|  |  |
| --- | --- |
| **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 **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/12 Интеллектуальный анализ больших**

**данных в системах поддержки принятия решений**

**Отчет**

**по лабораторной работе № 6**

**Название:** Коллекции

**Дисциплина:** Языки программирования для работы с большими данными

Студент гр. ИУ6-23М **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_** Г. Е. Горский **\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_**П.В. Степанов **\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2023

**Цель:** ознакомиться с базовыми принципами языка Java для работы с большими данными

**Вариант 1:**

1. Ввести строки из файла, записать в список ArrayList. Выполнить сортировку строк, используя метод sort() из класса Collections.
2. Задана строка, состоящая из символов '(', ')', '[', ']', '{', '}'. Проверить правильность расстановки скобок. Использовать стек.

**Решение:**

import java.io.File;  
import java.io.FileNotFoundException;  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.Scanner;  
/\*  
7. Ввести строки из файла, записать в список ArrayList.  
 Выполнить сортировку строк, используя метод sort() из класса Collections.  
 \*/  
public class Main {  
 public static void main(String[] args) {  
 System.out.println("Hello world!");  
 ArrayList<String> arrayList = new ArrayList<>();  
 try {  
 File myObj = new File("/Users/lol/Desktop/untitled/Lab6/Lab6.1.1/src/1.txt");  
 Scanner myReader = new Scanner(myObj);  
 while (myReader.hasNextLine()) {  
 arrayList.add(myReader.nextLine());  
 }  
 myReader.close();  
 } catch (FileNotFoundException e) {  
 System.out.println("An error occurred.");  
 e.printStackTrace();  
 }  
  
 System.out.println(arrayList);  
 Collections.sort(arrayList);  
 System.out.println(arrayList);  
 }  
}

Результат выполнения представлен на рисунке 1.

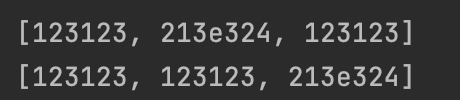
****

Рисунок 1 — пример выполнения

import java.util.Stack;  
public class Main {  
 public static void main(String[] args) {  
 System.out.println("Hello world!");  
 String str = "(', ')[', ']{', '}";  
 Stack<Character> stack = new Stack<>();  
 for(int i=0; i<=str.length()-1;i++){  
 stack.push(str.charAt(i));  
 //System.out.println(str.charAt(i));  
 }  
 // System.out.println(stack);  
 int a=-1;  
 for (int i=0; i<stack.size()-5; i+=6){  
 a=-1;  
 if(stack.get(i)=='(' || stack.get(i)=='{' || stack.get(i)=='['){  
 if(stack.get(i)=='(' && stack.elementAt(i+5)==')'){  
 System.out.println("correct" + stack.get(i) + " " + stack.elementAt(i+5));  
 a=1;  
 continue;  
 }  
 if(stack.get(i)=='{' && stack.elementAt(i+5)=='}'){  
 System.out.println("correct" + stack.get(i) + " " + stack.elementAt(i+5));  
 a=2;  
 continue;  
 }  
 if(stack.get(i)=='[' && stack.elementAt(i+5)==']'){  
 System.out.println("correct" + stack.get(i) + " " + stack.elementAt(i+5));  
 a=3;  
 }  
 if(a==-1){  
 System.out.println("incorrect");  
 }  
 }  
  
 }  
 }  
}

Результат выполнения представлен на рисунке 2.

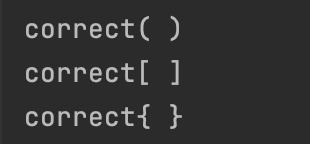


Рисунок 2 — пример выполнения

**Вариант 2:**

1. На плоскости задано N отрезков. Найти точку пересечения двх отрезков, имеющую минимальную абсциссу. Использовать класс TreeMap.
2. На клетчатом листе бумаги закрашена часть клеток. Выделить все различные фигуры, которые образовались при этом. Фигурой считается набор закрашенных клеток, достижимых друг из друга при движении в четырёх направлениях. Две фигуры являются различными, если их нельзя совместить поворотом на угол, кратный 90 градусам, и параллельным переносом. Используйте класс HashSet.

