Міністерство освіти і науки України

Івано-Франківський національний технічний університет нафти і газу

*Кафедра програмного забезпечення автоматизованих систем*

**Лабораторна робота №7**

Тема:

**«Виключенн грубих помилок.**

**Критерій Ірвіна»**

***Виконав*:**

ст. групи ПіТ-15-3

Турчин М.М.

Свирид О.Б.

***Перевірив:***

Незамай Б.С.

Івано-Франківськ

2015

**Завдання:** Розробити програму, яка генерує текстовий файл з даними відповідно до варіанту; програму для аналізу даних. Робота з файлом повинна бути винесена в окремий клас. Передбачити коректну обробку помилок вводу/виводу для роботи з файлом даних.

**Хід роботи**

Обсяг вибірки N номер варіанту у списку старости.

Код програми на Java:

**FileUtil.java**

**package** EMP.Lab\_7;  
  
**import** java.io.\*;  
**import** java.util.logging.Logger;  
  
*/\*\**

*\* Class for leading of file*

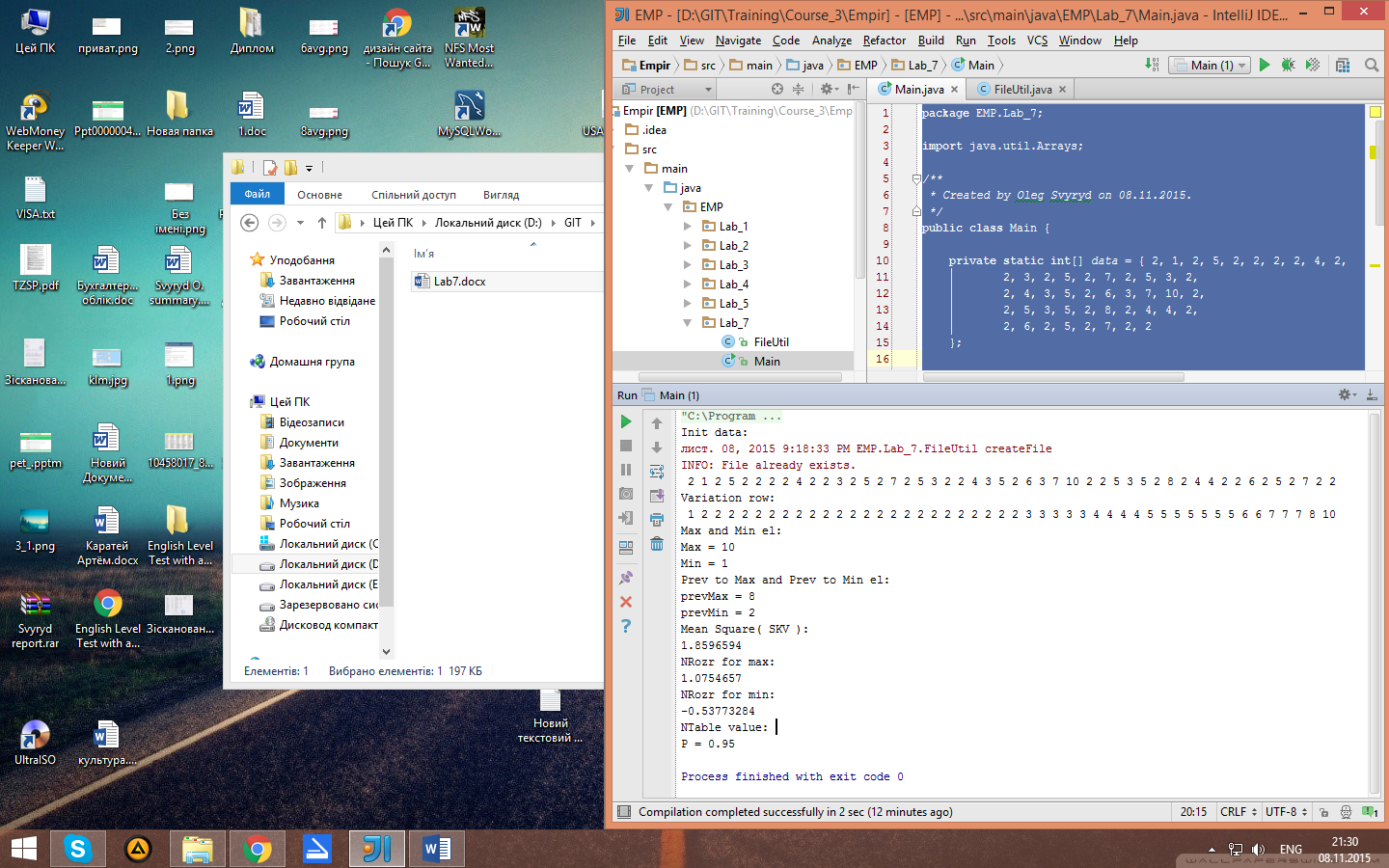
*\*/***public class** FileUtil {  
 **private static final** Logger LOG = Logger.getAnonymousLogger();  
  
 **private static final int *N*** = 48;  
 **private static** File *actual*;  
  
 **private static** File createFile(String name) {  
 *actual* = **new** File(name);  
  
 **try** {  
 **if** (*actual*.createNewFile()){  
 ***LOG***.info(**"File is created!"**);  
 }**else**{  
 ***LOG***.info(**"File already exists."**);  
 }  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
  
