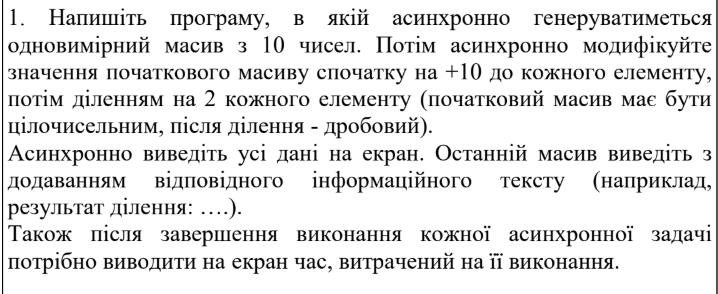
Група ТВ-22

П.І.Б. Іщук П.О.

Варіант за списком групи 5

**Практична робота №4**

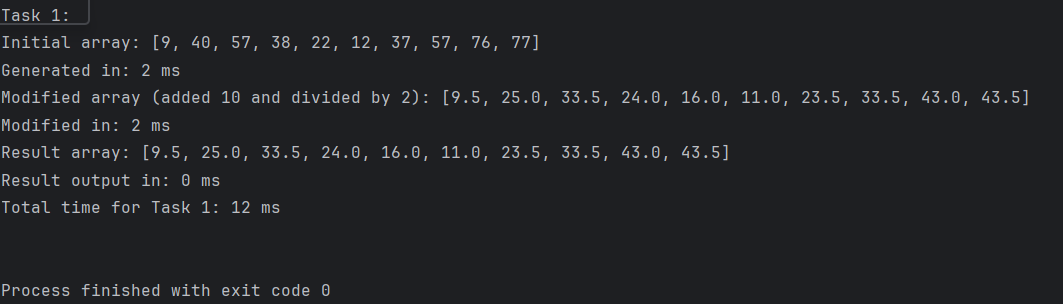
**Завдання** 1



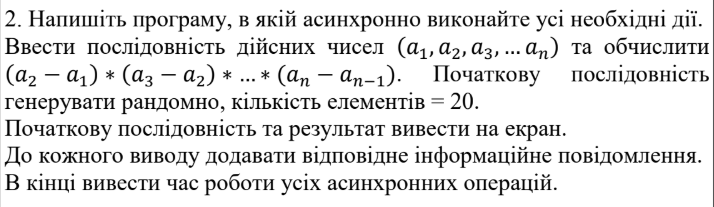
**Код програми**

import java.util.Arrays;  
import java.util.Random;  
import java.util.concurrent.CompletableFuture;  
import java.util.concurrent.TimeUnit;  
  
public class Task1 {  
  
 public static void main(String[] args) {  
 System.*out*.println("Task 1:");  
 long task1StartTime = System.*currentTimeMillis*();  
  
 CompletableFuture<int[]> initialArrayFuture = CompletableFuture.*supplyAsync*(() -> {  
 long start = System.*currentTimeMillis*();  
 int[] array = new Random().ints(10, 1, 100).toArray();  
 System.*out*.println("Initial array: " + Arrays.*toString*(array));  
 System.*out*.println("Generated in: " + (System.*currentTimeMillis*() - start) + " ms");  
 return array;  
 });  
  
 CompletableFuture<double[]> modifiedArrayFuture = initialArrayFuture.thenApplyAsync(array -> {  
 long start = System.*currentTimeMillis*();  
 int[] addedArray = Arrays.*stream*(array).map(x -> x + 10).toArray();  
 double[] dividedArray = Arrays.*stream*(addedArray).mapToDouble(x -> x / 2.0).toArray();  
 System.*out*.println("Modified array (added 10 and divided by 2): " + Arrays.*toString*(dividedArray));  
 System.*out*.println("Modified in: " + (System.*currentTimeMillis*() - start) + " ms");  
 return dividedArray;  
 });  
  
 modifiedArrayFuture.thenAcceptAsync(array -> {  
 long start = System.*currentTimeMillis*();  
 System.*out*.println("Result array: " + Arrays.*toString*(array));  
 System.*out*.println("Result output in: " + (System.*currentTimeMillis*() - start) + " ms");  
 }).thenRunAsync(() -> {  
 long task1EndTime = System.*currentTimeMillis*();  
 System.*out*.println("Total time for Task 1: " + (task1EndTime - task1StartTime) + " ms\n");  
 });  
  
 try {  
 TimeUnit.*SECONDS*.sleep(3);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
}

**Скриншот результатів роботи**



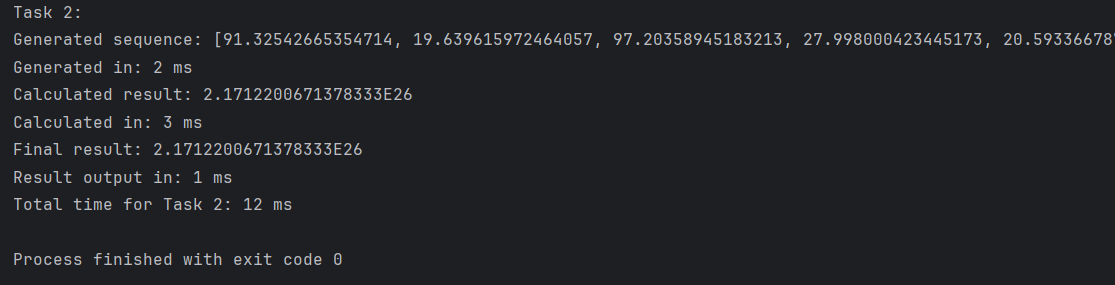
**Завдання** 2



**Код програми**

import java.util.Arrays;  
import java.util.Random;  
import java.util.concurrent.CompletableFuture;  
import java.util.concurrent.TimeUnit;  
  
public class Task2 {  
  
 public static void main(String[] args) {  
 System.*out*.println("Task 2:");  
 long task2StartTime = System.*currentTimeMillis*();  
  
 CompletableFuture<double[]> sequenceFuture = CompletableFuture.*supplyAsync*(() -> {  
 long start = System.*currentTimeMillis*();  
 double[] sequence = new Random().doubles(20, 1, 100).toArray();  
 System.*out*.println("Generated sequence: " + Arrays.*toString*(sequence));  
 System.*out*.println("Generated in: " + (System.*currentTimeMillis*() - start) + " ms");  
 return sequence;  
 });  
  
 CompletableFuture<Double> resultFuture = sequenceFuture.thenApplyAsync(sequence -> {  
 long start = System.*currentTimeMillis*();  
 double result = 1.0;  
 for (int i = 1; i < sequence.length; i++) {  
 result \*= (sequence[i] - sequence[i - 1]);  
 }  
 System.*out*.println("Calculated result: " + result);  
 System.*out*.println("Calculated in: " + (System.*currentTimeMillis*() - start) + " ms");  
 return result;  
 });  
  
 resultFuture.thenAcceptAsync(result -> {  
 long start = System.*currentTimeMillis*();  
 System.*out*.println("Final result: " + result);  
 System.*out*.println("Result output in: " + (System.*currentTimeMillis*() - start) + " ms");  
 }).thenRunAsync(() -> {  
 long task2EndTime = System.*currentTimeMillis*();  
 System.*out*.println("Total time for Task 2: " + (task2EndTime - task2StartTime) + " ms");  
 });  
  
 try {  
 TimeUnit.*SECONDS*.sleep(3);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
}

**Скриншот результатів роботи**

****