

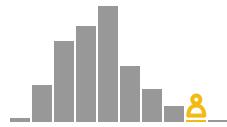
⚠ Afin d'éviter la fuite de questions, ce rapport ne doit pas être partagé avec les candidats.

👤 Belguith Nouha (nouha.belguith@expersi.com)

Campagne : Back-end - Java, Spring Framework, SQL - Senior Domaine(s) : Java, Language independent, Spring Framework Langage : Français
Date : 18/01/2024

MEILLEUR QUE

>99%
des professionnels



RANG

1 / 1

DURÉE

0h43 / 1h01

SCORE

1 424 / 1 700 (84%)

Java 1 057 / 1 110pts (95%)

MEILLEUR QUE >99% des professionnels

Connaissance du langage



647 / 700pts

Fiabilité



43 / 43pts

Modélisation



110 / 110pts

Résolution de problèmes



257 / 257pts

Language independent

40 / 250pts (16%)

MEILLEUR QUE 8% des professionnels

Fiabilité



0 / 30pts

Résolution de problèmes



40 / 220pts

Spring Framework

327 / 340pts (96%)

MEILLEUR QUE **>99%** des professionnels

Connaissance du langage



260 / 260pts

Spring Data



47 / 60pts

Spring Web



20 / 20pts

[Accéder au rapport détaillé](#)

Question 1: Visibilité des attributs privés



Java



00:05 / 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: Définition d'une fonction lambda

 Java



00:07 / 00:30



20 / 20 pts

Question

Comment définit-on une fonction lambda en Java ?

Réponse



`ExampleInterface test = () -> 9;`



`ExampleInterface test = () => 9;`



`ExampleInterface test = () #> 9;`



`ExampleInterface test = () >> 9;`

Résultat



Réponse correcte

Connaissance du langage +20pts

Question 3: java.lang.String.lines()



Java



00:50 / 00:50



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

Quelle méthode de `java.lang.String` utiliseriez-vous pour découper la chaîne ci-dessous en tableau ou collection de chaînes de caractère, le découpage étant fait vis-à-vis du caractère de saut de ligne ?

```
String str = "This is a string\nThis is the next line.\nHello world.";
```

Réponse

`split("/n")`

Résultat



Réponse correcte

Connaissance du langage +20pts

Réponse(s) correcte(s)

- `lines()`
- `split()`
- `.*lines.*`
- `.*split.*`

Question 4: Opération sur des entiers

 Java



00:06 / 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 5: 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 6: Client HTTP2 (java.net.http.*)

 Java

 00:04 / 00:40

 40 / 40 pts

Question

In Java 11+, quelle API est la plus appropriée pour faire des appels HTTP ?

Sélectionnez la meilleure réponse.

Réponse

- Les classes de `java.net.http.*` telle que `HttpClient`
- `java.net.HttpURLConnection`
- L'API d'une librairie tierce telle que celle d'Apache HttpClient

Résultat

 Réponse correcte
Connaissance du langage +40pts

Question 7: Copie superficielle vs copie profonde

 Java



00:28 / 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 8: Les interfaces

 Java  00:06 / 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 9: Annotation @FunctionalInterface

 Java

 00:08 / 00:30

 20 / 20 pts

Question

Quelle annotation peut être utilisée pour s'assurer qu'une interface est fonctionnelle ?

Réponse

- @Functional
- @lambda
- @FunctionalInterface
- @abstract

Résultat

 Réponse correcte
Connaissance du langage +20pts

Question 10: Usage de ThreadLocal

 Java

 00:34 / 01:00

 40 / 40 pts

Question

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

Ecrire seulement le nom de la classe.

Réponse

ThreadLocal

Résultat

 Réponse correcte
Connaissance du langage +40pts

Réponse(s) correcte(s)

- ThreadLocal
- (?i).*threadlocal.*

Question 11: Opérateur sur les bits : &

 Java

 00:04 / 00:20

 40 / 40 pts

Question

En base 2 (binaire), combien font `0001 & 0001` ?

Réponse

- `0010`
- `0000`
- `0001`

Résultat

 Réponse correcte
Connaissance du langage +40pts

Question 12: Méthode pour supprimer les espaces

 Java

 00:04 / 00:40

 40 / 40 pts

Question

En supposant que vous travaillez en Java 11+, quelle est la meilleure méthode pour supprimer les caractères d'espacement de début et de fin d'une chaîne de caractères ?