 **return** *actual*;  
 }  
  
 **public void** fillFile(**int**[] data) {  
 *actual* = FileUtil.*createFile*(**"test.txt"**);  
  
 **try**(BufferedWriter bufferedWriter = **new** BufferedWriter(**new** FileWriter(*actual*))) {  
 **for**(**int** i = 0; i < data.**length**; i++) {  
 bufferedWriter.write(*pushRow*(data[i]));  
 bufferedWriter.newLine();  
 }  
 } **catch** (IOException e) {  
 ***LOG***.warning(e.getMessage());  
 }  
 }  
  
 **public int**[] getData() {  
 **int**[] data = **new int**[N];  
  
 **try**(BufferedReader bufferedReader = **new** BufferedReader(**new** FileReader(actual))){  
 String strLine;  
 **int** count = 0;  
 **while** ((strLine = bufferedReader.readLine()) != **null**) {  
 data[count++] = getRow(strLine);  
 }  
 } **catch** (FileNotFoundException e) {  
 LOG.warning(e.getMessage());  
 } **catch** (IOException e) {  
 LOG.warning(e.getMessage());  
 } **catch** (NumberFormatException e) {  
 LOG.warning(e.getMessage());  
 }  
  
 **return** data;  
 }  
  
 **private static** String pushRow(**int** digit) {  
 **return** Integer.valueOf(digit).toString();  
 }  
 **private static int** getRow(String strLine) {  
 **return** Integer.parseInt(strLine);  
 }  
}

**Main.java:**

**package** EMP.Lab\_7;  
  
**import** java.util.Arrays;  
  
*/\*\*  
 \* Main class for calculating*

*\*/***public class** Main {  
  
 **private static int**[] *data* = { 2, 1, 2, 5, 2, 2, 2, 2, 4, 2,  
 2, 3, 2, 5, 2, 7, 2, 5, 3, 2,  
 2, 4, 3, 5, 2, 6, 3, 7, 10, 2,  
 2, 5, 3, 5, 2, 8, 2, 4, 4, 2,  
 2, 6, 2, 5, 2, 7, 2, 2  
 };  
  
 */\*\*  
 \* Return average of array  
 \** ***@param array*** *Init array of digits  
 \** ***@return*** *average  
 \*/* **protected static float** getAverage(**int**[] array) {  
 **float** sum = 0;  
 **for**(**int** i : array) {  
 sum += i;  
 }  
 **return** sum / array.**length**;  
 }  
  
 */\*\*  
 \* Calculate mean square  
 \** ***@param array*** *Array of numbers  
 \** ***@return*** *Mean square of array  
 \*/* **protected static float** getMeanSquare(**int**[] array) {  
 **return** (**float**) Math.*sqrt*(Main.*getAverage*(array));  
 }  
  
 */\*\*  
 \* Get variation row  
 \** ***@param array*** *\*/* **protected static void** sortArrayByAsc(**int**[] array){  
 Arrays.*sort*(array);  
 }  
  
**private static float** getP(**float** nRozr) {  
 **float** n1 = (**float**) (2.0f \* Math.*pow*(48, -0.5) + 0.6f);  
 **float** n3 = (**float**) (3.0f \* Math.*pow*(48, -0.5) + 1.15f);  
  
 **if**( nRozr < n1 ) {  
 **return** 0.9f;  
 }  
 **else if** ( (nRozr > n1 ) && (nRozr <= n3) ) {  
 **return** 0.95f;  
 }  
 **else** {  
 **return** 0.99f;  
 }  
 }  
  
 **public static void** main(String[] args) {  
 FileUtil fileUtil = **new** FileUtil();  
  
 System.***out***.println(**"Init data: "**);  
 fileUtil.fillFile(*data*);  
  
 **int**[] arr = fileUtil.getData();  
 *displayArray*(arr);  
  
 System.***out***.println(**"\nVariation row: "**);  
 *sortArrayByAsc*(arr);  
 *displayArray*(arr);  
  
 **int** max = arr[arr.**length** - 1];  
 **int** min = arr[0];  
 System.***out***.println(**"\nMax and Min el: "**);  
 System.***out***.println(**"Max = "** + max + **"\nMin = "** + min);  
  
 **int** prevMax = arr[arr.**length** - 2];  
 **int** prevMin = arr[1];  
 System.***out***.println(**"Prev to Max and Prev to Min el: "**);  
 System.***out***.println(**"prevMax = "** + prevMax + **"\nprevMin = "** + prevMin);  
  
 System.***out***.println(**"Mean Square( SKV ): "**);  
 **float** meanSquare = *getMeanSquare*(arr);  
 System.***out***.println(meanSquare);  
  
 **float** nRozr = (max - prevMax) / meanSquare;  
 **float** nRozr1 = (min - prevMin) / meanSquare;  
  
 System.***out***.println(**"NRozr for max: "**);  
 System.***out***.println(nRozr);  
 System.***out***.println(**"NRozr for min: "**);  
 System.***out***.println(nRozr1);  
 System.***out***.println(**"NTable value: "**);  
  
 System.***out***.println(**"P = "** + *getP*(nRozr));  
 }  
  
 **private static void** displayArray(**int**[] arr) {  
 **for**(**int** i : arr) {  
 System.***out***.print(**" "** + i);  
 }  
 }  
}

Результат виконання програми:



**Висновок:** в ході лабораторної роботи я реалізував програму на мові Java яка генерує текстовий файл з даними відповідно до варіанту; та програму для аналізу даних за певним завданням.