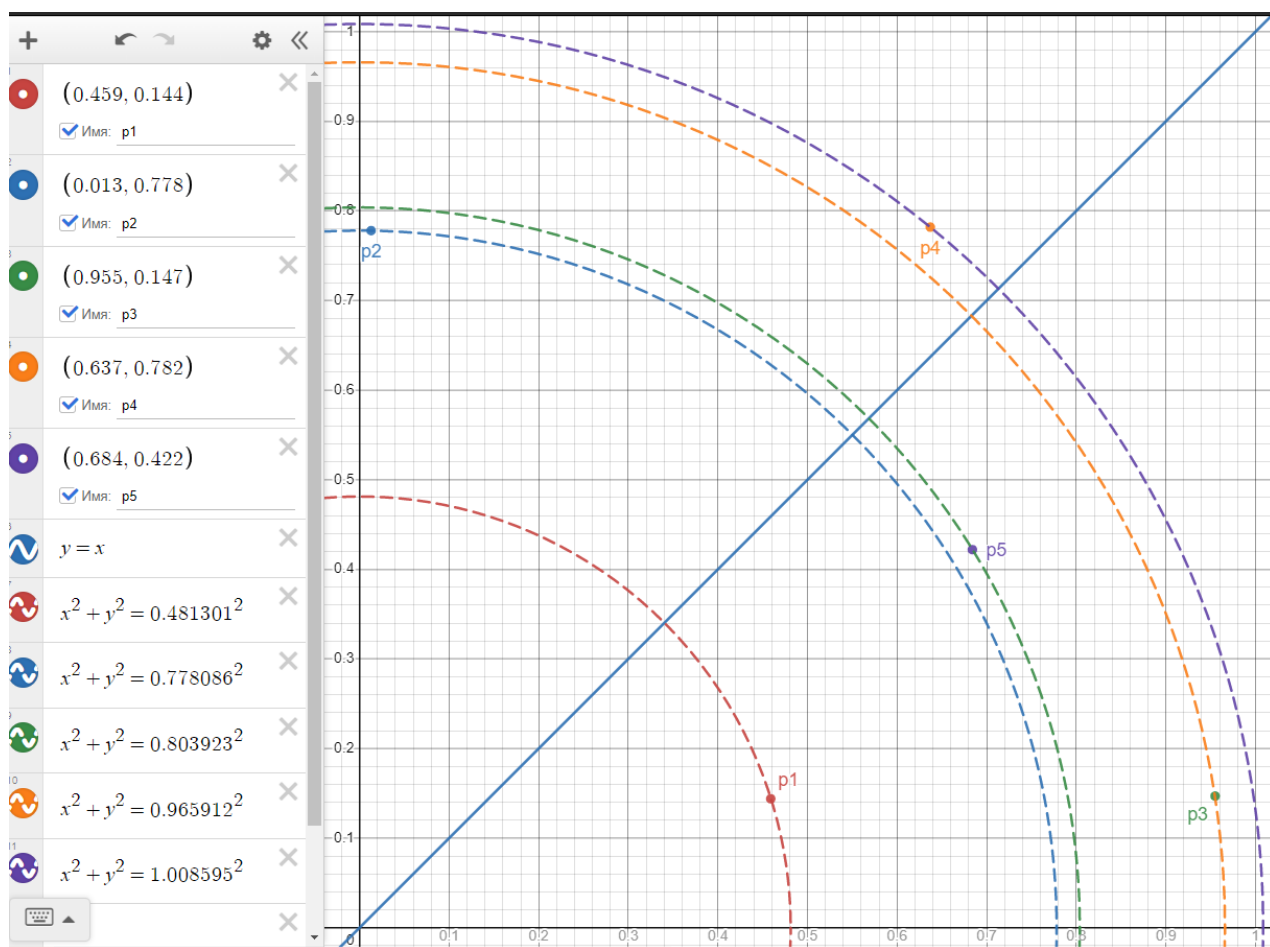


Рисунок 2 – Проверка результата

Вывод: в процессе выполнения лабораторной работы было создано консольное приложение, генерирующее несколько точек в определенной области плоскости и сортирующее эти точки по разным критериям. Классы для сортировки были унаследованы от `Comparable`.