Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Кафедра цифрових технологій в енергетиці

ЗВІТ   
з виконання лабораторної роботи №2  
з дисципліни «Технології Java конструювання програмного забезпечення»

«**РОЗРОБКА І ТЕСТУВАННЯ ПРОГРАМ В INTELLIJ IDEA**»

Варіант 17

Виконав: студент групи ТР-23

Ровний Г.О.

Перевірив: доцент, к.ф.-м.н.  
Тарнавський Ю.А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Завдання 1

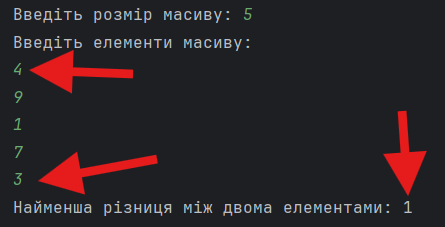


package org.example;  
  
import java.util.Arrays;  
import java.util.Scanner;  
  
public class Task7 {  
  
 public static int findMinDifference(int[] a) {  
 if (a == null || a.length < 2) {  
 throw new IllegalArgumentException("Масив повинен містити принаймні два елементи.");  
 }  
  
 Arrays.*sort*(a);  
 int minDifference = Integer.*MAX\_VALUE*;  
 for (int i = 1; i < a.length; i++) {  
 int difference = Math.*abs*(a[i] - a[i - 1]);  
 if (difference < minDifference) {  
 minDifference = difference;  
 }  
 }  
  
 return minDifference;  
 }  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Введіть розмір масиву: ");  
 int size = scanner.nextInt();  
 int[] a = new int[size];  
 System.*out*.println("Введіть елементи масиву:");  
 for (int i = 0; i < size; i++) {  
 a[i] = scanner.nextInt();  
 }  
 int minDifference = *findMinDifference*(a);  
 System.*out*.println("Найменша різниця між двома елементами: " + minDifference);  
 }  
}

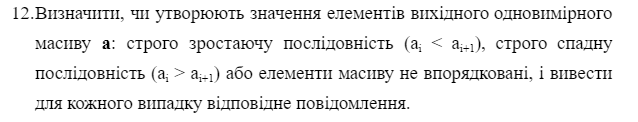
Тести:

package org.example;  
  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
  
class Task7Test {  
 @Test  
 public void testFindMinDifferencePositiveNumbers() {  
 int[] array = {4, 9, 1, 7, 3};  
 int result = Task7.*findMinDifference*(array);  
 *assertEquals*(1, result);  
 }  
  
 @Test  
 void testFindMinDifferenceNegativeNumbers() {  
 int[] array = {-5, -2, -8, -3};  
 int result = Task7.*findMinDifference*(array);  
 *assertEquals*(1, result);  
 }  
  
 @Test  
 void testFindMinDifferenceSameNumbers() {  
 int[] array = {5, 5, 5, 5};  
 int result = Task7.*findMinDifference*(array);  
 *assertEquals*(0, result);  
 }  
  
 @Test  
 void testFindMinDifferenceSingleElement() {  
 int[] array = {10};  
 Exception exception = *assertThrows*(IllegalArgumentException.class, () -> {  
 Task7.*findMinDifference*(array);  
 });  
 *assertEquals*("Масив повинен містити принаймні два елементи.", exception.getMessage());  
 }  
}

Скріншот:



# Завдання 2



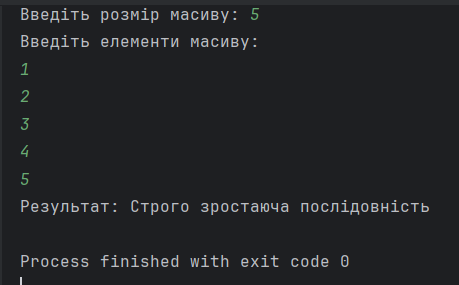
Код:

// Визначити, чи утворюють значення елементів вихідного одновимірного масиву a:  
// строго зростаючу послідовність (ai < ai+1), строго спадну послідовність (ai > ai+1)  
// або елементи масиву не впорядковані, і вивести для кожного випадку відповідне повідомлення.  
  
