

## Belguith Nouha (nouha.belguith@ensi-uma.tn)

Campagne : Spring Framework, Java - Senior

Domaine(s) : Java, Spring Framework

Langage : Français

Date : 14/09/2022

MEILLEUR QUE

98%

des professionnels



RANG

1 / 2



DURÉE

0h11 / 1h28



SCORE

1 175 / 1 600 (73%)

**Java** 935 / 1 340pts (70%)

MEILLEUR QUE **97%** des professionnels

Connaissance du langage



300 / 300pts

Fiabilité



243 / 283pts

Résolution de problèmes



392 / 757pts

## Spring Framework

240 / 260pts (92%)

MEILLEUR QUE **93%** des professionnels

Connaissance du langage



140 / 160pts

Spring Boot



40 / 40pts

Spring Core



20 / 20pts

Spring Data



40 / 40pts

## Question 1: Visibilité des attributs privés



Java



00:04 / 00:20



20 / 20 pts



### Question

Un attribut privé est visible depuis les sous-classes.

---



### Réponse



Vrai



Faux

---



### Résultat



Réponse correcte

Connaissance du langage +20pts

## Question 2: Garbage collector



Java



00:04 / 00:20



40 / 40 pts



### Question

Le garbage collector garanti qu'il y a suffisamment de mémoire pour exécuter un programme Java.



### Réponse



Vrai



Faux



### Résultat



Réponse correcte

Connaissance du langage +40pts

## Question 3: Copie superficielle vs copie profonde



Java



00:04 / 02:00



40 / 40 pts



### Question

Considérer le code suivant :

```
public static void main(String[] args) throws CloneNotSupportedException {

    LinkedList<Player> players1 = new LinkedList<>();
    Player p1 = new Player();
    p1.setBattingAverage(46.7 f);
    p1.setWorldRank(4);

    Player p2 = new Player();
    p2.setBattingAverage(56.9 f);
    p2.setWorldRank(1);

    players1.add(p1);
    players1.add(p2);

    LinkedList<Player> players2 = new LinkedList<>();
    for (Player p : players1) {
        players2.add(p);
    }

    players2.get(0).setWorldRank(5);
}
```

Une liste de joueurs a été créée, ainsi qu'une copie de cette liste. Quel type de méthode de copie a été utilisée ?



### Réponse

☒ Copie superficielle (shallow copy)

☐ Copie profonde (deep copy)

## > Résultat

✓ Réponse correcte  
Connaissance du langage +40pts

### Question 4: Opérateur sur les bits : >>

⌕ Java ⌚ 00:04 / 00:20 🔄 40 / 40 pts

## ? Question

Quel est le résultat de l'opération `2 >> 1` ?

## 📝 Réponse

- ☐ 0
- ☒ 1
- ☐ 2
- ☐ 3
- ☐ 4

## > Résultat

✓ Réponse correcte  
Connaissance du langage +40pts

## Question 5: Tendre vers zéro



Java



04:30 / 15:00



260 / 300 pts



### Question

Implémentez la méthode `closestToZero` pour renvoyer l'entier du tableau `ints` le plus proche de zéro. S'il y a deux entiers tout aussi proches de zéro, considérez l'entier positif comme étant le plus proche de zéro (par exemple si `ints` contient -5 et 5, retournez 5). Si `ints` est `null` ou vide, retournez 0 (zero).

Données : les entiers dans `ints` ont des valeurs allant de -2147483647 à 2147483647.



### Réponse

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

## Résultat

- ✓ Les résultats sont corrects avec un jeu de données simple [7, 5, 9, 1, 4]  
Résolution de problèmes +120pts
- ✓ La solution fonctionne avec -2147483647 ou 2147483647  
Fiabilité +20pts
- ✓ La solution fonctionne quand le tableau ne contient que des entiers négatifs  
Fiabilité +20pts
- ✓ Quand 2 entiers sont aussi proches de 0, alors le positif l'emporte  
Fiabilité +20pts
- ✓ La solution fonctionne quand le tableau ne contient que deux entiers négatifs égaux  
Fiabilité +20pts
- ✓ La solution utilise `java.lang.Math.abs()`  
Connaissance du langage +60pts
- ✗ La solution fonctionne avec un tableau vide  
Fiabilité ~~+20pts~~
- ✗ La solution fonctionne avec un tableau null  
Fiabilité ~~+20pts~~

## Question 6: Du désordre le plus grand gagne



Java



00:51 / 05:00



100 / 100 pts

### ? Question

Implémentez la méthode `Algorithm.findLargest(int[] numbers)` afin qu'elle retourne le plus grand nombre dans `numbers`.

Note : Le tableau contient toujours au moins un nombre.

### 📝 Réponse

```
1 import java.util.Arrays;
2
3 class Algorithm {
4
5     /** @return the largest number of the given array */
6     static int findLargest(int[] numbers) {
7         Arrays.sort(numbers);
8         return numbers[numbers.length-1];
9     }
10
11 }
```

### > Résultat



Fonctionne dans des cas simples

Résolution de problèmes +32pts



Fonctionne quand le tableau contient seulement Integer.MIN\_VALUE

Fiabilité +58pts



Fonctionne quand le plus grand élément est à la position 0

Fiabilité +5pts



Fonctionne quand le plus grand élément est à la fin du tableau

Fiabilité +5pts



## Question 7: Les interfaces



Java



00:05 / 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 8: Expressions switch



Java



00:09 / 01:00



40 / 40 pts

### ? Question

Sélectionnez les codes contenant des expressions "switch" valides en Java12+.

Plusieurs réponses attendues.



## Réponse



```
int state = switch (month) {  
    case JANUARY -> 0;  
    case FEBRUARY -> 1;  
    case MARCH -> 2;  
    default -> {  
        int l = logic(month);  
        l *= 9;  
        yield l;  
    }  
};
```



```
int number = switch (state) {  
    case "UP":  
        yield 5;  
    case "DOWN":  
        yield 7;  
    default:  
        System.out.println("Illegal state");  
        yield -1;  
};
```



```
int number = switch (state) {  
    case "UP":  
        break 5;  
    case "DOWN":  
        break 7;  
    default:  
        System.out.println("Illegal state");  
        break -1;  
};
```



```
int state = switch (month) {  
    case JANUARY -> 0;  
    case FEBRUARY -> 1;  
    case MARCH -> 2;  
    default -> {  
        int l = logic(month);  
        l *= 9;  
        return l;  
    }  
};
```



```
int number = switch (state) {  
    case "UP":  
        return 5;  
    case "DOWN":  
        return 7;  
    default:  
        System.out.println("Illegal state");  
        return -1;  
};
```

---

## Résultat



Réponse correcte

Connaissance du langage +40pts

## Question 9: String.isBlank()



Java



00:08 / 01:00



20 / 20 pts

### ? Question

Quel est le nom de la méthode de la classe `java.lang.String` qui permet de tester si une chaîne de caractères est vide ou ne contient que des caractères d'espacement?

```
String str1 = "";           // yes
String str1 = " ";          // yes
String str1 = "Hello world."; // no
```

### ✎ Réponse

isBlank

### > Résultat



Réponse correcte

Connaissance du langage +20pts

### ✓ Réponse(s) correcte(s)

- `String.isBlank()`
- `.*isBlank.*`

## Question 10: Soldes d'été



Java



03:42 / 15:00



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. Un seul produit peut bénéficier de la réduction.

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




JAVA

```
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          Arrays.sort(prices);
9          double sum = prices[prices.length-1] - (prices[prices.length-1]*discount/100d);
10         for (int i = 0; i < prices.length-1; i++) {
11             sum +=prices[i];
12         }
13         return (int)sum;
14     }
15
16     /* Ignore and do not change the code below */
17     // #region main
18     public static void main(String args[]) {
19         Scanner in = new Scanner(System.in);
20         int discount = in.nextInt();
21         int n = in.nextInt();
22         int[] prices = new int[n];
23         for (int i = 0; i < n; i++) {
24             prices[i] = in.nextInt();
25         }
26         PrintStream outputStream = System.out;
27         System.setOut(System.err);
28         int price = calculateTotalPrice(prices, discount);
29         System.setOut(outputStream);
30         System.out.println(price);
31     }
32     // #endregion
33 }
```

## 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 11: Recherche d'image

 Java  00:07 / 40:00  35 / 400 pts

## Question

Objectif  
Trouver un motif dans une image.

## Règles

On vous donne 2 images : une image de base, et un motif qui peut être présent dans cette image. Chaque image est représentée sous la forme d'un tableau de chaînes de caractères, où chaque élément représente une ligne de pixels de l'image, et chaque caractère représente un pixel. Il n'est pas nécessaire de comprendre en détail cet encodage pour résoudre ce problème, mais pour information, des explications sont fournies plus bas. Vous devez retourner la position  $x, y$  de ce motif dans l'image, ou  $[-1, -1]$  si le motif n'est pas présent dans l'image. Si le motif apparaît plusieurs fois dans l'image, retournez la position du plus haut (le  $y$  le plus petit), et en cas d'égalité, le plus à gauche (le  $x$  le plus petit). La position du motif est déterminée par les coordonnées  $x, y$  de son coin en haut à gauche.  $x$  représente la colonne,  $y$  représente la ligne, et les coordonnées  $[0, 0]$  représentent le coin en haut à gauche.

## Implémentation

Implément la méthode `solve(imageWidth, imageHeight, image, patternWidth, patternHeight, pattern)` où les paramètres sont : `imageWidth`: la largeur de l'image `imageHeight`: la hauteur de l'image `image`: l'image, un tableau de strings, où chaque caractère représente un pixel `patternWidth`: la largeur du motif `patternHeight`: la hauteur du motif `pattern`: le motif, un tableau de strings, où chaque caractère représente un pixel et qui doit retourner : si le motif est présent dans l'image : la position  $x, y$ , sous la forme d'un tableau de 2 entiers, représentant le coin en haut à gauche du 1er motif, en partant du haut vers le bas, et de gauche à droite. si le motif n'est pas présent dans l'image :  $[-1, -1]$

## Conditions de Victoire

Le motif est effectivement situé aux coordonnées  $x, y$ . Si le motif apparaît plusieurs fois dans l'image, retournez la position du plus haut (le  $y$  le plus petit), et en cas d'égalité, le plus à gauche (le  $x$  le plus petit).

## Conditions de Défaite

La sous-image qui commence aux coordonnées  $x, y$  ne correspond pas au motif. Vous prenez plus d'une seconde à répondre. Il y a un autre motif qui correspond, plus haut ou plus à gauche que le votre.

## Contraintes

$1 \leq \text{imageWidth} \leq 400$

$1 \leq \text{imageHeight} \leq 400$

$1 \leq \text{patternWidth} \leq 400$

$1 \leq \text{patternHeight} \leq 400$

Temps de réponse  $\leq 1\text{s}$



## Détails sur l'encodage

Initialement, une image est un tableau d'entiers à 2 dimensions, où chaque entier représente un pixel (en utilisant la notation RGB). Toutes nos images possèdent au maximum 62 couleurs différentes, de manière à ce que nous puissions encoder chaque entier en un caractère parmi 0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ . Nous avons simplement transformé l'image en un tableau de strings, où la 1ère string est la 1ère ligne de l'image, etc. Par exemple, cette image 3x3 :



est représentée en tableau d'entiers à 2 dimensions comme :

```
[  
  [0xff0000, 0x00ff00, 0x0000ff],  
  [0xd3d3d3, 0x000000, 0xd3d3d3],  
  [0xff0000, 0x00ff00, 0x0000ff]  
]
```

qui est ensuite encodée en un tableau de strings en :

```
[  
  "012",  
  "343",  
  "012"  
]
```





# Réponse


JAVA


```
1 import java.util.*;
2 import java.io.*;
3 import java.math.*;
4
5 class Player {
6
7     public static int[] solve(int imageWidth, int imageHeight, String[] image, int
patternWidth, int patternHeight, String[] pattern) {
8         // Write your code here
9         // To debug: System.err.println("Debug messages...");
10
11         return new int[2];
12     }
13
14     /* Ignore and do not change the code below */
15     // #region main
16     public static void main(String args[]) {
17         Scanner in = new Scanner(System.in);
18         int imageWidth = in.nextInt();
19         int imageHeight = in.nextInt();
20         if (in.hasNextLine()) {
21             in.nextLine();
22         }
23         String[] image = new String[imageHeight];
24         for (int i = 0; i < imageHeight; i++) {
25             image[i] = in.nextLine();
26         }
27         int patternWidth = in.nextInt();
28         int patternHeight = in.nextInt();
29         if (in.hasNextLine()) {
30             in.nextLine();
31         }
32         String[] pattern = new String[patternHeight];
33         for (int i = 0; i < patternHeight; i++) {
34             pattern[i] = in.nextLine();
35         }
36         PrintStream outStream = System.out;
37         System.setOut(System.err);
38         int[] coordinates = solve(imageWidth, imageHeight, image, patternWidth,
patternHeight, pattern);
39         System.setOut(outStream);
40         for (int i = 0; i < 2; i++) {
41             System.out.println(coordinates[i]);
42         }
43     }
44     // #endregion
45 }
```


## Résultat


 4 pixels  
Résolution de problèmes ~~+35pts~~


 Flag  
Résolution de problèmes ~~+35pts~~

 Mario  
Résolution de problèmes ~~+35pts~~

 Not found  
Résolution de problèmes ~~+35pts~~


 V pour Vendetta  
Résolution de problèmes ~~+35pts~~


 Mario 2  
Résolution de problèmes ~~+35pts~~


 4 pixels 2  
Résolution de problèmes ~~+35pts~~

 Flag 2  
Résolution de problèmes +35pts

 Mario 3  
Résolution de problèmes ~~+30pts~~

 Bender Not found  
Résolution de problèmes ~~+30pts~~

 Jeton de poker  
Résolution de problèmes ~~+30pts~~

 Bender  
Résolution de problèmes ~~+30pts~~

## Question 12: [Core] Injection



Spring Framework



00:07 / 00:45



0 / 20 pts



### Question

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



### Réponse



@Autowired



@PostConstruct



@Bean



@Component



@InjectBean



### Résultat



Réponse incorrecte

Connaissance du langage ~~+20pts~~

## Question 13: [Boot] @SpringBootApplication



Spring Framework



00:05 / 00:45



20 / 20 pts



### Question

Quelle fonctionnalité de Spring Boot n'est pas activée par défaut lorsqu'une application est marquée par `@SpringBootApplication` ?



### Réponse

- ☐ Analyse des composants (component scan)
- ☐ Auto-configuration
- ☐ Injection de dépendances (dependency injection)
- ☒ **Tâches planifiées (scheduled tasks)**



### Résultat



Réponse correcte  
Spring Boot +20pts

## Question 14: [Data] @ManyToOne(mappedBy = "user")



Spring Framework



00:05 / 01:30



20 / 20 pts

### ? Question

Examinez la relation entre les deux entités ci-dessous :

```
@Entity
public class User {
    @XXX
    private Address address;

    // getters and setters ...
}

@Entity
public class Address {
    @ManyToOne
    private User user

    // getters and setters ...
}
```

Quelle annotation devrait remplacer @XXX pour que la relation soit bidirectionnelle ?

### ✎ Réponse



```
@OneToMany(mappedBy = "user")
```



```
@OneToMany(useParentId = true)
```



```
@OneToMany(mappingRelationshipOwner = Address.class)
```



```
@OneToMany(mappingRelationshipOwner = User.class)
```

## Résultat

 Réponse correcte  
Spring Data +20pts

### Question 15: [Core] Singleton par défaut

 Spring Framework  00:03 / 00:30  20 / 20 pts

## Question

Par défaut, combien d'instances d'un bean Spring trouveriez-vous dans le contexte d'application ?

---

## Réponse

- ☒ Une instance (singleton)
  - ☐ Une instance par classe qui le spécifie comme dépendance
  - ☐ Une instance par requête web
  - ☐ Une instance par thread
- 

## Résultat

 Réponse correcte  
Spring Core +20pts

## Question 16: [Core] Constructeurs Autowired



Spring Framework



00:10 / 00:50



40 / 40 pts



### Question

Quel énoncé est correct concernant l'annotation `@Autowired(required=true)` sur les constructeurs d'une classe ?



### Réponse

- ☐ `@Autowired` n'est utilisable que sur les champs et les méthodes setter
- ☒ `@Autowired` peut être ajouté sur un constructeur mais ce n'est pas nécessaire
- ☐ `@Autowired` peut être ajouté sur autant de constructeurs que souhaité



### Résultat



Réponse correcte

Connaissance du langage +40pts



## Question 17: [Core] Application events



Spring Framework



00:04 / 00:35



40 / 40 pts



### Question

Quel événement du cycle de vie de l'application n'existe pas dans Spring Framework ?



### Réponse

- ☐ ContextRefreshedEvent
- ☐ ContextStartedEvent
- ☐ ContextClosedEvent
- ☒ ContextLostEvent



### Résultat



Réponse correcte

Connaissance du langage +40pts

## Question 18: [Core] Beans



Spring Framework



00:11 / 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
- (?i).\*bean.\*

## Question 19: [Core] Lazy



Spring Framework



00:09 / 00:40



40 / 40 pts



### Question

Un bean est annoté avec `@Lazy`. Quand est-il initialisé ?

Plusieurs réponses attendues.



### Réponse

- ☐ Au démarrage du conteneur Spring
- ☒ Quand il est référencé par un autre bean
- ☒ Quand il est explicitement récupéré de la bean factory associée
- ☐ Quand il référence un autre bean



### Résultat



Réponse correcte

Connaissance du langage +40pts

## Question 20: [Boot] @SpringBootTest



Spring Framework



00:18 / 00:45



20 / 20 pts



### Question

Quel est l'intérêt de l'ajout de l'annotation `@SpringBootTest` à une classe de test ?



### Réponse

- ☐ Cela est nécessaire pour autoriser les tests unitaires sur des services simulés
- ☒ **Cela met en place un contexte d'application complet pour les tests d'intégration**
- ☐ Cela permet le démarrage de l'application si un cas de test échoue
- ☐ Cela démarre un Spring Boot REPL pour l'examen manuel de l'application



### Résultat



Réponse correcte  
Spring Boot +20pts

## Question 21: [Data] findAllByGivenName



Spring Framework



00:17 / 01:00



20 / 20 pts

### ? Question

Étant donné l'entité `User` et le `UserRepository` ci-dessous :

```
@Entity
public class User {

    @Id
    private Long id;
    private String givenName;

    // ... getters and setters
}
```

Quel nom de méthode écrire à la place de `XXX` dans le `UserRepository` ci-dessous pour utiliser les conventions de nommage de Spring Data et créer automatiquement une requête qui trouve tous les utilisateurs avec un `givenName` spécifié sans passer par une requête personnalisée ?

```
@Repository
public interface UserRepository extends JpaRepository<Long, User> {
    public Collection<User> XXX(String givenName);
}
```

### ✎ Réponse

findAllByGivenName

### > Résultat



Réponse correcte  
Spring Data +20pts

## Réponse(s) correcte(s)

- findAllByGivenName
- (?i).\*find(All)?ByGivenName.\*

# 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).