Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра автоматики та управління в технічних системах

**Лабораторна робота № 2**

по дисципліні «СТП-1»

|  |  |
| --- | --- |
| Виконав:  студент групи ІТ-51  Давиденко Ігор Володимирович  Дата здачі \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Захищено з балом \_\_\_\_\_\_\_\_\_\_\_ | Перевірено:  ac. Федорчук В. В |

Київ 2017

Варіант 2

Завдання

1. Write a program that asks the user the number of array elements, then in loop the user enter values for each item.

Find the number of paired items;   
Find the sum of the elements multiplied by 3;   
Find the difference between the maximum and the minimum elements array;   
Find the arithmetic mean of the array;   
Find the sum of the largest and smallest elements of the array;   
Find the maximum by module element of the array

2. Output parameters such as the array dimension are entered from the keyboard, the contents of the array generate with Random#nextInt. After the user inputs the dimension to generate the matrix, and output take it to the screen, then complete the task and display the result.

The given positive integer n is an integer square matrix of order n. Get bi, ..., bn, where bi is:

a) the sum of the elements located behind the first negative element in i-th line (if all elements of the line are non-negative, take bi = 100);

b) the sum of the elements preceding the last negative element i-th line (if all elements of the line are non-negative, then take bi = -l);   
An integer matrix of order n is given. Find line numbers:

a) all elements of which are zeros;

b) the elements in each of them are the same.

Контрольні питання

1. Possible theoretical questions (questions will not be limited to this list):

* How to find out the size of the array
* How to find out the size of a multidimensional array
* How to display the contents of the array on the screen
* What is an array?
* Based on which two-dimensional arrays are constructed
* What is a pseudo-random number generator.

Короткі теоретичні відомості

Генератор псевдовипадкових чисел

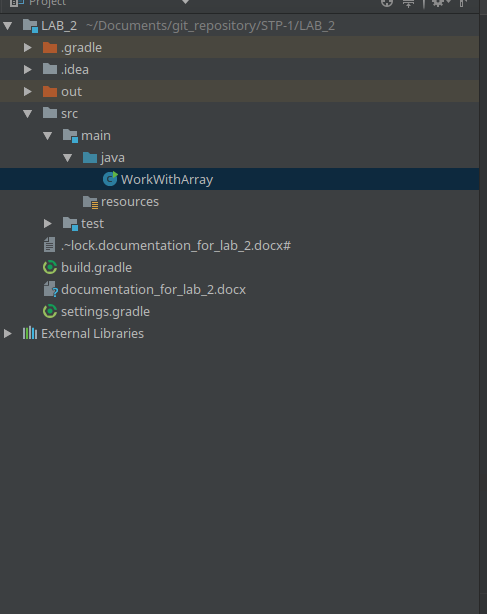
Можна створити таку послідовність чисел, властивості якої будуть схожі на властивості послідовності випадкових чисел. Такі послідовності називаються псевдовипадковими.

Вперше запропонував їх використовувати Джон фон Нейман у 1946 р. Його метод полягав в наступному: n-розрядне число підносилось до квадрата і з нього вибиралися середні n цифр. Метод був дуже недосконалий, послідовності майже завжди вироджувалися в нуль або зациклювалися з коротким періодом. Пізніше було запропоновано багато різних алгоритмів отримання псевдовипадкових чисел.

В основі програмних генераторів як правило лежать рекурентні формули. Як правило, вони генерують цілі числа рівномірно розподілені на відрізку від 0, до деякого максимального m. Щоб отримати числа з плаваючою комою, рівномірно розподілені на [0,1), кожен отриманий результат ділять на m.