package org.example;  
  
import java.util.Scanner;  
  
public class Task12 {  
  
 public static String checkSequence(int[] a) {  
 if (a.length < 2) {  
 throw new IllegalArgumentException("Масив повинен містити принаймні два елементи");  
 }  
  
 boolean isIncreasing = true;  
 boolean isDecreasing = true;  
  
 for (int i = 1; i < a.length; i++) {  
 if (a[i] > a[i - 1]) {  
 isDecreasing = false;  
 } else if (a[i] < a[i - 1]) {  
 isIncreasing = false;  
 }  
 }  
  
 if (isIncreasing) {  
 return "Строго зростаюча послідовність";  
 } else if (isDecreasing) {  
 return "Строго спадна послідовність";  
 } else {  
 return "Елементи масиву не впорядковані";  
 }  
 }  
  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Введіть розмір масиву: ");  
 int size = scanner.nextInt();  
 int[] a = new int[size];  
 System.*out*.println("Введіть елементи масиву:");  
 for (int i = 0; i < size; i++) {  
 a[i] = scanner.nextInt();  
 }  
 String result = *checkSequence*(a);  
 System.*out*.println("Результат: " + result);  
 }  
  
}

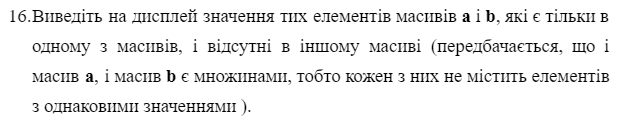
Тести:

package org.example;  
  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
  
class Task12Test {  
  
 @Test  
 void testStrictlyIncreasingSequence() {  
 int[] array = {1, 2, 3, 4, 5};  
 String result = Task12.*checkSequence*(array);  
 *assertEquals*("Строго зростаюча послідовність", result);  
 }  
  
 @Test  
 void testStrictlyDecreasingSequence() {  
 int[] array = {5, 4, 3, 2, 1};  
 String result = Task12.*checkSequence*(array);  
 *assertEquals*("Строго спадна послідовність", result);  
 }  
  
 @Test  
 void testUnorderedSequence() {  
 int[] array = {3, 1, 4, 2};  
 String result = Task12.*checkSequence*(array);  
 *assertEquals*("Елементи масиву не впорядковані", result);  
 }  
  
 @Test  
 void testSingleElement() {  
 int[] array = {5};  
 Exception exception = *assertThrows*(IllegalArgumentException.class, () -> {  
 Task12.*checkSequence*(array);  
 });  
 *assertEquals*("Масив повинен містити принаймні два елементи", exception.getMessage());  
 }  
  
}

Скріншот:



# Завдання 3



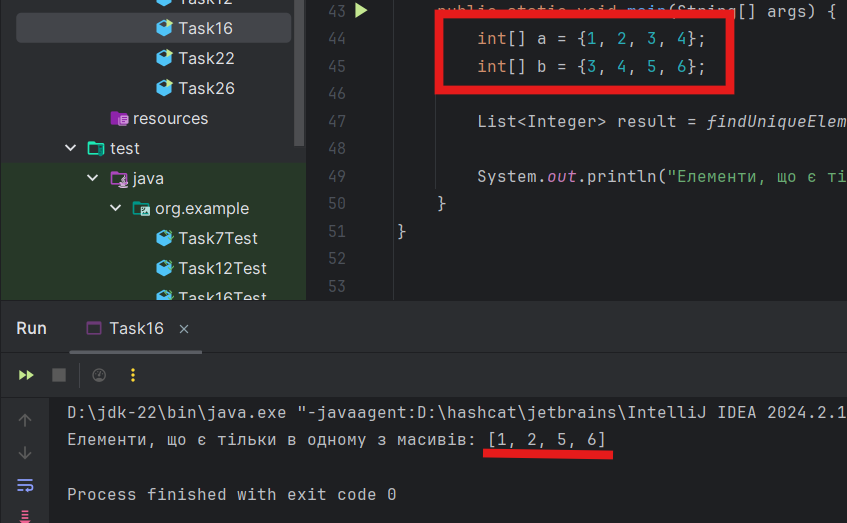
Код:

// Виведіть на дисплей значення тих елементів масивів a і b,  
// які є тільки в одному з масивів, і відсутні в іншому масиві  
// (передбачається, що і масив a, і масив b є множинами, тобто  
// кожен з них не містить елементів з однаковими значеннями).  
  
package org.example;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class Task16 {  
  
 public static List<Integer> findUniqueElements(int[] a, int[] b) {  
 List<Integer> uniqueElements = new ArrayList<>();  
 for (int valueA : a) {  
 boolean foundInB = false;  
 for (int valueB : b) {  
 if (valueA == valueB) {  
 foundInB = true;  
 break;  
 }  
 }  
 if (!foundInB) {  
 uniqueElements.add(valueA);  
 }  
 }  
 for (int valueB : b) {  
 boolean foundInA = false;  
 for (int valueA : a) {  
 if (valueB == valueA) {  
 foundInA = true;  
 break;  
 }  
 }  
 if (!foundInA) {  
 uniqueElements.add(valueB);  
 }  
 }  
  
 return uniqueElements;  
 }  
  
 public static void main(String[] args) {  
 int[] a = {1, 2, 3, 4};  
 int[] b = {3, 4, 5, 6};  
  
 List<Integer> result = *findUniqueElements*(a, b);  
  
 System.*out*.println("Елементи, що є тільки в одному з масивів: " + result);  
 }  
}

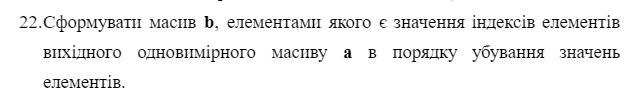
Тести:

package org.example;  
  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
  
import java.util.List;  
  
class Task16Test {  
  
 @Test  
 void testFindUniqueElementsDifferentSets() {  
 int[] arrayA = {1, 2, 3, 4};  
 int[] arrayB = {3, 4, 5, 6};  
 List<Integer> result = Task16.*findUniqueElements*(arrayA, arrayB);  
 *assertIterableEquals*(List.*of*(1, 2, 5, 6), result);  
 }  
  
 @Test  
 void testFindUniqueElementsNoUniqueElements() {  
 int[] arrayA = {1, 2, 3};  
 int[] arrayB = {1, 2, 3};  
 List<Integer> result = Task16.*findUniqueElements*(arrayA, arrayB);  
 *assertTrue*(result.isEmpty());  
 }  
  
 @Test  
 void testFindUniqueElementsOnlyInA() {  
 int[] arrayA = {1, 2, 3};  
 int[] arrayB = {4, 5, 6};  
 List<Integer> result = Task16.*findUniqueElements*(arrayA, arrayB);  
 *assertIterableEquals*(List.*of*(1, 2, 3, 4, 5, 6), result);  
 }  
  
}

Скріншот:



# Завдання 4



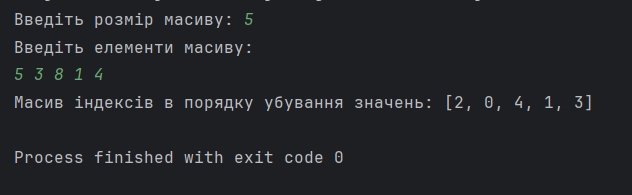
Код:

// Сформувати масив b, елементами якого є значення індексів  
// елементів вихідного одновимірного масиву a в порядку убування  
// значень елементів.  
  
