Міністерство освіти і науки України

НТУУ «КПІ ім. Ігоря Сікорського»

Кафедра автоматизації проектування енергетичних процесів і систем

Лабораторна робота №2

Разработка и тестирование программ в среде IntelliJ IDEA

Варіант №24

Виконав студент ІІ-го курсу

ТЕФ групи ТІ-02

Чуй Богдан Сергійович

Термін здачі:

Дата здачі:

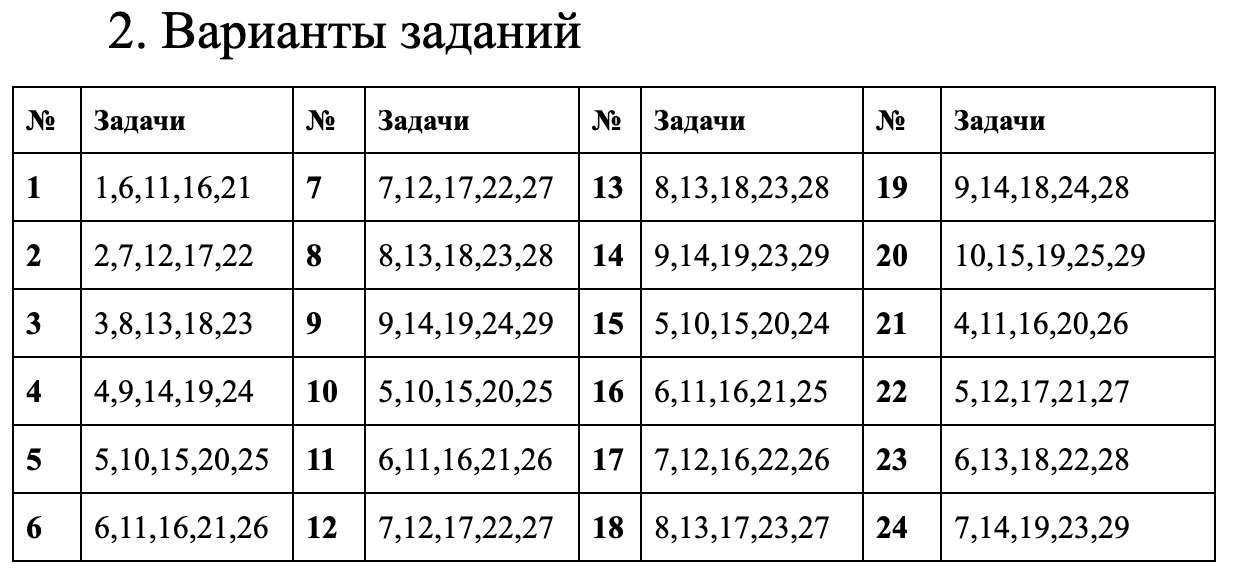
Перевірив: Тарнавський Ю.А.

Київ – 2021

7)Определить   абсолютное значение наименьшей разности между двумя любыми значениями элементов исходного одномерного массива **a**

14)Проверьте, являются ли элементы массива **a** множеством (для этого среди элементов массива не должно быть двух элементов с одинаковым значением)..    
19)Выведите на дисплей значения и индексы только тех элементов массива **a**, значения которых не равны значениям других элементов, т.е. уникальных элементов массива.

23)Определить индексы и значения равных элементов (если они есть) исходного   одномерного массива **a**.

29)Сформировать массив **b** из массива **a**  следующим образом: массив  **b** состоит из тех элементов массива **a**, которые повторяются в массиве (по одному значению для одинаковых элементов), например, для массива **a**: 3 7 4 3 8 7 5, массив **b** будет иметь вид: 3 7.

**import** java.util.Random;

*/\*\**

*\* Определить абсолютное значение наименьшей разности*

*\* между двумя любыми значениями элементов исходного одномерного массива a.*

*\*/*

**public class** Task7 {

**public static int**[] getRandomArray(**int** number) {

Random rd = **new** Random(); *// creating Random object*

**int**[] arr = **new int**[number];

**for** (**int** i = 0; i < arr.**length**; i++)

arr[i] = rd.nextInt() % 101; *// storing random integers in an array*

**return** arr;

}

**public static int** leastDefference(**int**[] args) {

**int** min = Integer.***MAX\_VALUE***;

**for** (**int** i = 0; i < args.**length** - 1; i++) {

**for** (**int** j = i + 1; j < args.**length**; j++) {

**int** tempMin = Math.*abs*(args[i] - args[j]);

**if** (tempMin < min)

min = tempMin;

