

МАГИСТЕРСКАЯ ПРОГРАММА 09.04.01/07 Интеллектуальные системы анализа,
обработки и интерпретации больших данных

Вариант 15

Дисциплина: Языки программирования для работы с большими данными

Преподаватель	<hr/>	П.В. Степанов
	(Подпись, дата)	(И.О. Фамилия)

Москва, 2022

Цель работы:

Получение навыков работы с коллекциями в Java.

Выполнение:

Задание 1:

1. Не используя вспомогательных объектов, переставить отрицательные элементы данного списка в конец, а положительные – в начало этого списка.
2. Ввести строки из файла, записать в список ArrayList. Выполнить сортировку строк, используя метод sort() из класса Collections.

Листинг выполнения подзадачи 1 (файл lr616.java)

```
package lr61;

import java.util.ArrayList;
import java.util.Scanner;

public class lr616 {
    public static void main(String[] args) {
        ArrayList<Integer> my_list = new ArrayList<>();
        Scanner in = new Scanner(System.in);
        for (int i = 0; i < 10; i++) {
            my_list.add(in.nextInt());
        }
        int i = 0;
        int k = 0;
        while (i != 9) {
            k = i;
            if (my_list.get(k) >= 0) {
                while (k > 0 && my_list.get(k - 1) < 0) {
                    int mid = my_list.get(k);
                    my_list.set(k, my_list.get(k - 1));
                    my_list.set(k - 1, mid);
                    k--;
                }
            } else {
                while (k < 9 && my_list.get(k + 1) > 0) {
                    int mid = my_list.get(k);
                    my_list.set(k, my_list.get(k + 1));
                    my_list.set(k + 1, mid);
                    k++;
                }
            }
            if (i == k) {
                i++;
            }
        }
        System.out.println(my_list);
    }
}
```

```
C:\Users\stale\.jdk\openjdk-17.0.2\bin\java.exe "-javaagent:0
1
2
-3
-4
5
6
-7
-8
9
-1
[1, 2, 5, 6, 9, -3, -4, -7, -8, -1]

Process finished with exit code 0
```

Рисунок 1 - Результат выполнения кода решения подзадачи 1

Листинг выполнения подзадачи 2 (файл lr617.java)

```
package lr617;

import java.io.File;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Scanner;

public class lr617 {
    public static void main(String[] args) throws FileNotFoundException {
        File file_in = new
File("C:\\Users\\stale\\IdeaProjects\\java_magister\\lr6\\src\\lr61\\lr617_in.fts");
        Scanner in = new Scanner(file_in);
        ArrayList<String> my_list = new ArrayList<>();
        while (in.hasNextLine()) {
            my_list.add(in.nextLine());
        }
        Collections.sort(my_list);
        System.out.println(my_list);
    }
}
```

```
C:\Users\stale\.jdk\openjdk-17.0.2\bin\jav
[aaa, xx, z, zzz]

Process finished with exit code 0
```

Рисунок 2 - Результат выполнения кода решения подзадачи 2

Задание 2:

1. Во входном файле расположены два набора положительных чисел; между наборами стоит отрицательное число. Построить два списка C1 и C2, элементы которых содержат соответственно числа 1-го и 2-го набора таким образом, чтобы внутри одного списка числа были упорядочены по возрастанию. Затем объединить списки C1 и C2 в один упорядоченный список, изменяя только значения полей ссылочного типа.
2. На плоскости задано N точек. Вывести в файл описания всех прямых, которые проходят более чем через одну точку из заданных. Для каждой прямой указать, через сколько точек она проходит. Использовать класс HashMap.

Содержимое файла для подзадачи 1 (файл lr625_in.fts)

```
5 4 2 1 -7 8 7 6 3
```

Листинг выполнения подзадачи 1 (файл lr625.java)

```
package lr625;

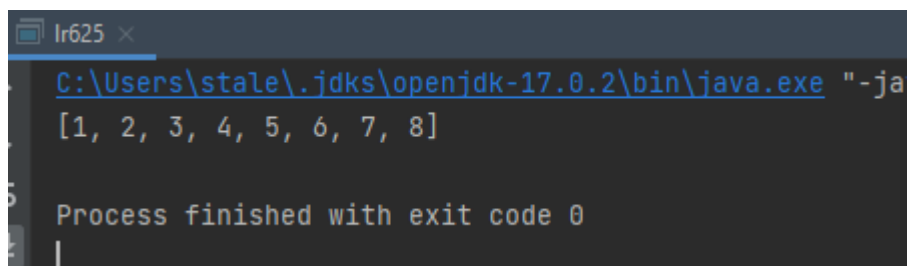
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Collections;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.Scanner;

public class lr625 {
    public static void main(String[] args) throws FileNotFoundException {
        File file_in = new
File("C:\\Users\\stale\\IdeaProjects\\java_magister\\lr6\\src\\lr62\\lr625_in.fts");
        Scanner in = new Scanner(file_in);
        LinkedList<Integer> C1 = new LinkedList<>();
        LinkedList<Integer> C2 = new LinkedList<>();
        int i = in.nextInt();
        while (i > 0){
            C1.add(i);
            i = in.nextInt();
        }
        i = in.nextInt();
        while (in.hasNext()){
            C2.add(i);
            i = in.nextInt();
        }
        C2.add(i);
        Collections.sort(C1);
        Collections.sort(C2);
        LinkedList<Integer> C = new LinkedList<>();
        while (C1.size() != 0 || C2.size() != 0){
            if (C1.size() > 0 && C2.size() > 0){
                if (C1.peekFirst() <= C2.peekFirst()){
                    C.add(C1.pollFirst());
                }else{
                    C.add(C2.pollFirst());
                }
            }else if (C1.size() == 0){
                C.add(C2.pollFirst());
            }
        }
    }
}
```

```

    }else{
        C.add(C1.pollFirst());
    }
}
System.out.println(C);
}
}