**Решение:**

public class Main {  
 public static void main(String[] args) {  
 System.out.println("Hello world!");  
 float k1,k2,b1,b2;  
 int temp=0;  
 TreeMap<Integer, Integer> firstOtrezok = new TreeMap<>();  
 TreeMap<Integer, Integer> secondOtrezok = new TreeMap<>();  
 TreeMap<Integer,Float> peresechenie = new TreeMap<>();  
 Random rnd = new Random();  
 firstOtrezok.put(0,rnd.nextInt(1,5)); //x1  
 firstOtrezok.put(1,rnd.nextInt(1,5)); //y1  
 firstOtrezok.put(2,rnd.nextInt(1,5)); //x2  
 firstOtrezok.put(3,rnd.nextInt(1,5)); //y2  
 secondOtrezok.put(0,rnd.nextInt(1,5)); //x3  
 secondOtrezok.put(1,rnd.nextInt(1,5)); //y3  
 secondOtrezok.put(2,rnd.nextInt(1,5)); //x4  
 secondOtrezok.put(3,rnd.nextInt(1,5)); //y4  
 System.out.println(firstOtrezok + " "+secondOtrezok);  
 if (firstOtrezok.get(0)>= firstOtrezok.get(2)){  
 int tempX=firstOtrezok.get(2);  
 int tempY=firstOtrezok.get(3);  
 firstOtrezok.put(2,firstOtrezok.get(0));  
 firstOtrezok.put(0, tempX);  
 firstOtrezok.put(3, firstOtrezok.get(1));  
 firstOtrezok.put(1, tempY);  
 }  
 if (secondOtrezok.get(0)>= secondOtrezok.get(2)){  
 int tempX=secondOtrezok.get(2);  
 int tempY=secondOtrezok.get(3);  
 secondOtrezok.put(2,secondOtrezok.get(0));  
 secondOtrezok.put(0, tempX);  
 secondOtrezok.put(3, secondOtrezok.get(1));  
 secondOtrezok.put(1, tempY);  
 }  
 System.out.println(firstOtrezok + " "+secondOtrezok);  
 if (firstOtrezok.get(1).equals(firstOtrezok.get(3))){  
 k1=0;  
 }else {  
 //k1 = ( у2 - у1 ) / ( x2 - x1 )  
 k1 = (float)(firstOtrezok.get(3)-firstOtrezok.get(1)) / (firstOtrezok.get(2)- firstOtrezok.get(0));  
 }  
 if (secondOtrezok.get(1).equals(secondOtrezok.get(3))){  
 k2=0;  
 }else {  
 //k2 = ( у4 - у3 ) / ( x4 - x3 )  
 k2 = (float)(secondOtrezok.get(3)-secondOtrezok.get(1)) / (secondOtrezok.get(2)- secondOtrezok.get(0));  
 }  
 if(k1==k2){  
 System.out.println("отрезки не пересекаются");  
 return;  
 }  
 b1 = firstOtrezok.get(1) - k1\* firstOtrezok.get(0);  
 b2 = secondOtrezok.get(1) - k1\* secondOtrezok.get(0);  
 if(firstOtrezok.get(2)>secondOtrezok.get(2)){  
 temp=firstOtrezok.get(2);  
 }else temp=secondOtrezok.get(2);  
 for (float i=0; i<temp; i+=0.0001){  
 //k1 x + b1 = k2 x + b2  
 if ((k1\*i+b1)==(k2\*i+b2)){  
 //System.out.println(i);  
 peresechenie.put(0,i); //x  
 peresechenie.put(1,k1\*i+b1); //y  
 }  
 }  
 if(peresechenie.isEmpty()){  
 System.out.println("пересечения нет или недостаточная точность поиска по Х");  
 }else {  
 System.out.println("x=" + peresechenie.firstEntry().getValue() + " y="+ peresechenie.get(peresechenie.firstEntry().getKey()+1));  
 }  
 }  
}

Результат выполнения представлен на рисунке 3.

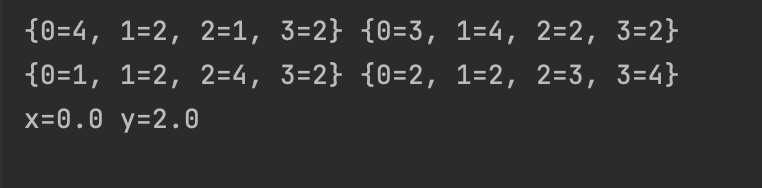


Рисунок 3 — пример выполнения

public class Main {  
 public static void main(String[] args) {  
 HashSet<String> set = new HashSet<>();  
 int[][] matrix;  
 ArrayList<String> pos = new ArrayList<>();  
 matrix= new int[][]{  
 {1, 1, 1, 0, 0},  
 {1, 0, 0, 1, 1},  
 {0, 0, 1, 0, 0},  
 {0, 0, 1, 0, 0}};  
 for(int i=0;i<4; i++){  
 for (int j=0; j<5; j++){  
 System.out.print(matrix[i][j]);  
 }  
 System.out.println();  
 }  
 int[][] mat1;  
 int countX=0;  
 int countY=0;  
 for(int i=0;i<3; i++){  
 countX=0;  
 countY=0;  
 for (int j=0; j<4; j++){  
 if(matrix[i][j]==1){  
 if(matrix[i][j+1]==1){  
 countX++;  
 }  
 if(matrix[i+1][j]==1){  
 countY++;  
 }  
 }  
 }  
 countX++;  
 countY++;  
 pos.add("X"+countX+"Y"+countY);  
 }  
 for (int i=0;i< pos.size(); i++){  
 set.add(pos.get(i));  
 }  
 System.out.println(set);  
 }  
}

Результат выполнения представлен на рисунке 4.

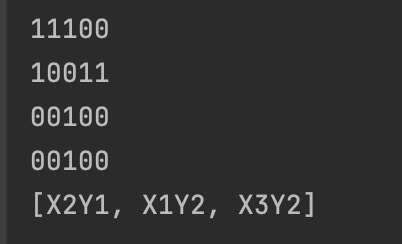


Рисунок 4 — пример выполнения

**Вывод:** были разработаны классы и методы согласно вариантам.