Xід роботи

1. Створю проект “LAB\_2”, котрий має наступну структуру:



2. Додам в проект файл “WorkWithArray.java”, і введу наступний код:

import jdk.internal.util.xml.impl.Input;

import java.util.\*;

public class WorkWithArray {

public static void main(String []args)

{

Scanner inputText = new Scanner(System.in);

int size = 0;

sizeStart : do {

System.out.printf("\n Введите размер масива: \n");

try{

size = inputText.nextInt();

} catch (Exception e){

System.out.printf("\n Спробуй еще раз....)");

inputText = new Scanner(System.in);

continue sizeStart;

}

}while (size == 0);

int [] arrayNumbers = new int [size];

int index = 0;

start: do {

System.out.printf("\n Введите %s елемент масива: ", index);

try{

arrayNumbers[index] = inputText.nextInt();

} catch (Exception e){

System.out.printf("\n Спробуй еще раз....)");

inputText = new Scanner(System.in);

continue start;

}

index++;

}while(index != size);

System.out.printf("TASK 1 \n");

System.out.printf("Paired number : %s \n", pairedNumber2(arrayNumbers));

System.out.printf("Sum of the elements multiplied by 3 : %s \n", SumMultipliedElements(arrayNumbers));

System.out.printf("Difference between the maximum and the minimum elements array : %s \n", DifferenceMaxMin(arrayNumbers));

System.out.printf("Arithmetic mean of the array : %s \n", ArithmeticMeanArray(arrayNumbers));

System.out.printf("Sum of the largest and smallest elements of the array : %s \n", SumLargestAndSmallestElem(arrayNumbers));

System.out.printf("Maximum by module element of the array: %s \n", maximumModuleElem(arrayNumbers));

Arrays.sort(arrayNumbers);

System.out.printf("Array : %s", Arrays.toString(arrayNumbers));

size = 0;

sizeStart : do {

System.out.printf("\n Введите размер квадратного масива: \n");

try{

size = inputText.nextInt();

} catch (Exception e){

System.out.printf("\n Спробуй еще раз....)");

inputText = new Scanner(System.in);

continue sizeStart;

}

}while (size == 0);

int [][] sqArrayNumbers = generateSquareArray(size);

System.out.printf("\n TASK 2 \n");

WorkWithArray.showSquareArray(sqArrayNumbers);

System.out.printf("The sum of the elements located behind the first negative element in i-th line : %s \n", getSumBeforeNegativeEl(sqArrayNumbers).toString());

System.out.printf("The sum of the elements preceding the last negative element i-th line : %s \n", getSumAfterNegativeEl(sqArrayNumbers).toString());

System.out.printf("All elements of which are zeros: %s \n", getNumberRowNullElem(sqArrayNumbers).toString());

System.out.printf("The elements in each of them are the same: %s \n", getNumberRowSameElem(sqArrayNumbers).toString());

}

public static int[] generateData(int sizeArray) // метод которий генерирует даные

{

Random randomNumber = new Random();

int [] randomNumbers = new int [sizeArray];

// int from = -10;

// int to = 100;

for (int i = 0; i < sizeArray; i++) {

//randomNumbers[i] = from + (int) (Math.random() \* to);

randomNumbers[i] = randomNumber.nextInt();

}

return randomNumbers;

}

public static int pairedNumber(int [] randomNumbers){ // метод которий повертает парние числа в масиве

if (randomNumbers.length > 0){

int pairedNumbers = 0;

Arrays.sort(randomNumbers);

int arrayLength = randomNumbers.length;

for (int i = 0; i < arrayLength-1; i+=2)

if (randomNumbers[i] == randomNumbers[i+1])

pairedNumbers++;

return pairedNumbers;

}

else

return 0;

}

public static int pairedNumber2(int [] randomNumbers){ // метод которий повертает парние числа в масиве

if (randomNumbers.length > 0){

int pairedNumbers = 0;

Arrays.sort(randomNumbers);

int arrayLength = randomNumbers.length;

for (int i = 0; i < arrayLength-1; i++)

for(int j = i+1; j < arrayLength; j++ )

if (randomNumbers[i] == randomNumbers[j]){

pairedNumbers++;

break;

}

return pairedNumbers;

}

else

return 0;

}

public static int SumMultipliedElements(int [] randomNumbers){ // сума елементов умноженых на три

if (randomNumbers.length > 0){

int sumMultipliedEl = 0;

int lengthArray = randomNumbers.length;

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

if (randomNumbers[i] % 3 == 0)

sumMultipliedEl += randomNumbers[i];

return sumMultipliedEl;

}

else

return 0;

