 **Hamdi BACCARA** (hamdi.baccara@softeam.fr)

Campagne : Java, Spring Framework - Prep

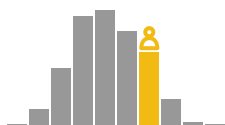
Domaine(s) : Java, Spring Framework

Langage : Français

Date : 20/03/2021

MEILLEUR QUE

87%
des professionnels



RANG

3 / 9



DURÉE

1h04 / 1h17



SCORE

1 300 / 1 980 (66%)

Java 1 160 / 1 540pts (75%)

MEILLEUR QUE **96%** des professionnels

Connaissance du langage



597 / 677pts

Fiabilité



158 / 208pts

Modélisation



130 / 130pts

Résolution de problèmes



275 / 525pts

Spring Framework 140 / 440pts (32%)

MEILLEUR QUE **18%** des professionnels

Connaissance du langage



140 / 440pts

Question 1: [Core] Beans



Spring Framework



00:19 / 00:30



20 / 20 pts



Question

Comment appelle-t-on les objets qui sont instanciés, managés et détruits par un conteneur IoC Spring ?



Réponse

Bean



Résultat



Réponse correcte

Connaissance du langage +20pts



Réponse(s) correcte(s)

- beans
- Spring beans
- bean
- Spring bean

Question 2: [Data] find by Id



Spring Framework



01:00 / 01:00



20 / 20 pts



Question

On considère une entité *User* et l'interface *UserRepository* associée, héritée de *CrudRepository*, qui permet son accès depuis la base.

```
public interface UserRepository extends CrudRepository<User, Long> {  
    User XXXXXX(Long id);  
}
```

Quelle méthode devriez-vous définir pour permettre de récupérer un objet *User* à partir de son id ?

Écrivez le nom de la méthode.



Réponse

getUserById



Résultat



Réponse correcte

Connaissance du langage +20pts



Réponse(s) correcte(s)

- `.*findUserById.*`
- `.*getUserById.*`
- `.*readUserById.*`
- `.*findById.*`

Question 3: [Rest] RestController



Spring Framework



00:15 / 00:30



20 / 20 pts



Question

Comment déclare-t-on un endpoint REST dans Spring ?



Réponse



@RestEndpoint



@RestController



@Rest



@REST



Résultat



Réponse correcte

Connaissance du langage +20pts

Question 4: [Core] Détection de bean



Spring Framework



00:33 / 00:45



20 / 20 pts



Question

Vous souhaitez que la classe suivante soit détectée par un scan Spring et enregistrée en tant que bean. Quelle annotation utiliseriez-vous ?

```
@XXX
public class LoggingServiceImpl implements LoggingService {
    /// implementation
}
```



Réponse



Component



Bean



Scannable



Register



Résultat



Réponse correcte

Connaissance du langage +20pts

Question 5: [Web] STOMP



Spring Framework



00:11 / 00:50



0 / 20 pts



Question

Quel est le nom du protocole, basé sur du texte, utilisé par Spring pour envoyer des messages par Websockets ?



Réponse

WS



Résultat



Réponse correcte

Connaissance du langage +20pts



Réponse(s) correcte(s)

- STOMP
- Streaming Text Oriented Messaging Protocol

Question 6: [Core] Injection



Spring Framework



00:08 / 00:45



20 / 20 pts



Question

Quelle annotation peut-on utiliser pour injecter un bean Spring ?



Réponse



@Autowired



@PostConstruct



@Bean



@Component



@InjectBean



Résultat



Réponse correcte
Connaissance du langage +20pts

Question 7: [Security] Contexte de sécurité



Spring Framework



00:21 / 01:15



0 / 40 pts



Question

Le code ci-dessous est supposé faire un appel à une méthode statique d'une classe communément utilisée de Spring Security (**XXX** dans l'image) pour récupérer le nom de l'utilisateur actuellement loggué.

Quelle est cette classe ?

```
String name = XXX.getContext().getAuthentication().getName();
```

Écrivez le nom de cette classe (sans le nom du package).



