|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 09.03.01 **Информатика и вычислительная техника**

**Отчет**

**по лабораторной работе № 11**

**Дисциплина:** Разработка приложений на языке C#

**Название работы:** Интерфейсы, сортировка с компаратором



Студент гр. ИУ6-72Б **\_\_**16.12.2022**\_\_\_\_\_\_\_** И.С.Марчук

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** А.М. Минитаева

(Подпись, дата) (И.О. Фамилия)

Москва, 2022

**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:

internal class Point

{

public double x;

public double y;

public Point(double x = 0, double y = 0)

{

this.x = x;

this.y = y;

}

public string ToString()

{

return $"x = {x} y = {y}";

}

}

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

internal class DistComp : IComparer<Point>

{

public int Compare(Point p1, Point p2)

{

return (int)((

p1.x \* p1.x + p1.y \* p1.y -

(p2.x\* p2.x + p2.y \* p2.y)) \* 10000);

}

}

internal class DistCompX : IComparer<Point>

{

public int Compare(Point p1, Point p2)

{

return (int)((p1.x - p2.x) \* 10000);

}

}

internal class DistCompY : IComparer<Point>

{

public int Compare(Point p1, Point p2)

{

return (int)((p1.y - p2.y) \* 10000);

}

}

internal class DistCompXY : IComparer<Point>

{

public int Compare(Point p1, Point p2)

{

return (int)((p1.y - p1.x - (p2.y - p2.x)) \* 10000);

}

}

Основная программа:

static void Main(string[] args)

{

Random rand = new Random();

List<Point> points = new List<Point>();

for (int i = 0; i < rand.Next(2, 5); i++)

{

points.Add(new Point(Math.Round( rand.NextDouble(), 4), Math.Round(rand.NextDouble(), 4)));

}

Console.WriteLine(" Заданные точки: \n");

foreach (var point in points)

{

Console.WriteLine(point.ToString());

}

Console.WriteLine();

Console.WriteLine(" От начала координат: \n");

points.Sort(new DistComp());

foreach (var point in points)

{

Console.WriteLine(point.ToString());

}

Console.WriteLine();

Console.WriteLine(" От оси абсцисс: \n");

points.Sort(new DistCompX());

foreach (var point in points)

{

Console.WriteLine(point.ToString());

}

Console.WriteLine();

Console.WriteLine(" От оси ординат: \n");

points.Sort(new DistCompY());

foreach (var point in points)

{

Console.WriteLine(point.ToString());

}

Console.WriteLine();

Console.WriteLine(" От диагонали: \n");

points.Sort(new DistCompXY());

foreach (var point in points)

{

Console.WriteLine(point.ToString());

}

}

Пример выполнения программы приведен на рисунке 1.

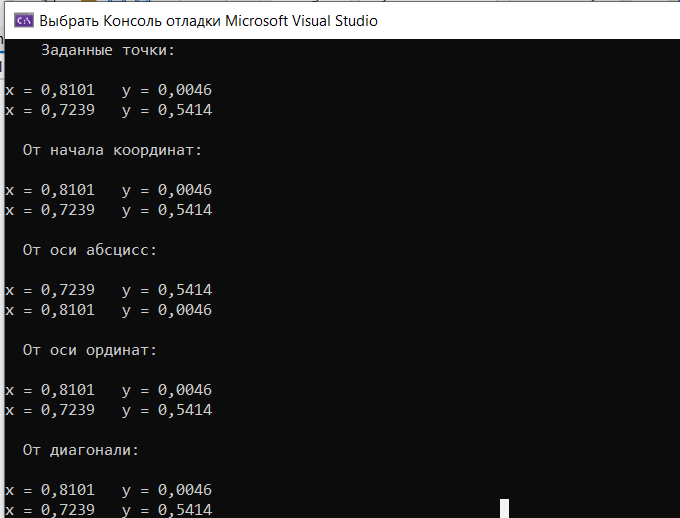


Рисунок 1 – Выполнение программы

**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 IComparable.