}

public static int DifferenceMaxMin(int[] randomNumber){ // разница межу максимальним и минимальним елементом

if (randomNumber.length > 0){

Arrays.sort(randomNumber);

int maxElemrnt = randomNumber[randomNumber.length-1];

int minElement = randomNumber[0];

return maxElemrnt - minElement;

}

else

return 0;

}

public static float ArithmeticMeanArray(int[] randomNumbers){ // середне ариметичне масива

if (randomNumbers.length > 0){

int sumElements = 0;

int numElements = randomNumbers.length;

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

sumElements += randomNumbers[i];

return sumElements / numElements;

}

else

return 0;

}

public static int SumLargestAndSmallestElem(int[] randomNumbers){ // сума самых больших и наименьших элементов массива

if (randomNumbers.length > 0){

Arrays.sort(randomNumbers);

int sumMaxElements = 0;

int sumMinElements = 0;

int lengthArray = randomNumbers.length;

int maxElem = randomNumbers[lengthArray-1];

int minElem = randomNumbers[0];

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

if (randomNumbers[i] == minElem)

sumMinElements += randomNumbers[i];

else

break;

for (int j = lengthArray - 1; j > 0; j--)

if(randomNumbers[j] == maxElem)

sumMaxElements += randomNumbers[j];

else

break;

return sumMinElements + sumMaxElements;

}

else

return 0;

}

public static int maximumModuleElem(int[] randomNumbers){ // максимальный элемент модуля массива

if(randomNumbers.length > 0){

int lengthArray = randomNumbers.length;

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

randomNumbers[i] = Math.abs(randomNumbers[i]);

Arrays.sort(randomNumbers);

return randomNumbers[lengthArray - 1];

}

else

return 0;

}

public static int[][] generateSquareArray(int size){ // генерация квадратного масива

Random randomNumber = new Random();

// int from = -10;

// int to = 100;

int [][] randomNumbers = new int [size][size];

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

for (int j = 0; j < size; j++) {

//randomNumbers[i][j] = from + (int) (Math.random() \* to);

randomNumbers[i][j] = randomNumber.nextInt();

}

return randomNumbers;

}

public static List<String> getSumBeforeNegativeEl(int [][] randomNumbers){ // сумма элементов, расположенных за первым отрицательным элементом в i-й строке (если все элементы линии неотрицательны, возьмите bi = 100);

int row = randomNumbers.length;

int col = randomNumbers[0].length;

int positionNegativeEl = 0;

int sumElem = 0;

List<String> SumBeforeNegativeEl = new ArrayList<String>();

boolean isNegativeEl = false;

for (int i = 0; i < row; i++) {

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

if (randomNumbers[i][j] < 0) {

positionNegativeEl = j;

isNegativeEl = true;

break;

}

if (isNegativeEl) {

for (int k = 0; k < positionNegativeEl; k++)

sumElem += randomNumbers[i][k];

SumBeforeNegativeEl.add(sumElem + " ");

sumElem = 0;

isNegativeEl = false;

positionNegativeEl = 0;

} else {

SumBeforeNegativeEl.add(100 + " ");

positionNegativeEl = 0;

}

}

return SumBeforeNegativeEl;

}

public static List<String> getSumAfterNegativeEl(int [][] randomNumbers){ // б) сумма элементов, предшествующих последнему отрицательному элементу i-й линии (если все элементы линии неотрицательны, то возьмем bi = -l);

int row = randomNumbers.length;

int col = randomNumbers[0].length;

int positionNegativeEl = 0;

int sumElem = 0;

List<String> SumAfterNegativeEl = new ArrayList<String>();

boolean isNegativeEl = false;

for (int i = 0; i < row; i++) {

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

if (randomNumbers[i][j] < 0) {

positionNegativeEl = j;

isNegativeEl = true;

break;

}

if (isNegativeEl) {

for (int k = positionNegativeEl; k < col; k++)

sumElem += randomNumbers[i][k];

SumAfterNegativeEl.add(sumElem + " ");

sumElem = 0;

isNegativeEl = false;

positionNegativeEl = 0;

} else {

SumAfterNegativeEl.add(-1 + " ");

positionNegativeEl = 0;

}

}

return SumAfterNegativeEl;

}

public static List<String> getNumberRowNullElem(int [][] randomNumbers){ // все элементы которых являются нулями

int row = randomNumbers.length;

int col = randomNumbers[0].length;

List<String> zeroRow = new ArrayList<String>();

boolean isNull = false;

for (int i = 0; i < row; i++) {

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

if (randomNumbers[i][j] != 0) {

isNull = false;

break;

} else

isNull = true;

if(isNull){

zeroRow.add(i + "\t");

isNull = false;

}

}

return zeroRow;

}

public static List<String> getNumberRowSameElem(int [][] randomNumbers){ // елементи есть одинаковие

int row = randomNumbers.length;

int col = randomNumbers[0].length;

List<String> sameRow = new ArrayList<String>();

boolean isSame = false;

for (int i = 0; i < row; i++) {

second: for (int j = 0; j < col-1; j++)

for(int k = j+1; k < col; k++)

if (randomNumbers[i][j] != randomNumbers[i][k])

isSame = false;

else{

isSame = true;

break second;

}

if(isSame) {

sameRow.add(i + "\t");

isSame = false;

}

}

return sameRow;

}

public static void showSquareArray(int [][] randomNumbers)

{

int row = randomNumbers.length;

int col = randomNumbers[0].length;

for (int i = 0; i < row; i++){

System.out.printf("row "+i+" : ");

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

System.out.printf(randomNumbers[i][j] + " ");

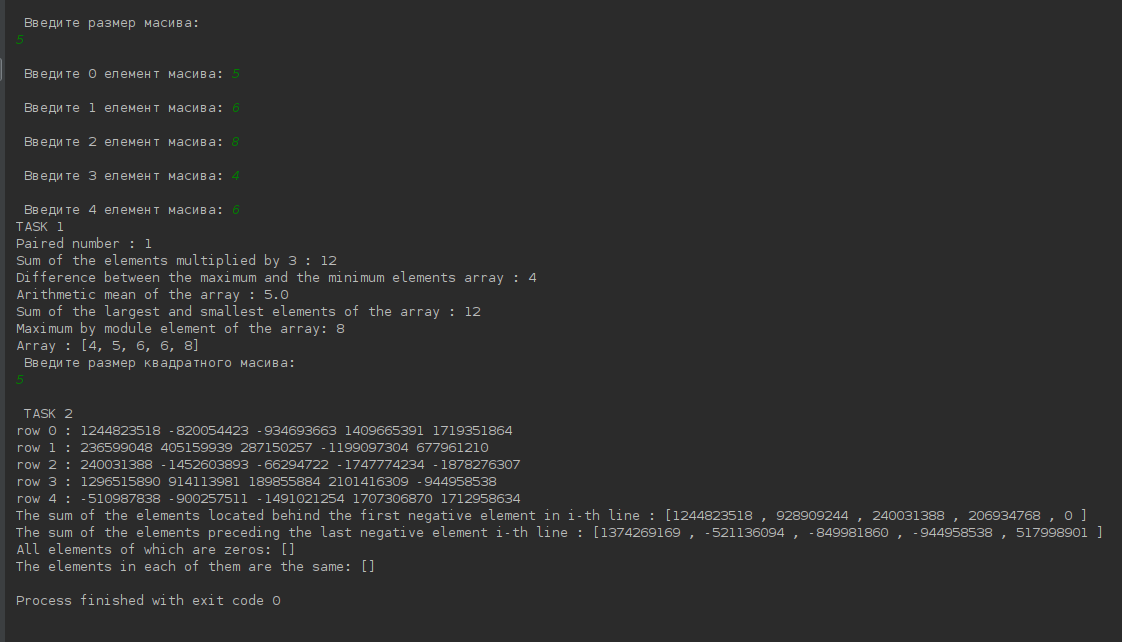
System.out.printf("\n");

}

}

}

3. В результаті отримаю наступне:



Висновок: Під час лабораторної роботи я навчився працювати з одномірними та багатомірними масивами і також навчився використовувати цикли, а саме це є конструкції for, while, do/while, for/in.