Réponse



Résultat



Réponse correcte

Connaissance du langage +40pts



Réponse(s) correcte(s)

- org.springframework.security.core.context.SecurityContextHolder
- SecurityContextHolder

Question 8: [Core] Bean conditionnel



Spring Framework



01:22 / 01:30



0 / 40 pts



Question

On considère les deux classes suivantes.

L'une qui définit une vérification de condition custom :

```
public class ModelExistsCondition implements Condition {  
    public boolean modelVariableExists(  
        ConditionContext context, AnnotatedTypeMetadata metadata) {  
        Environment env = context.getEnvironment();  
        return env.containsProperty("MODEL");  
    }  
}
```

et une autre qui définit un bean :

```
@Bean  
XXX  
public class ModelProcessor {  
    // implementation  
}
```

Vous souhaitez que le bean *ModelProcessor* soit créé seulement si la variable d'environnement *MODEL* existe.

En utilisant *ModelExistsCondition*, par quelle annotation remplacer *XXX* ci-dessus ?


Écrivez l'annotation complète.



Réponse

@Validate

Résultat

 Réponse correcte
Connaissance du langage ~~+40pts~~

Réponse(s) correcte(s)

- @Conditional(ModelExistsCondition.class)
- Conditional(ModelExistsCondition.class)
- @Conditional(ModelExistsCondition)
- Conditional(ModelExistsCondition)

Question 9: [Security] OAuth2



Spring Framework



00:40 / 00:40



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

Quelle annotation de Spring Boot permet d'activer le SSO (Single Sign On) avec un provider OAuth2 déclaré au niveau de propriétés externes ?

Cette annotation a été introduite par Spring Boot 2.0.0.



Réponse



@OAuth2ClientAutoConfiguration



@EnableOAuth2Client



@OAuth2Client



@EnableSso



Résultat



Réponse incorrecte

Connaissance du langage ~~+40pts~~

Question 10: [Core] Meilleure méthode d'injection



Spring Framework



01:15 / 01:15



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

Observez les deux façons d'injecter un service dans un bean :

```
@Component
public class MyService {

    private AnotherService anotherService;

    @Autowired
    public MyService(AnotherService anotherService) {
        this.anotherService = anotherService;
    }

    ///code
}
```

```
@Component
public class MyService {

    @Autowired
    private AnotherService anotherService;

    ///code

    public void setAnotherService(AnotherService anotherService) {
        this.anotherService = anotherService;
    }


}
```

Le résultat est le même pour les deux codes. Mais, y en a-t-il un qui est préférable à l'autre ?

Réponse

- ☐ Celui par setter est préférable car il autorise le développeur à ignorer l'injection ce qui permet de prototyper plus vite
- ☒ Celui par constructeur est préférable car il force le développeur à fournir le service ce qui réduit le risque de "null pointer exception"
- ☐ Non, il n'y a pas de différence en pratique
- ☐ Non, aucun des deux ne convient, l'injection par réflexion est la meilleure façon de procéder

Résultat

 Réponse incorrecte
Connaissance du langage ~~+40pts~~

Question 11: [Core] Constructeurs Autowired



Spring Framework



00:37 / 00:50



0 / 40 pts


Question

Combien de constructeurs d'une classe peuvent être annotés avec **@Autowired** ?

Réponse

- ☒ 0 - @Autowired n'est utilisable que sur les champs et les méthodes setXXX
- ☒ 1 - sinon Spring ne saurait lequel choisir
- ☐ multiple - si chacun d'eux a un @Qualifier différent
- ☐ multiple - dans tous les cas

Résultat

 Réponse incorrecte
Connaissance du langage ~~+40pts~~

Question 12: [Core] Unicité d'un bean



Spring Framework



00:32 / 01:00



0 / 40 pts



Question

Que se passe-t-il quand on demande à une **BeanFactory** de fournir une instance de bean pour lequel il existe de multiples candidats, alors qu'un seul bean candidat était attendu ?



Réponse

- ☒ Spring choisit le premier bean qu'il trouve dans le contexte scanné
- ☐ **NoUniqueBeanDefinitionException est lancé**
- ☐ Aucun bean n'est cablé (retourne null)
- ☐ Spring choisit le premier bean par ordre alphabétique



Résultat



Réponse incorrecte

Connaissance du langage ~~+40pts~~

Question 13: [Web] Requête GET



Spring Framework



00:30 / 01:00



0 / 40 pts



Question

On considère le code ci-dessous :

```
1 @Controller
2 public class OwnerController {
3
4     XXX
5     public Cart getCart(@PathVariable int cartId) {
6         ...
7     }
8 }
```

En utilisant Spring MVC, avec quelle annotation doit-on remplacer **XXX**, à la ligne 4, si l'on souhaite que la méthode **getCart()** soit exécutée suite à un appel de l'URL **/cart/{cartId}** par la méthode **GET** ?

Écrivez l'annotation complète.



Réponse

@Get



Résultat



Réponse correcte

Connaissance du langage ~~+40pts~~



Réponse(s) correcte(s)

- RequestMapping(value="/cart/{cartId}", method=RequestMethod.GET)
- @RequestMapping(value="/cart/{cartId}", method=RequestMethod.GET)
- RequestMapping(value = "/cart/{cartId}", method = RequestMethod.GET)
- @RequestMapping(value = "/cart/{cartId}", method = RequestMethod.GET)
- GetMapping("/cart/{cartId}")
- @GetMapping("/cart/{cartId}")

Question 14: [Data] Requêtes personnalisées



Spring Framework



00:14 / 00:45



40 / 40 pts



Question

Quelle est l'annotation Spring utilisée pour ajouter une méthode associée à une requête SQL personnalisée vers un repository JPA (XXXXXX dans le code ci-dessous) ?

```
@XXXXXX("select c from Car c where c.manufacturer = 'Ford'")  
List<Car> findAllFordCars();
```



Réponse



query



sql



autosql



select



Résultat



Réponse correcte

Connaissance du langage +40pts

Question 15: Blocs statiques



Java



00:19 / 00:30



20 / 20 pts



Question

Que devez-vous utiliser pour exécuter un code une seule fois, au moment où la classe est chargée ?



Réponse



Un bloc statique



Un bloc d'instance



Un constructeur



Une méthode



Résultat



Réponse correcte

Connaissance du langage +20pts

Question 16: Héritage multiple



Java



00:20 / 00:20



20 / 20 pts

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



Question

Une classe Java peut avoir plusieurs classes parentes.



Réponse



Vrai



Faux



Résultat



Réponse correcte
Connaissance du langage +20pts

Question 17: Thread



Java



00:07 / 00:20



20 / 20 pts



Question

Quelle méthode est appelée lorsqu'un thread est exécuté ?



Réponse

☐ do

☒ run

☐ exec

☐ execute

☐ play



Résultat



Réponse correcte

Connaissance du langage +20pts

Question 18: Classe « friendly »



Java



00:10 / 00:20



20 / 20 pts



Question

Déclarer une classe sans niveau d'accès explicite est en général conseillé.



Réponse



Vrai



Faux



Résultat



Réponse correcte

Modélisation +20pts

Question 19: Utilisation des exceptions



Java



00:21 / 00:30



20 / 20 pts



Question

Vous réalisez une bibliothèque en Java. Parmi ces options, laquelle privilégiez-vous pour traiter un comportement inattendu ?



Réponse

- ☒ `throw new ComportementInattenduException()`
- ☐ `System.exit(-1);`
- ☐ `System.err.println("Erreur: Comportement inattendu")`
- ☐ `return false`



Résultat



Réponse correcte
Modélisation +20pts

Question 20: Singleton



Java



00:09 / 00:20



20 / 20 pts



Question

Quelle signature de méthode est généralement utilisée pour obtenir un singleton ?



Réponse



getSingleton()



instance()



getInstance()



Résultat



Réponse correcte

Connaissance du langage +20pts

Question 21: Classe abstraite



Java



00:09 / 00:20



20 / 20 pts



Question

Une classe abstraite peut contenir des méthodes concrètes.



Réponse



Vrai



Faux



Résultat



Réponse correcte

Connaissance du langage +20pts

Question 22: java.util.Optional<T>



Java



00:28 / 00:30



20 / 20 pts



Question

Quelle classe permet de représenter la présence ou l'absence d'un objet sans qu'il soit nécessaire de vérifier si l'objet est *null* ou non ?



Réponse



java.util.Optional<T>



java.util Wrapper<T>



java.util.Maybe<T>



java.util.NotRequired<T>



java.util.Nullable<T>



Résultat



Réponse correcte

Connaissance du langage +20pts

Question 23: Opération sur des entiers



Java



00:21 / 00:30



20 / 20 pts



Question

```
int i1 = 5;
```

```
int i2 = 2;
```

```
int i3 = i1 / i2;
```

Quelle est la valeur de *i3* ?



Réponse

☐ 3

☐ 2.5

☒ 2

☐ NaN



Résultat



Réponse correcte

Connaissance du langage +20pts

Question 24: Accès d'une méthode synchronisée par deux threads



Java



00:24 / 00:40



0 / 40 pts



Question

Que se passe-t-il si une méthode synchronisée est appelée simultanément par deux threads sur des instances d'objet différentes ?



Réponse

- ☒ Seulement un thread à la fois accédera à la méthode
- ☐ Les deux threads pourront accéder à la méthode en même temps
- ☐ Aucun thread ne pourra accéder à la méthode et une exception sera générée



Résultat



Réponse incorrecte

Connaissance du langage +40pts

Question 25: Enum



Java



00:20 / 00:30



40 / 40 pts



Question

```
enum Planet {  
    MERCURY, VENUS, EARTH  
}
```

En considérant le code ci-dessus, quelles affirmations sont correctes ?



Réponse



Planet.MERCURY == Planet.MERCURY est vraie



Planet.MERCURY == Planet.VENUS est vraie



Planet.MERCURY.equals(Planet.MERCURY) est vraie



Résultat



Réponse correcte

Connaissance du langage +40pts

Question 26: Fonction Predicate



Java



00:35 / 00:50



40 / 40 pts



Question

En Java 8, quelle interface par défaut peut être utilisée comme une fonction à un argument qui retourne un booléen ?



Réponse



Optional



Predicate



Functions



Comparator



Résultat



Réponse correcte

Connaissance du langage +40pts

Question 27: Interface vs. implementation



Java



00:22 / 00:30



40 / 40 pts



Question

Parmi ces déclarations de méthodes, laquelle est à privilégier ?



Réponse

- ☐ `public ArrayList getOrders()`
- ☐ `public Vector getOrders()`
- ☒ `public List getOrders()`



Résultat



Réponse correcte

Modélisation +40pts

Question 28: Taille d'une ArrayList



Java



00:10 / 00:30



40 / 40 pts



Question

```
ArrayList l = new ArrayList(2);  
l.add(1);  
l.add(1);  
l.add(1);
```

Quelle est la valeur retournée par `l.size()` ?



Réponse



1



2



3



5



Ce code lève une exception



Résultat



Réponse correcte

Connaissance du langage +40pts

Question 29: Les interfaces



Java



00:09 / 00:20



40 / 40 pts



Question

En Java 8, les interfaces peuvent contenir des méthodes implémentées.



Réponse



Vrai



Faux



Résultat



Réponse correcte
Connaissance du langage +40pts

Question 30: Usage de ThreadLocal



Java



01:00 / 01:00



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

Quelle classe peut être utilisée pour stocker des valeurs spécifiques au thread ?

Ecrire seulement le nom de la classe.



Réponse



Résultat



Réponse correcte

Connaissance du langage ~~+40pts~~



Réponse(s) correcte(s)

- ThreadLocal
- .*ThreadLocal.*

Question 31: Référence de méthode



Java



00:25 / 01:00



40 / 40 pts



Question

Quel code ci-contre utilise une référence de méthode de manière correcte ?



Réponse



```
public class Main {
    public static void main(String[] args) {
        Stream.of(1, 2, 3)
            .forEach(Main::printNumber);
    }
    public static void printNumber(int n) {
        System.out.println(n);
    }
}
```



```
public class Main {
    public static void main(String[] args) {
        Stream.of(1, 2, 3)
            .forEach(Main.printNumber);
    }
    public static void printNumber(int n) {
        System.out.println(n);
    }
}
```




```
public class Main {
    public static void main(String[] args) {
        Stream.of(1, 2, 3)
            .forEach(Main#printNumber);
    }
    public static void printNumber(int n) {
        System.out.println(n);
    }
}
```






```
public class Main {
    public static void main(String[] args) {
        Stream.of(1, 2, 3)
            .forEach((n) -> printNumber(n));
    }
    public static void printNumber(int n) {
        System.out.println(n);
    }
}
```

Résultat

 Réponse correcte
Connaissance du langage +40pts

Question 32: Opérateur sur les bits : |

 Java  00:25 / 00:25  40 / 40 pts


Question

En base 2 (binaire), combien font **01** | **11** ?

Réponse

- ☐ 00
- ☐ 01
- ☐ 10
- ☒ 11
- ☐ 100

Résultat

 Réponse correcte
Connaissance du langage +40pts

Question 33: Héritage multiple d'interfaces



Java



00:30 / 00:30



40 / 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

`public interface A extends B, C, D {}`

Cette interface est correcte si B, C et D sont également des interfaces.



Réponse



Vrai



Faux



Résultat



Réponse correcte
Connaissance du langage +40pts

Question 34: String buffers



Java



00:15 / 00:30



60 / 60 pts



Question

Saisissez le nom d'une classe appartenant au package java.lang qui permet de concaténer efficacement les chaînes de caractères.



Réponse

StringBuilder



Résultat



Réponse correcte

Connaissance du langage +60pts



Réponse(s) correcte(s)

- StringBuilder
- StringBuffer
- java.lang.StringBuilder
- java.lang.StringBuffer

Question 35: Classification de paquets robotisée



Language independent



15:00 / 15:00



40 / 150 pts

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



Question

Objectif

Utiliser le bras robotique de l'usine pour trier les colis.

Règles

Vous travaillez pour une usine autonome. Votre objectif est de trier les colis qui arrivent sur le bon tas en fonction de leur volume et poids. Un colis est encombrant si son volume (Largeur x Hauteur x Profondeur) est supérieur ou égal à 1 000 000 cm³ ou si l'une de ses dimensions est supérieure ou égale à 150 cm. Un colis est lourd si sa masse est supérieure ou égale à 20kg. Vous devez répartir les colis dans 3 tas : STANDARD : les colis normaux (ni encombrants, ni lourds) seront traités normalement. SPECIAL : les colis lourds ou encombrants ne pourront pas être traités automatiquement. REJECTED : les colis à la fois encombrants et lourds seront refusés.

Implémentation

Implémentez la méthode ***solve(width, height, length, mass)*** (les unités sont le centimètre pour les dimensions et le kilogramme pour la masse). La méthode doit retourner une chaîne de caractères : le nom du tas où placer la boîte.

Conditions de Victoire

Les colis sont répartis sur les bons tas.

Conditions de Défaite

Votre programme indique une action invalide ou fausse.

Contraintes

$20 \leq \text{width, height, length} \leq 200$

$10 \leq \text{mass} \leq 1000$



Réponse

```
1 import java.util.*;
2 import java.io.*;
3 import java.math.*;
4
5 class Player {
6
7     public static String solve(int width, int height, int length, int mass) {
8         // Write your code here
9         // To debug: System.err.println("Debug messages...");
10
11         boolean volumeIsSpecial = false;
12         if (width * height * length >= 1000000 || width > 150 || height > 150 || length >
13             150){
14             volumeIsSpecial = true;
15         }
16         boolean massIsSpecial = false;
17         if (mass >= 20){
18             massIsSpecial = true;
19         }
20         if (volumeIsSpecial && massIsSpecial){
21             return "REJECTED";
22         }
23         if (volumeIsSpecial || massIsSpecial){
24             return "SPECIAL";
25         }
26         return "STANDARD";
27     }
28
29     /* Ignore and do not change the code below */
30     // #region main
31     public static void main(String args[]) {
32         Scanner in = new Scanner(System.in);
33
34         // game loop
35         while (true) {
36             int width = in.nextInt();
37             int height = in.nextInt();
38             int length = in.nextInt();
39             int mass = in.nextInt();
40             PrintStream outputStream = System.out;
41             System.setOut(System.err);
42             String action = solve(width, height, length, mass);
43             System.setOut(outputStream);
44             System.out.println(action);
45         }
46     }
47     // #endregion
48 }
```

