Кафедра комп’ютерних систем та мереж

**ЛАБОРАТОРНА РОБОТА №1**

Тема: «Найпростіша статична обробка емпіричних даних»

*Виконав студент*

*групи Піт-15-3:*

***Свирид О.Б.***

*Перевірив:*

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

**Мета роботи:** ознайомитися з принципом написання програм щодо статистичної обробки емпіричних даних, зображення алгоритму побудови варіаційного ряду.

**Хід роботи:**

Схема алгоритму сортування:

Ні

for(int j = 0; j< i; j++)

Так

Так

Початок

Ввід length

Ввід масиву А

for(int i = length-1; i <0; i--)

Array[j] > Array[j + 1]

tmp = A[j];

A[j] = A[j+1];

A[j+1] = tmp;

Так

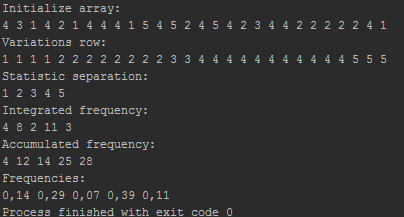
Кінець

Програмний код:

package EMP.Lab\_1;  
  
import java.util.Arrays;  
import java.util.Random;  
  
public class Calculate {  
  
 //variant + 10  
 private static final int SIZE = 28;  
 private static final int *MAX\_VALUE* = 5;  
  
 private int[] array = new int[SIZE];  
  
 protected void initializeArray(){  
 for(int i = 0; i < SIZE; i++) {  
 array[i] = new Random().nextInt(*MAX\_VALUE*) + 1;  
 System.*out*.print(array[i] + " ");  
 }  
 }  
  
 private void sortArrayByAsc(){  
 Arrays.*sort*(array);  
 }  
  
 protected void getStatisticRow(){  
 sortArrayByAsc();  
  
 for (int i = 0; i < SIZE; i++)  
 if ( i == 0 || array[i] != array[i - 1] )  
 System.*out*.print(array[i] + " ");  
 }  
  
 protected void getIntegratedFrequency(){  
 sortArrayByAsc();  
  
 int freq = 0;  
 for (int i = 0; i < SIZE; i++) {  
 freq++;  
  
 if(i == 0) { freq = 0; continue; }  
  
 if(i == (SIZE - 1))  
 System.*out*.println(freq+1);  
  
 if(array[i] != array[i - 1]) {  
 System.*out*.print(freq + " ");  
 freq = 0;  
 }  
 }  
 }  
  
 protected void getAccumulatitedFrequency(){  
 sortArrayByAsc();  
  
 int acumFreq = 0;  
 for(int i = 0; i < SIZE; i++){  
 acumFreq++;  
  
 if(i == 0){  
 acumFreq = 0;  
 continue;  
 }  
  
 if(i == (SIZE - 1))  
 System.*out*.println(acumFreq + 1);  
  
 if(array[i] != array[i - 1]) {  
 System.*out*.print(acumFreq + " ");  
 }  
 }  
 }  
  
 protected void getFrequencies(){  
 sortArrayByAsc();  
  
 float freq = 0;  
 for (int i = 0; i < SIZE; i++) {  
 freq++;  
  
 if(i == 0) { freq = 0; continue; }  
  
 if(i == (SIZE - 1))  
 System.*out*.format("%2.2f ", (freq + 1) / SIZE);  
  
 if(array[i] != array[i - 1]) {  
 System.*out*.format("%2.2f ", freq / SIZE);  
 freq = 0;  
 }  
 }  
 }  
  
 protected void sort(){  
 System.*out*.println("\nVariations row: ");  
 sortArrayByAsc();  
 for(int i: array)  
 System.*out*.print(i + " ");  
 }  
};

package EMP.Lab\_1;  
  
public class Lab1 {  
 public static void main(String[] args) {  
 Calculate calculate = new Calculate();  
  
 System.*out*.println("Initialize array: ");  
 calculate.initializeArray();  
  
 calculate.sort();  
  
 System.*out*.println("\nStatistic separation: ");  
 calculate.getStatisticRow();  
  
 System.*out*.println("\nIntegrated frequency: ");  
 calculate.getIntegratedFrequency();  
  
 System.*out*.println("Accumulated frequency: ");  
 calculate.getAccumulatitedFrequency();  
  
 System.*out*.println("Frequencies: ");  
 calculate.getFrequencies();  
 }  
}

Результат виконання програми:



**Висновок**: на даній лабораторній роботі було розроблено програму, яка генерує випадкову послідовність даних з 28 елементів, які набувають значень із набору (1,2,3,4,5), виводити вихідні дані, варіаційний ряд, статистичний розподіл, інтегральну частоту та частість.