  
package org.example;  
  
import java.util.Arrays;  
import java.util.Comparator;  
import java.util.Scanner;  
  
public class Task22 {  
  
 public static int[] createIndexArray(int[] a) {  
 Integer[] indices = new Integer[a.length];  
 for (int i = 0; i < a.length; i++) {  
 indices[i] = i; // Додаємо індекс ел.  
 }  
 Arrays.*sort*(indices, new Comparator<Integer>() {  
 @Override  
 public int compare(Integer index1, Integer index2) {  
 return Integer.*compare*(a[index2], a[index1]); // Убування значень  
 }  
 });  
 return Arrays.*stream*(indices).mapToInt(Integer::intValue).toArray();  
 }  
  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Введіть розмір масиву: ");  
 int size = scanner.nextInt();  
 int[] a = new int[size];  
 System.*out*.println("Введіть елементи масиву:");  
 for (int i = 0; i < size; i++) {  
 a[i] = scanner.nextInt();  
 }  
 int[] b = *createIndexArray*(a);  
 System.*out*.println("Масив індексів в порядку убування значень: " + Arrays.*toString*(b));  
 }  
}

Тести:

package org.example;  
  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
  
class Task22Test {  
  
 @Test  
 void testCreateIndexArraySortedDescending() {  
 int[] arrayA = {5, 3, 8, 1, 4};  
 int[] expectedIndexArray = {2, 0, 4, 1, 3}; // Індикація елементів у порядку убування  
 int[] result = Task22.*createIndexArray*(arrayA);  
 *assertArrayEquals*(expectedIndexArray, result);  
 }  
  
 @Test  
 void testCreateIndexArrayAllEqual() {  
 int[] arrayA = {7, 7, 7};  
 int[] expectedIndexArray = {0, 1, 2}; // Індикація елементів, порядок може бути довільним  
 int[] result = Task22.*createIndexArray*(arrayA);  
 *assertArrayEquals*(expectedIndexArray, result);  
 }  
  
 @Test  
 void testCreateIndexArraySingleElement() {  
 int[] arrayA = {10};  
 int[] expectedIndexArray = {0}; // Єдиний елемент  
 int[] result = Task22.*createIndexArray*(arrayA);  
 *assertArrayEquals*(expectedIndexArray, result);  
 }  
  
 @Test  
 void testCreateIndexArrayEmpty() {  
 int[] arrayA = {};  
 int[] expectedIndexArray = {}; // Порожній масив  
 int[] result = Task22.*createIndexArray*(arrayA);  
 *assertArrayEquals*(expectedIndexArray, result);  
 }  
}

Скріншот:



# Завдання 5



Код:

// Сформувати масив b, елементами якого є елементи вихідного  
// одновимірного масиву a, розташовані в зворотному порядку.  
  
package org.example;  
  
import java.util.Scanner;  
import java.util.Arrays;  
  
public class Task26 {  
  
 public static int[] reverseArray(int[] a) {  
 int[] b = new int[a.length];  
 for (int i = 0; i < a.length; i++) {  
 b[i] = a[a.length - 1 - i];  
 }  
 return b;  
 }  
  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Введіть розмір масиву: ");  
 int size = scanner.nextInt();  
 int[] a = new int[size];  
 System.*out*.println("Введіть елементи масиву:");  
 for (int i = 0; i < size; i++) {  
 a[i] = scanner.nextInt();  
 }  
 int[] b = *reverseArray*(a);  
 System.*out*.println("Масив в зворотному порядку: " + Arrays.*toString*(b));  
 }  
}

Тести:

package org.example;  
  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
  
class Task26Test {  
  
 @Test  
 void testReverseArray() {  
 int[] arrayA = {1, 2, 3, 4, 5};  
 int[] expectedArrayB = {5, 4, 3, 2, 1}; // Очікую  
 int[] result = Task26.*reverseArray*(arrayA);  
 *assertArrayEquals*(expectedArrayB, result);  
 }  
 @Test  
 void testReverseArraySingleElement() {  
 int[] arrayA = {10};  
 int[] expectedArrayB = {10};  
 int[] result = Task26.*reverseArray*(arrayA);  
 *assertArrayEquals*(expectedArrayB, result);  
 }  
 @Test  
 void testReverseArrayEmpty() {  
 int[] arrayA = {};  
 int[] expectedArrayB = {};  
 int[] result = Task26.*reverseArray*(arrayA);  
 *assertArrayEquals*(expectedArrayB, result);  
 }  
}

Скріншот:

