



Rapport de l'évaluation: Développeur java

david petit

Candidat

ID : 148

Adresse mail :

Téléphone :

Diplôme :

Ecole/Université :

Année d'expérience : 0

Votre commentaire :

Commentaire candidat: ceci est un
exemple de rapport!.

Session

ID session : 5mag

Début : 2016-08-06 14:14:29

Fin : 2016-08-06 14:41:09

Test	Type de test	Score absolu	Percentile	Score Global 9 / 13
1. [JAVA] Java core (JSE)	QCM	4 / 8	<input type="text" value="73%"/>	
2. [JAVA] Optimisation de distance	Coding	5 / 5	<input type="text" value="80%"/>	

1. [JAVA] Java core (JSE)

1-1. Détails QCM

Question	Candidat answer	Correct answer	Marks
Question 1 " public class Test implements Device { public void dolt() { } } abstract class Phone1 extends Test { } abstract class Phone2 extends Test { public void dolt(int x) { } } class Phone3 extends Test implements Device { public void doStuff() { } } interface Device { public void dolt(); } Que se passe t'il au moment de la compilation de ce code?":	compilation sans erreur	compilation sans erreur	1
Question 2 "	BD	ne compile pas	0

<pre>class Top { public Top(String s) { System.out.print("B"); } } public class Test extends Top { public Test(String s) { System.out.print("D"); } public static void main(String[] args) { new Test("C"); } }</pre> <p>Qu'affiche ce programme?":</p>			
<p>Question 3 "</p> <pre>class Top { protected final void flipper() { System.out.println("Top"); } } public class Test extends Top { public final void flipper() { System.out.println("Test"); } } public static void main(String[] args) { new Top().flipper(); } }</pre> <p>Qu'affiche ce programme?":</p>	skiped this question	ne compile pas	0
<p>Question 4 "</p> <pre>class Dog { public void bark() { System.out.print("woof "); } } class Hound extends Dog { public void sniff() { System.out.print("sniff "); } public void bark() { System.out.print("howl "); } } public class DogShow { public static void main(String[] args) { new DogShow().go(); } void go() { new Hound().bark(); ((Dog) new Hound()).bark(); ((Dog) new Hound()).sniff(); } }</pre> <p>Qu'affiche ce programme?":</p>	ne compile pas	ne compile pas	1
<p>Question 5 "</p>	alpha subsub	alpha subsub	1

<pre> class Alpha { static String s = " "; protected Alpha() { s += "alpha "; } } class SubAlpha extends Alpha { private SubAlpha() { s += "sub "; } } public class SubSubAlpha extends Alpha { private SubSubAlpha() { s += "subsub "; } public static void main(String[] args) { new SubSubAlpha(); System.out.println(s); } } } </pre> <p>Qu'affiche ce programme?":</p>			
<p>Question 6 "</p> <pre> class Beta { } class Alpha { static Beta b1; Beta b2; } public class Tester { public static void main(String[] args) { Beta b1 = new Beta(); Beta b2 = new Beta(); Alpha a1 = new Alpha(); Alpha a2 = new Alpha(); a1.b1 = b1; a1.b2 = b1; a2.b2 = b2; a1 = null; b1 = null; b2 = null; // do stuff } } </pre> <p>Au niveau de do stuff combien d'objets sont éligibles au Gargage Collector? ":</p>	2	1	0
<p>Question 7 "</p> <pre> public class Ebb { static int x = 7; public static void main(String[] args) { String s = ""; for (int y = 0; y < 3; y++) { x++; switch (x) { case 8: s += "8 "; case 9: s += "9 "; case 10: { s += "10 "; break; } default: s += "d "; case 13: s += "13 "; } } System.out.println(s); } } </pre> <p>Qu'affiche ce programme?":</p>	skiped this question	8 9 10 9 10 10	0

<p>Question 8 " import java.util.*;</p> <pre> public class MapEQ { public static void main(String[] args) { Map m = new HashMap(); ToDos t1 = new ToDos("Monday"); ToDos t2 = new ToDos("Monday"); ToDos t3 = new ToDos("Tuesday"); m.put(t1, "doLaundry"); m.put(t2, "payBills"); m.put(t3, "cleanAttic"); System.out.println(m.size()); } } class ToDos { String day; ToDos(String d) { day = d; } public boolean equals(Object o) { return ((ToDos) o).day == this.day; } public int hashCode() { return 9; } } </pre> <p>Qu'affiche ce programme?":</p>	2	2	1