}

}

**return** min;

}

**public static void** main(String[] args) {

**int** [] array = *getRandomArray*(10);

**for** (**int** b:array) {

System.***out***.print(b + **" "**);

}

System.***out***.println(**"\n"**+*leastDefference*(array));

}

}

**public class** Task14 {

*/\*\**

*\* Проверьте, являются ли элементы массива a множеством (для этого среди элементов массива не должно быть двух элементов с одинаковым значением)*

*\**

*\*/*

**public static boolean** isUnique(**int**[]arr){

**boolean** flag = **true**;

**for**(**int** i = 0; i < arr.**length**; i++){

**for**(**int** j = i + 1 ;j < arr.**length**; j++)

**if**(arr[i] == arr[j])

flag = **false**;

}

**return** flag;

}

**public static void** main(String[] args) {

**int** []array = {1,2,2,4,5,6,7,8};

System.***out***.println(*isUnique*(array));

}

}

**import** java.util.ArrayList;

**import** java.util.List;

**import** java.util.stream.IntStream;

*/\*\**

*\* Выведите на дисплей значения и индексы только тех элементов массива a, значения которых*

*\* не равны значениям других элементов, т.е. уникальных элементов массива.*

*\*/*

**public class** Task19 {

**public static** ArrayList<Integer> findElements(**int** []array){

ArrayList<Integer> indexArray = **new** ArrayList<Integer>();

**for** (**int** i = 0; i < array.**length**; i++) {

**for** (**int** j = i+1; j < array.**length**; j++) {

**if**(array[i]==array[j]){

indexArray.add(i);

indexArray.add(j);

}

}

}

ArrayList<Integer> unique = **new** ArrayList<Integer>();

**for**(**int** i = 0; i < array.**length**; i++){

**if**(!indexArray.contains(i)){

unique.add(array[i]);

}

}

**return** unique;

}

**public static void** print(**int**[] array, ArrayList<Integer> unique){

**for** (**int** i = 0; i < unique.size();i++){

**for** (**int** j = i+1; j < array.**length**; j++) {

**if**(array[j] == unique.get(i).intValue()){

System.***out***.println(**"element-> "** + array[j] + **" index-> "** + j);

}

}

}

}

**public static void** main(String[] args) {

**int** []array = {1,2,1,2,3};

*print*(array,*findElements*(array));

**for** (**int** i:*findElements*(array)) {

System.***out***.println(i);

}

}

}

**port** java.util.ArrayList;

**import** java.util.Iterator;

**import** java.util.List;

*/\*\**

*\* Определить индексы и значения равных элементов (если они есть) исходного одномерного массива a*

*\*/*

**public class** Task23 {

**public static** List findElements(**int** []array){

List<Integer> numbers = **new** ArrayList<Integer>();

**for** (**int** i = 0; i < array.**length**; i++){

**for**(**int** j = i+1; j <array.**length**; j++){

**if**(array[i] == array[j]){

**if**(!numbers.contains(**new** Integer(array[i]))){

numbers.add(**new** Integer(array[i]));

}

}

}

}

**return** numbers;

}

**public static** List findIndex(List numbers, **int**[]array){

List<Integer> index = **new** ArrayList<Integer>();

**for**(**int** i = 0; i < array.**length**; i++){

Iterator<Integer> iterator = numbers.iterator();

**while** (iterator.hasNext()) {

**if**(iterator.next().intValue() == array[i]){

index.add(**new** Integer(i));

}

}

}

**return** index;

}

**public static void** main(String[] args) {

**int** []array = {1,2,3,4,2,6,8,9,1,2};

List<Integer> numbers = *findElements*(array);

Iterator<Integer> iterator = numbers.iterator();

System.***out***.println(**"Numbers: "**);

**while** (iterator.hasNext()){

System.***out***.print(iterator.next().intValue() + **" "**);

}

System.***out***.println(**"\nindex: "**);

List<Integer> index = *findIndex*(numbers, array);

iterator = index.iterator();

**while** (iterator.hasNext()){

System.***out***.print(iterator.next().intValue() + **" "**);

}

}

}

**import** java.util.ArrayList;

**import** java.util.Iterator;

**import** java.util.List;

*/\*\**

*\* Сформировать массив b из массива a следующим образом: массив b состоит*

*\* из тех элементов массива a, которые повторяются в массиве (по одному значению для одинаковых элементов),*

