

! Afin d'éviter la fuite de questions, ce rapport ne doit pas être partagé avec les candidats.

## ⌚ consultant Nouha (nouha.belguithnb@gmail.com)

Campagne : Spring Framework, Java, MongoDB - Senior Domaine(s) : Java, MongoDB, Spring Framework Langage : Français Date : 12/09/2022

MEILLEUR QUE

**87%**

des professionnels



RANG

1 / 1

DURÉE

0h37 / 1h02

SCORE

880 / 1 120 (79%)

**Java** 660 / 700pts (94%)

MEILLEUR QUE >99% des professionnels

Connaissance du langage



200 / 200pts

Fiabilité



108 / 148pts

Résolution de problèmes



352 / 352pts

**MongoDB** 20 / 160pts (12%)

MEILLEUR QUE **8%** des professionnels

Data manipulation & queries



20 / 60pts

Data modeling & storage



0 / 100pts

# Spring Framework

200 / 260pts (77%)

MEILLEUR QUE **72%** des professionnels

Connaissance du langage



**100** / 160pts

Spring Boot



**40** / 40pts

Spring Core



**20** / 20pts

Spring Data



**40** / 40pts

## Question 1: Visibilité des attributs privés

 Java  00:10 / 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: Copie superficielle vs copie profonde

 Java



01:07 / 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 3: Les interfaces

 Java  00:07 / 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 4: Garbage collector

 Java  00:17 / 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 5: Expressions switch

 Java  00:25 / 01:00  1x (1 sec)  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 6: String.isBlank()

 Java

 00:19 / 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 7: Du désordre le plus grand gagne



Java



02:02 / 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.*;
2 class Algorithm {
3
4     /** @return the largest number of the given array */
5     static int findLargest(int[] numbers) {
6         Arrays.sort(numbers);
7         return numbers[numbers.length-1];
8     }
9
10 }
```

### 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 8: Rendu de monnaie

Java



24:21 / 40:00



20x (6 min)



360 / 400 pts

### Question

Les supermarchés s'équipent de plus en plus de caisses automatiques. La plupart de ces caisses n'acceptent que le paiement par carte bancaire bien qu'une part non négligeable de consommateurs paye encore en espèces (avec des billets et des pièces).

Une des problématiques rencontrées avec le paiement en espèces est le rendu de monnaie : comment rendre une somme donnée de façon optimale, c'est-à-dire avec le nombre minimal de pièces et billets ? C'est un problème qui se pose à chacun de nous quotidiennement, à fortiori aux caisses automatiques.

Dans cet exercice, on vous demande d'essayer de trouver une solution optimale pour rendre la monnaie dans un cas précis : quand une caisse automatique ne contient que des pièces de **2€**, des billets de **5€** et de **10€**.

Pour simplifier le problème, nous considérerons que toutes ces pièces et billets sont disponibles en **quantité illimitée**.

Voici quelques exemples de rendu de monnaie :

Monnaie à rendre Solutions possibles Solution optimale 1 Impossible Impossible 6 **2 + 2 + 2** **2 + 2 + 2** **10** **2 + 2 + 2 + 2** **5 + 5** **10** **10** 9223372036854775807 ... (**10** \* 922337203685477580) + **5 + 2**

Le rendu de monnaie est exprimé par un objet **Change**. Cet objet a 3 propriétés : **coin2**, **bill5** et **bill10** qui, respectivement, stockent le nombre de pièces de 2€, de billets de 5€ et de billets de 10€.

Par exemple, si on reprend l'exemple n°2 du tableau (6€), on devrait obtenir un objet **Change** avec : **coin2** vaut 3 (3 pièces de 2€) **bill5** vaut 0 (pas de billet de 5€) **bill10** vaut 0 (pas de billet de 10€). Implémentez la méthode **optimalChange(long s)** qui retourne un objet **Change** contenant les pièces et billets dont la somme vaut **s**. S'il est impossible de rendre la monnaie (comme dans l'exemple n°1), retournez **null**.

Pour obtenir un maximum de points votre solution devra toujours rendre la monnaie quand c'est possible et avec le nombre minimal de pièces et billets.

Données : **s** est toujours un entier (**long**) strictement positif inférieur ou égal à 9223372036854775807

 Réponse

```
1 import java.util.*;
2 import java.math.*;
3
4 // Do not modify Change
5 class Change {
6     long coin2 = 0;
7     long bill5 = 0;
8     long bill10 = 0;
9 }
10
11 class Solution {
12
13     static Change optimalChange(long s) {
14         long rest = 0;
15         if(s < 2) {
16             return null;
17         }
18         Change c = new Change();
19         c.bill10 = s / 10;
20         rest = s % 10;
21         if(rest < 5 && rest % 2 != 0){
22             c.bill10--;
23             rest = rest + 10;
24         }
25         c.bill5 = rest / 5;
26         rest = rest % 5;
27         if(rest != 0 && rest % 2 != 0){
28             if(c.bill5 > 0){
29                 c.bill5--;
30                 rest = rest + 5;
31             }
32         }
33         if(rest % 2 == 0){
34             c.coin2 = rest / 2;
35         }
36         return c;
37     }
38 }
39 }
```

## ▶ Résultat

✓ La monnaie est correcte pour une somme de 10€  
Résolution de problèmes +40pts

✓ La monnaie est optimale pour une somme de 10€ (1\*10)  
Résolution de problèmes +40pts

✓ `null` est retourné quand la somme vaut 1  
Fiabilité +40pts

✓ Le programme rend correctement la monnaie quand la somme vaut 31  
Résolution de problèmes +40pts

✓ La monnaie est optimale pour une somme de 31€ (2\*10 +5 + 3\*2)  
Résolution de problèmes +40pts

✗ `null` est retourné quand la somme vaut 3  
Fiabilité +40pts

✓ Le programme rend correctement la monnaie quand la somme vaut 8 (4 \* 2)  
Résolution de problèmes +40pts

✓ Résultat correct et dans les temps avec 9223372036854775802€  
Résolution de problèmes +40pts

✓ Résultat correct et dans les temps avec 9223372036854775803€  
Résolution de problèmes +40pts

✓ La monnaie est optimale avec 9223372036854775803€  
Résolution de problèmes +40pts

## Question 9: Sharding



MongoDB



00:40 / 00:40



1x (9 sec)



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 affirmation est vraie concernant le sharding dans MongoDB ?

## Réponse

- Chaque shard contient des données différentes selon la clé du shard décidée par l'utilisateur
- Chaque shard contient des données différentes selon la clé du shard décidée par le système et qui ne peut pas être modifiée
- Chaque nœud de shard contient une copie exacte des données pour assurer la redondance

## Résultat

✗ Réponse incorrecte  
Data modeling & storage -40pts

## Question 10: Replica sets

 MongoDB



00:28 / 00:30



0 / 40 pts

### Question

Quel énoncé est faux concernant les replica sets dans MongoDB ?

### Réponse

- Ils assurent la redondance des données
- Ils augmentent la disponibilité des données
- Ils augmentent les performances d'écriture de la base de données
- Ils empêchent l'apparition d'un point de défaillance unique dans le cluster

### Résultat

 Réponse incorrecte  
Data modeling & storage -40pts

## Question 11: Vérification de l'existence d'un champ



00:49 / 01:00



0 / 40 pts

### Question

Vous avez les documents suivants dans une collection :

```
[  
  {  
    "_id": 5,  
    "name": "Stiles",  
    "age": 22  
  },  
  {  
    "_id": 6,  
    "name": "Malia",  
    "age": 17,  
    "dob": ISODate("2003-09-01")  
  },  
  {  
    "_id": 7,  
    "name": "Alisson",  
    "age": 21,  
    "dob": ISODate("1999-09-02")  
  },  
  {  
    "_id": 8,  
    "name": "Kira",  
    "age": 24,  
    "dob": ISODate("1996-08-11")  
  }]
```

Vous voulez retourner les documents dans lesquels l'attribut `dob` est défini. Quelles requêtes pouvez-vous utiliser ? Sélectionnez toutes les bonnes réponses.

## Réponse



```
db.users.find( { dob: { $exists: true} } );
```



```
db.users.find( { dob: { $ne: null} } );
```



```
db.users.find( { dob: { $exists: 1} } );
```



```
db.users.find( { dob: { $exists: 0} } );
```

## Résultat



Réponse incorrecte

Data manipulation & queries -40pts

## Question 12: Requête avec une plage de dates

 MongoDB  00:47 / 01:00  20 / 20 pts

### Question

Vous avez les documents suivants dans une collection :

```
[  
  {  
    "_id": 5,  
    "name": "Stiles",  
    "age": 22  
  },  
  {  
    "_id": 6,  
    "name": "Malia",  
    "age": 17,  
    "dob": ISODate("2003-09-01")  
  },  
  {  
    "_id": 7,  
    "name": "Alisson",  
    "age": 21,  
    "dob": ISODate("1999-09-02")  
  },  
  {  
    "_id": 8,  
    "name": "Kira",  
    "age": 24,  
    "dob": ISODate("1996-08-11")  
  }  
]
```

Vous voulez récupérer les utilisateurs pour lesquels la date de naissance (DOB) se situe entre deux dates spécifiques. Quelle requête devez-vous utiliser ?

## Réponse



```
db.users.find({  
    dob: {  
        $gte: ISODate("2000-08-11"),  
        $lte: ISODate("2015-08-11")  
    }  
});
```



```
db.users.find({  
    dob: {  
        $after: ISODate("2000-08-11"),  
        $before: ISODate("2015-08-11")  
    }  
});
```



```
db.users.find({  
    dob: {  
        $between: [  
            ISODate("2000-08-11"), ISODate("2015-08-11")  
        ]  
    }  
});
```



```
db.users.find({  
    dob: {  
        $range: [  
            ISODate("2000-08-11"), ISODate("2015-08-11")  
        ]  
    }  
});
```

## Résultat

 Réponse correcte  
Data manipulation & queries +20pts

## Question 13: Nombre d'index



00:42 / 00:50



0 / 20 pts

## Question

Vous avez les documents suivants dans une collection :

```
[  
  {  
    "_id": 5,  
    "name": "Stiles",  
    "subjects": ["history", "math"]  
  },  
  {  
    "_id": 6,  
    "name": "Malia",  
    "age": 17,  
    "subjects": ["history", "science"],  
    "dob": ISODate("2003-09-01")  
  },  
  {  
    "_id": 7,  
    "name": "Alisson",  
    "age": 21,  
    "code": "0092830",  
    "dob": ISODate("1999-09-02")  
  }  
]
```

Supposons que vous ayez créé un seul index, sur le champ `code`. Combien d'index existent maintenant sur cette collection ?

Répondre par un nombre entier.

## Réponse

3

## Résultat

Réponse incorrecte  
Data modeling & storage -20pts

## Réponse(s) correcte(s)

• 2

### Question 14: [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 15: [Core] Singleton par défaut

 Spring Framework



00:07 / 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: [Boot] @SpringBootApplication

 Spring Framework



00:19 / 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 17: [Data] @ManyToOne(mappedBy = "user")

 Spring Framework

 01:14 / 01:30

 1x (4 sec)

 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 18: [Core] Lazy

 Spring Framework  00:40 / 00:40  20 / 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

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

Plusieurs réponses attendues.

## 📝 Réponse

- Au démarrage du container 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 partiellement correcte  
Connaissance du langage +20/40pts

## Question 19: [Core] Beans

 Spring Framework



00:16 / 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 20: [Core] Constructeurs Autowired

 Spring Framework



00:22 / 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 21: [Core] Application events

 Spring Framework



00:25 / 00:35



0 / 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 incorrecte  
Connaissance du langage **+40pts**

## Question 22: [Boot] @SpringBootTest

 Spring Framework



00:35 / 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 23: [Data] findAllByGivenName

 Spring Framework

 00:52 / 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).