Résultat

-  Nombreuses boites
Résolution de problèmes ~~+40pts~~
-  Seulement des boites STANDARD et REJECTED
Résolution de problèmes +40pts
-  Seulement des boites STANDARD et SPECIAL
Résolution de problèmes ~~+35pts~~
-  Test cas limites
Résolution de problèmes ~~+35pts~~

Question 36: Chaînes de caractères égales



Java



01:17 / 02:30



50 / 50 pts



Question

A.isFoo(String param) devrait retourner *true* si *param* est égal à la chaîne "*foo*", sinon elle devrait retourner *false*.

Implémentez *A.isFoo(String param)*



Réponse

```
1 // Java code below
2 class A {
3
4     static boolean isFoo(String param) {
5         return "foo".equals(param);
6     }
7
8 }
```



Résultat



Utilisation de equals plutôt que '=='
Connaissance du langage +17pts



Fonctionne si param est null
Fiabilité +33pts

Question 37: Encodage simple



Language independent



15:00 / 15:00



2x (19 sec)



60 / 150 pts

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



Question

Implémentez la méthode ***encode(plainText)*** qui renvoie un message encodé.

Elle reçoit un paramètre ***plainText***, qui est une chaîne de caractères par exemple ***aaaabcccaaa***.

Vous devez l'encoder en comptant les occurrences consécutives de chaque lettre, par exemple dans ***aaaabcccaaa***, il y a :

4 fois la lettre ***a*** puis 1 ***b*** puis 3 ***c*** puis 3 ***a***

Par conséquent, vous devez retourner la chaîne ***4a1b3c3a***.

Contraintes :

plainText est une chaîne de caractères en minuscule (entre a et z) ***plainText*** n'est jamais ***null*** et a une taille maximale de 15000 caractères

EXEMPLES :

PlainText

aabaa

CypherText

2a1b2a

PlainText

aaaabcccaaa

CypherText


4a1b3c3a




Réponse


```
1  import java.util.*;
2  import java.io.*;
3  import java.math.*;
4
5  class Solution {
6
7      public static String encode(String plainText) {
8          // Write your code here
9          // To debug: System.err.println("Debug messages...");
10         int i = 0;
11         int count = 0;
12         Character ci = null;
13         String cypher = "";
14         while (i < plainText.length()){
15             Character c = plainText.charAt(i);
16             if (ci == null){
17                 count++;
18                 ci = c;
19             }
20             else if (c != ci){
21                 cypher = "" + count + ci;
22                 count = 0;
23             }
24             else{
25                 count++;
26             }
27             i++;
28         }
29         cypher = "" + count + ci;
30         return cypher;
31     }
32
33     /* Ignore and do not change the code below */
34     // #region main
35     public static void main(String args[]) {
36         Scanner in = new Scanner(System.in);
37         String plainText = in.nextLine();
38         PrintStream outputStream = System.out;
39         System.setOut(System.err);
40         String encoded = encode(plainText);
41         System.setOut(outputStream);
42         System.out.println(encoded);
43     }
44     // #endregion
45 }
```


Résultat

 p
Résolution de problèmes +30pts

 2000y
Fiabilité +30pts

 abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz
Résolution de problèmes ~~+30pts~~

 10000 char => 3331 char
Résolution de problèmes ~~+30pts~~

 aaaabcccaaa
Résolution de problèmes ~~+30pts~~

Question 38: Héritage



Java



00:41 / 02:00



50 / 50 pts



Question

Complétez la réponse pour rendre le bout de code suivant valide:

```
A a = new A();  
A b = new B();
```



Réponse

```
1  /** class A */  
2  class A {  
3  
4  }  
5  
6  /** class B */  
7  class B extends A {  
8  
9  }
```



Résultat



B étend A

Modélisation +50pts

Question 39: Soldes d'été



Language independent



13:04 / 15:00



3x (1 min)



300 / 300 pts



Question

C'est bientôt les soldes d'été !

