 **Brahim Saddike** (brahim.saddike4@gmail.com)

Campagne : [SOFTEAM] Training Java 2

Domaine(s) : Java

Langage : Anglais

Date : 07/12/2020

SCORE

69%

810 / 1 180 pts



DURÉE

0h14 / 0h49

Java 810 / 1 180pts (69%)

Connaissance du langage



200 / 240pts

Fiabilité



158 / 231pts

Résolution de problèmes



452 / 709pts

Question 1: Abstract class



Java



00:09 / 00:20



20 / 20 pts



Question

An abstract class can contain concrete methods.



Réponse



True



False



Résultat



Réponse correcte
Connaissance du langage +20pts

Question 2: Visibility of attributes



Java



00:13 / 00:20



20 / 20 pts



Question

Privates attributes are visible from subclasses.



Réponse



True



False



Résultat



Réponse correcte

Connaissance du langage +20pts

Question 3: Thread



Java



00:09 / 00:20



20 / 20 pts



Question

Which method is called when a thread is executed?



Réponse



do()



run()



exec()



execute()



play()



Résultat



Réponse correcte
Connaissance du langage +20pts

Question 4: Interfaces



Java



00:06 / 00:20



40 / 40 pts



Question

In Java 8, interfaces can contain concrete methods.



Réponse



True



False



Résultat



Réponse correcte
Connaissance du langage +40pts

Question 5: Equals and hashCode



Java



00:20 / 00:20



0 / 40 pts

⚠ Le temps alloué à cette question s'est écoulé. La réponse du candidat a été automatiquement récupérée à la fin du décompte.



Question

If two objects are equals then they should have the same hashCode.



Réponse



True



False



Résultat



Réponse incorrecte

Connaissance du langage ~~+40pts~~

Question 6: Garbage collector



Java



00:07 / 00:20



40 / 40 pts



Question

The garbage collector ensures that there is enough memory to run a Java program.



Réponse



True



False



Résultat



Réponse correcte

Connaissance du langage +40pts

Question 7: Largest wins from chaos



Java



03:45 / 05:00



90 / 100 pts



Question

Algorithm.findLargest(int[] numbers) should return the largest number from *numbers*. The array *numbers* always contains at least one number.

Implement *Algorithm.findLargest(int[] numbers)*.



Réponse

```
1 class Algorithm {
2
3     /** @return the largest number of the given array */
4     static int findLargest(int[] numbers) {
5         // Your code goes here
6         int largest = 0;
7         if(numbers.length <= 1) return numbers[0];
8         for(int i =0; i<numbers.length; i++){
9             if(largest < numbers[i]){
10                 largest = numbers[i];
11             }
12         }
13         return largest;
14     }
15 }
16 }
```



Résultat



It works using simple data sample

Résolution de problèmes +32pts



Still works when the array contains only Integer.MIN_VALUE

Fiabilité +58pts



Still works if the largest number is at position 0 in the array

Fiabilité +5pts



Still works if the largest number is at the last position in the array

Fiabilité +5pts

Question 8: Combination options in a tournament



Java



04:49 / 15:00



300 / 300 pts

? Question

You have to organize a chess tournament in which players will compete head-to-head.

Here is how we proceed to form the duels: select a first player randomly, then, select his opponent at random among the remaining participants. The pair of competitors obtained forms one of the duels of the tournament. We proceed in the same manner to form all the other pairs.

In this exercise, we would like to know how many pairs it is possible to form knowing that the order of opponents in a pair does not matter.

For example, with 4 players named A, B, C and D, it is possible to get 6 different pairs : AB, AC, AD, BC, BD, CD.

Implement **count** to return the number of possible pairs. Parameter **n** corresponds to the number of players.

Try to optimize your solution so that, ideally, the duration of treatment is the same for any **n**.

Input: $2 \leq n \leq 10000$



Réponse

```
1 import java.util.*;
2 import java.io.*;
3 import java.nio.*;
4 import java.math.*;
5
6 class Solution {
7
8     /** Counts the number of pairs for n players. */
9     static int count(int n) {
10         if(2 <= n || n>=1000){
11             return n*(n-1)/2;
12         }
13         return 0;
14     }
15 }
```

Résultat

- ✓ The solution works with a simple data set
Résolution de problèmes +128pts
- ✓ The solution does not use excessive memory (no cache)
Résolution de problèmes +43pts
- ✓ The solution returns the correct result in constant time
Résolution de problèmes +86pts
- ✓ The solution works with data close to 10000
Résolution de problèmes +43pts

Question 9: Move towards zero



Java



03:54 / 15:00



280 / 300 pts

Question

Implement ***closestToZero*** to return the integer in the array ***ints*** that is closest to zero.
If there are two integers equally close to zero, consider the positive element to be closer to zero
(example: if ***ints*** contains -5 and 5, return 5). If ***ints*** is ***null*** or empty, return 0.