<p>2. [JAVA] Optimisation de distance</p>																																					
<p>2-1. Sujet coding</p> <p>Optimisation de distance Description Le célèbre acteur Tony Montana se déplace à New York. Il a beaucoup d'amis là-bas, tous vivent sur Star Avenue. Puisque, il rendra visite à tous ses amis, très souvent, il veut trouver une maison près d'eux. En effet, Tony veut minimiser la distance totale qui le sépare de tous ses amis et vous demande d'écrire un programme qui résout son problème.</p>	<p>2-2. Unit test cases</p> <table border="1" data-bbox="660 1585 1315 2069"> <thead> <tr> <th>Test name</th> <th>Candidate Output Value</th> <th>Correct output</th> <th>Execution time</th> <th>Memory consumption</th> <th>Result</th> </tr> </thead> <tbody> <tr> <td>Unit test1</td> <td>6</td> <td></td> <td>0.05 sec</td> <td>51011 KB</td> <td>Passed</td> </tr> <tr> <td>Unit test2</td> <td>4411924</td> <td></td> <td>0.24 sec</td> <td>51011 KB</td> <td>Passed</td> </tr> <tr> <td>Unit test3</td> <td>2347947</td> <td></td> <td>0.37 sec</td> <td>0 KB</td> <td>Passed</td> </tr> <tr> <td>Unit test4</td> <td>159927</td> <td></td> <td>0.15 sec</td> <td>0 KB</td> <td>Passed</td> </tr> <tr> <td>Unit test5</td> <td>6146449</td> <td></td> <td>0.40 sec</td> <td>51011 KB</td> <td>Passed</td> </tr> </tbody> </table> <p>2-3. Candidat Source code</p>	Test name	Candidate Output Value	Correct output	Execution time	Memory consumption	Result	Unit test1	6		0.05 sec	51011 KB	Passed	Unit test2	4411924		0.24 sec	51011 KB	Passed	Unit test3	2347947		0.37 sec	0 KB	Passed	Unit test4	159927		0.15 sec	0 KB	Passed	Unit test5	6146449		0.40 sec	51011 KB	Passed
Test name	Candidate Output Value	Correct output	Execution time	Memory consumption	Result																																
Unit test1	6		0.05 sec	51011 KB	Passed																																
Unit test2	4411924		0.24 sec	51011 KB	Passed																																
Unit test3	2347947		0.37 sec	0 KB	Passed																																
Unit test4	159927		0.15 sec	0 KB	Passed																																
Unit test5	6146449		0.40 sec	51011 KB	Passed																																

Spécification

Votre principale mission est de compléter la fonction suivante `int getMindistance (...)` qui retourne la somme minimale des distances depuis la maison optimale de Tonny vers chacun de ses amis (on ne demande pas de donner la position optimale de Tonny).

Le squelette du code fourni est composé de:

Distance.java // fichier de la fonction `getMindistance (...)`

Entrée

Fonction `getMindistance (...)` prendra en entrée un tableau contenant une série d'entier représentant les numéros de rue (des nombres entiers positifs) $s_1, s_2, \dots, s_i, \dots, s_r$ où les amis de Tonny vivent ($0 < s_i < 30\ 000$). Notez que plusieurs amis peuvent vivre dans le même numéro de la rue.

Sortie

Votre programme doit retourner un entier représentant la somme minimale des distances depuis la maison optimale de Tonny vers chacun de ses amis. La distance entre deux numéros de rue s_i et s_j est $D_{ij} = |s_i - s_j|$.

Exemple

Entrée: 1 2 5 4

Sortie: 6

Language : java

Compilation:Successfully

Marks Scored: 5/5

Main File: Distance.java

```
import java.util.Vector;

public class Distance {

    public static int getMindistance(Vector<Integer> rues) {
        // votre code ici

        int dist = 0;
        Vector<Integer> list_dist = new Vector<Integer>();
        for (int i = 0; i < rues.size(); i++)
        {
            dist = 0;
            for (int j = 0; j < rues.size(); j++)
                dist += Math.abs(rues.get(i) - rues.get(j));
            list_dist.add(dist);
        }
        int res = list_dist.get(0);
        for (int d : list_dist)
        {
            if (res > d)
                res = d;
        }
        return res;
    }
}
```