```



```

lr625 x
C:\Users\stale\.jdk\openjdk-17.0.2\bin\java.exe "-ja
[1, 2, 3, 4, 5, 6, 7, 8]
Process finished with exit code 0

```

Рисунок 3 - Результат выполнения кода решения подзадачи 1

Исходные данные для выполнения подзадачи 2 (файл lr626_in.fts)

```

1 1
2 2
3 3
4 4
5 1
6 1
7 7

```

Листинг выполнения подзадачи 2 (файл lr626.java)

```

package lr62;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.nio.file.StandardOpenOption;
import java.util.ArrayList;
import java.util.Collections;
import java.util.HashMap;
import java.util.Scanner;

public class lr626 {
    public static void main(String[] args) throws IOException {
        File file_in = new
File("C:\\Users\\stale\\IdeaProjects\\java_magister\\lr6\\src\\lr62\\lr626_in.fts");
        Path path =
Paths.get("C:\\Users\\stale\\IdeaProjects\\java_magister\\lr6\\src\\lr62\\lr626_out.f
ts");

        Scanner in = new Scanner(file_in);
        ArrayList<ArrayList<Integer>> dots = new ArrayList<>();
        HashMap<Float, HashMap<Float, Integer>> lines = new HashMap<>();
        boolean first = true;
        while (in.hasNext()){
            int x = in.nextInt();
            int y = in.nextInt();
            ArrayList<Integer> temp = new ArrayList<>();
            temp.add(x);
            temp.add(y);
            dots.add(temp);
        }
    }
}

```

```

        for (int i = 0; i < dots.size(); i++) {
            Float x1 = Float.valueOf(dots.get(i).get(0));
            Float y1 = Float.valueOf(dots.get(i).get(1));
            for (int j = i + 1; j < dots.size(); j++) {
                Float x2 = Float.valueOf(dots.get(j).get(0));
                Float y2 = Float.valueOf(dots.get(j).get(1));
                Float k = (y2 - y1) / (x2 - x1);
                Float b = y1 - x1 * (y2 - y1) / (x2 - x1);
                if (lines.containsKey(k)) {
                    lines.get(k).put(b, lines.get(k).get(b) + 1);
                } else {
                    HashMap<Float, Integer> temp = new HashMap<>();
                    temp.put(b, 2);
                    lines.put(k, temp);
                }
            }
        }
        for (Float i: lines.keySet()) {
            for (Float j: lines.get(i).keySet()) {
                int ans = lines.get(i).get(j);
                if (ans > 2) {
                    ans = (int) (1 + Math.sqrt(8 * ans - 7)) / 2;
                }
                String str = "Line " + i + "x + " + j + " contains " + ans + " dots
on it";
                if (first) {
                    Files.write(path, Collections.singleton(str),
StandardCharsets.UTF_8);
                    first = false;
                } else {
                    Files.write(path, Collections.singleton(str),
StandardCharsets.UTF_8, StandardOpenOption.APPEND);
                }
            }
        }
    }
}

```

Результат выполнения подзадачи 2 (файл lr626_out.fts)

```

Line 1.0x + 0.0 contains 5 dots on it
Line 0.0x + 1.0 contains 3 dots on it
Line -0.25x + 2.5 contains 2 dots on it
Line -1.0x + 6.0 contains 2 dots on it
Line -3.0x + 16.0 contains 2 dots on it
Line -1.5x + 10.0 contains 2 dots on it
Line 3.0x + -14.0 contains 2 dots on it
Line 6.0x + -35.0 contains 2 dots on it
Line -0.33333334x + 2.6666667 contains 2 dots on it
Line -0.6666667x + 5.0 contains 2 dots on it

```

Ссылка на программное решение:

Программное решение представлено в репозитории распределённой системы управления версиями Git:

https://github.com/stalekc/java_magister/tree/main/lr6/src

Вывод:

При выполнении лабораторной работы были получены навыки работы с коллекциями в Java.