*\* например, для массива a: 3 7 4 3 8 7 5, массив b будет иметь вид: 3 7.*

*\*/*

**public class** Task29 {

**public static** List findElements(**int** []array){

List<Integer> numbers = **new** ArrayList<Integer>();

**for** (**int** i = 0; i < array.**length**; i++){

**for**(**int** j = i+1; j <array.**length**; j++){

**if**(array[i] == array[j]){

**if**(!numbers.contains(**new** Integer(array[i]))){

numbers.add(**new** Integer(array[i]));

}

}

}

}

**return** numbers;

}

**public static int**[] listToArray(List list){

**int**[] array = **new int**[list.size()];

Iterator<Integer> iterator = list.iterator();

**for** (**int** i = 0; i < array.**length**; i++){

array[i] = iterator.next();

}

**return** array;

}

}

**import** org.junit.Assert;

**import** org.junit.jupiter.api.Test;

**public class** Task7Test {

@Test

**void** getRandomArray() {

**boolean** flag = **true**;

**int** []arr = Task7.*getRandomArray*(10);

**if**(arr.**length** != 10)

flag = **false**;

Assert.*assertTrue*(flag);

}

@Test

**void** leastDefference() {

**int** []arr = {1,5,7,-10,-20,98,14,56};

**int** result = Task7.*leastDefference*(arr);

Assert.*assertTrue*(result == 2);

}

@Test

**void** main() {

}

}

**import** org.junit.Assert;

**import** org.junit.jupiter.api.Assertions;

**import** org.junit.jupiter.api.Test;

**import static** org.junit.jupiter.api.Assertions.\*;

**class** Task14Test {

@Test

**void** isUnique() {

**int** []array = {1,2,3,4,5,6,7,8,8};

*assertFalse*(Task14.*isUnique*(array));

}

}

**import** org.junit.jupiter.api.Test;

**import** java.util.ArrayList;

**import static** org.junit.jupiter.api.Assertions.\*;

**class** Task19Test {

**int**[]**array** = {1,2,1,2,3};

@Test

**void** findElements() {

ArrayList<Integer> expected = **new** ArrayList<Integer>();

expected.add(**new** Integer(3));

*assertEquals*(Task19.*findElements*(**array**), expected);

}

}

**import** org.junit.Assert;

**import** org.junit.jupiter.api.Test;

**import** java.util.Iterator;

**import** java.util.List;

**import static** org.junit.jupiter.api.Assertions.\*;

**class** Task23Test {

**int** []**array** = {1,2,3,4,2,6,8,9,1,2};

@Test

**void** findElements() {

**boolean** flag = **true**;

List<Integer> numbers = Task23.*findElements*(**array**);

Iterator<Integer> iterator = numbers.iterator();

**if**(iterator.next().intValue() != 1 ) flag = **false**;

**if**(iterator.next().intValue() != 2 ) flag = **false**;

Assert.*assertTrue*(flag);

}

@Test

**void** findIndex() {

**boolean** flag = **true**;

List<Integer> index = Task23.*findIndex*(Task23.*findElements*(**array**), **array**);

Iterator<Integer> iterator = index.iterator();

**if**(iterator.next().intValue() != 0 ) flag = **false**;

**if**(iterator.next().intValue() != 1 ) flag = **false**;

**if**(iterator.next().intValue() != 4 ) flag = **false**;

**if**(iterator.next().intValue() != 8 ) flag = **false**;

**if**(iterator.next().intValue() != 9 ) flag = **false**;

Assert.*assertTrue*(flag);

}

**import** org.junit.Assert;

**import** org.junit.jupiter.api.Test;

**import** java.util.Iterator;

**import** java.util.List;

**import static** org.junit.jupiter.api.Assertions.\*;

**class** Task29Test {

**int** []**array** = {1,2,3,4,2,6,8,9,1,2};

@Test

**void** findElements() {

**boolean** flag = **true**;

List<Integer> numbers = Task23.*findElements*(**array**);

Iterator<Integer> iterator = numbers.iterator();

**if** (iterator.next().intValue() != 1) flag = **false**;

**if** (iterator.next().intValue() != 2) flag = **false**;

Assert.*assertTrue*(flag);

}

@Test

**void** listToArray() {

**int** []arr = Task29.*listToArray*(Task29.*findElements*(**array**));

**boolean** flag = **true**;

**for**(**int** i = 0; i<arr.**length**; i++){

**if**(arr[i] != **array**[i]){

flag = **false**;

}

}

Assert.*assertTrue*(flag);

}

}