Input: integers in ***ints*** have values ranging from -2147483647 to 2147483647.

Réponse

```
1 class A {
2
3     /** @return the number that is closest to zero */
4     static int closestToZero(int[] ints) {
5         if(ints == null || ints.length == 0){
6             return 0;
7         }
8         int clos = ints[0];
9         for(int i=0; i<ints.length; i++){
10             if(Math.abs(ints[i]) < Math.abs(clos)){
11                 clos=ints[i];
12             }
13         }
14         return clos;
15     }
16 }
```

Résultat

- ✓ The result is correct with a simple data set [7, 5, 9, 1, 4]
Résolution de problèmes +120pts
- ✓ The solution works with 2147483647 or -2147483647
Fiabilité +20pts
- ✓ The solution works when the array contains only negative integers
Fiabilité +20pts
- ✗ When two integers are as close to 0, then the positive wins
Fiabilité ~~+20pts~~
- ✓ The solution works when the array contains only two equal negative integers
Fiabilité +20pts
- ✓ The solution uses java.lang.Math.abs()
Connaissance du langage +60pts
- ✓ The solution works with an empty array
Fiabilité +20pts
- ✓ The solution works with a null array
Fiabilité +20pts

Question 10: Approximation of π



Java



00:23 / 12:00



0 / 300 pts

? Question

In this exercise we will calculate an approximation of π (Pi).

The technique is as follows:

Take a random point P at coordinate (x, y) such that $0 \leq x \leq 1$ and $0 \leq y \leq 1$. If $x^2 + y^2 \leq 1$, then the point is inside the quarter disk of radius 1, otherwise the point is outside.

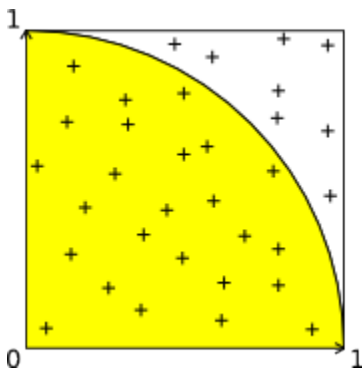


Fig 1. An example using 33 random points.

We know that the probability that the point is inside the quarter disk is equal to $\pi/4$.

Write the method ***double approx(double[][] pts)*** who will use the points ***pts*** to return an approximation of the number π .

Input:


Each item in ***pts*** is a point. A point is represented by an array containing exactly two numbers, respectively, ***x*** and ***y*** such that $0 \leq x \leq 1$ and $0 \leq y \leq 1$. ***pts*** is never null and always contains at least one item.

Réponse

```
1  /**
2   * This class defines a method to approximate pi
3   */
4  class Pi {
5
6      /**
7       * Approximate pi using the given points.
8       */
9      static double approx(double[][] pts) {
10         return 0d;
11     }
12 }
```

Résultat

 Approximation of π is correct (related to pts)
Résolution de problèmes ~~+257pts~~

 The point P(1, 0) is inside the quarter disk
Fiabilité ~~+43pts~~

Glossaire

Connaissance du langage

La mesure de cette compétence permet de déterminer l'expérience du candidat dans la pratique d'un langage de programmation. **Privilégiez cette compétence si, par exemple, vous recherchez un développeur qui devra être rapidement opérationnel.**

Modélisation

Cette mesure fournit une indication sur la capacité du candidat à appliquer des solutions standard pour résoudre des problèmes récurrents. Un développeur ayant un bon niveau dans cette compétence augmentera la qualité (maintenabilité, évolutivité) de vos applications. Cette compétence ne dépend pas spécifiquement d'une technologie. **Privilégiez cette compétence si, par exemple, vous recherchez un développeur qui sera amené à travailler sur les briques qui structurent vos applications, à anticiper les besoins de demain pour développer des solutions pérennes.**

Résolution de problèmes

Cette compétence correspond aux aptitudes du candidat à comprendre et à structurer son raisonnement pour trouver des solutions à des problèmes complexes. Cette compétence ne dépend pas spécifiquement d'une technologie. **Privilégiez cette compétence si, par exemple, vos applications ont une composante technique importante (R&D, innovation).**

Fiabilité

La fiabilité caractérise la capacité du candidat à réaliser des solutions qui prennent en compte les cas particuliers. Plus cette compétence est élevée, plus vos applications sont robustes (moins de bugs).