Vous travaillez pour un magasin qui souhaite offrir une réduction de **discount%** sur le produit le plus cher acheté par un client donné pendant la période des soldes.

Le responsable du magasin vous demande de développer la méthode **calculateTotalPrice()**.

Cette méthode :

prend en paramètres la liste de prix des produits achetés par le client et le pourcentage de réduction **discount**. retourne le prix de vente total (arrondi à l'entier inférieur si le total ne tombe pas rond).

Contraintes:

$0 \leq \text{discount} \leq 100$ $0 < \text{prix d'un produit} < 100000$ $0 < \text{nombre de produits} < 100$



Réponse

```
1  import java.util.*;
2  import java.io.*;
3  import java.math.*;
4
5  class Solution {
6
7      public static int calculateTotalPrice(int[] prices, int discount) {
8          // Write your code here
9          // To debug: System.err.println("Debug messages...");
10
11         int maxPrice = 0;
12         double totalPrice = 0;
13         for (int i = 0; i < prices.length; i++){
14             int currentPrice = prices[i];
15             if (currentPrice > maxPrice){
16                 maxPrice = currentPrice;
17             }
18             totalPrice += currentPrice;
19         }
20         totalPrice -= (double) maxPrice * discount / 100;
21
22         return (int) Math.floor(totalPrice);
23     }
24
25     /* Ignore and do not change the code below */
26     // #region main
27     public static void main(String args[]) {
28         Scanner in = new Scanner(System.in);
29         int discount = in.nextInt();
30         int n = in.nextInt();
31         int[] prices = new int[n];
32         for (int i = 0; i < n; i++) {
33             prices[i] = in.nextInt();
34         }
35         PrintStream outStream = System.out;
36         System.setOut(System.err);
37         int price = calculateTotalPrice(prices, discount);
38         System.setOut(outStream);
39         System.out.println(price);
40     }
41     // #endregion
42 }
```

Résultat

- ✓ Simple somme
Résolution de problèmes +35pts
- ✓ Bonne vente
Résolution de problèmes +35pts
- ✓ Gros discount
Résolution de problèmes +35pts
- ✓ Arrondi correct
Fiabilité +35pts
- ✓ Un produit gratuit
Résolution de problèmes +35pts
- ✓ Fin de solde
Résolution de problèmes +35pts
- ✓ Grosse vente
Résolution de problèmes +30pts
- ✓ Même prix
Fiabilité +30pts
- ✓ Un seul produit
Fiabilité +30pts

Question 40: Correction simple



Java



02:00 / 02:00



0 / 100 pts

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



Question

La méthode **sumRange** devrait retourner la somme des entiers compris entre 10 et 100 inclusifs contenus dans le tableau passé en paramètre.

Corrigez la méthode **sumRange**.

Note : le paramètre **ints** n'est jamais null.



Réponse

```
1 // Java code below
2 import java.util.*;
3 import java.io.*;
4 import java.nio.*;
5 import java.math.*;
6
7 class Solution {
8
9     static int sumRange(int[] ints) {
10         int sum = 0;
11         for (int i = 0; i < ints.length; i++) {
12             int n = ints[i];
13             if (n >= 10 || n <= 100) sum += n;
14         }
15         return sum;
16     }
17 }
```


> Résultat

✗ La solution fonctionne avec { 1, 4, 9, 12, 98, -10, 10 }
Résolution de problèmes ~~+50pts~~

Ⓝ La solution fonctionne avec { 30, 4, 9, 12, 98, -10, 10 }
Fiabilité ~~+50pts~~

Question 41: Synchronisation d'un compteur

⏮ Java ⌚ 01:23 / 02:00 ➡ 1x (4 sec) ⏭ 100 / 100 pts

? Question

Transformez la méthode **Counter.increment()** pour qu'elle supporte l'accès concurrentiel de plusieurs threads.

📝 Réponse

```
1 class Counter {  
2  
3     private static int count = 0;  
4  
5     /**  
6      * Increments count in a thread-safe manner.  
7      */  
8     public static synchronized int increment() {  
9         count = count + 1;  
10        return count;  
11    }  
12  
13 }
```

> Résultat

✓ Accès concurrentiel synchronisé
Connaissance du langage +100pts

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).