Example:

```
String str = "    Hello world.    ";
```

Réponse

- str.trim()
- str.strip()
- str.removeTrailingWhitespace()
- str.trimWhitespace()
- str.stripLeading().stripTrailing()

Résultat

 Réponse correcte
Connaissance du langage +40pts

Question 13: String buffers

 Java



00:25 / 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

StringBuffer

Résultat

 Réponse correcte
Connaissance du langage +60pts

Réponse(s) correcte(s)

- java.lang.StringBuilder
- java.lang.StringBuffer
- (?i).*StringBuilder.*
- (?i).*StringBuffer.*

Question 14: Méthode stream anyMatch()

 Java

 00:21 / 01:00

 40 / 40 pts

Question

Quelle méthode de `Stream` permet de vérifier si un certain prédicat correspond à l'un des éléments du stream ?

Ecrire le nom de la méthode la plus adaptée pour cela.

Réponse

`anyMatch()`

Résultat

 Réponse correcte
Connaissance du langage +40pts

Réponse(s) correcte(s)

- `anyMatch`
- `anyMatch()`
- `anyMatch(.*)`

Question 15: String.isBlank()

 Java

 00:16 / 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 = "";           // oui
String str1 = "      ";     // oui
String str1 = "Hello world.>"; // non
```

La méthode demandée est une méthode qui existe depuis Java version 11.

Réponse

`isBlank()`

Résultat

 Réponse correcte
Connaissance du langage +20pts

Réponse(s) correcte(s)

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

Question 16: Utilisation des exceptions

 Java



00:07 / 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 17: Design pattern 01

 Java



00:27 / 00:50



40 / 40 pts

Question

```
public class A {  
  
    private A() {}  
  
    private static class AHolder {  
        private static final A INSTANCE = new A();  
    }  
  
    public static A getInstance() {  
        return AHolder.INSTANCE;  
    }  
}
```

Si vous connaissez le patron de conception (design pattern) utilisé dans ce bout de code, saisissez son nom dans le champ texte (1 mot seulement).

Réponse

singleton

Résultat

 Réponse correcte
Modélisation +40pts

Réponse(s) correcte(s)

- Singleton
- (?i).*singleton.*

Question 18: Héritage - Utilisation du mot-clé "extends"

 Java  01:37 / 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  }
```

 Voir le code playback

Résultat

 B étend A
Modélisation +50pts

Question 19: Exception

Java



03:02 / 05:00



147 / 200 pts

Question

Mettez à jour le code en appliquant les règles suivantes : Si une exception est levée par `s.execute()` alors appeler `c.rollback()` et propager l'exception, sinon appeler `c.commit()`. Dans tous les cas, `c.close()` doit être appelée avant de quitter la méthode `a(Service s, Connection c)`

Réponse

```
1 class A {
2
3     /**
4      * Executes the service with the given connection.
5      */
6     void a(Service s, Connection c) {
7         // update this code
8         s.setConnection(c);
9         try{
10             s.execute();
11             c.commit();
12         }
13         catch(Exception e){
14             c.rollback();
15         }
16         finally {
17             c.close();
18         }
19     }
20 }
21
22 }
23
24 interface Service {
25     void execute() throws Exception;
26     void setConnection(Connection c);
27 }
28
29 interface Connection {
30     void commit();
31     void rollback();
32     void close();
33 }
```

▶ Voir le code playback

❯ Résultat

✓ Fonctionne correctement quand il n'y a pas d'exception
Connaissance du langage +56pts

✓ Utilisation de finally
Connaissance du langage +65pts

✓ Rollback et close quand il y a une exception
Connaissance du langage +26pts

✗ L'exception est propagée
Connaissance du langage +53pts

Question 20: Approximation de π

Java



10:21 / 12:00



300 / 300 pts

Question

Dans cet exercice nous allons calculer une estimation du nombre π (Pi).

La technique est la suivante :

On prend un point P au hasard de coordonnées (x, y) tel que $0 \leq x \leq 1$ et $0 \leq y \leq 1$. Si $x^2 + y^2 \leq 1$, alors le point est à l'intérieur du quart de disque de rayon 1, sinon le point est à l'extérieur.

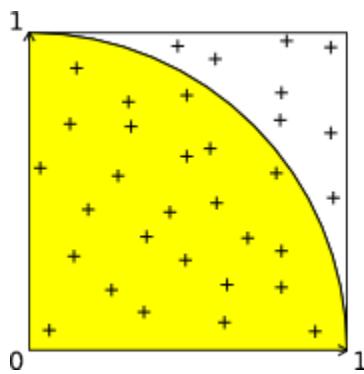


Fig 1. Exemple avec 33 points aléatoires.

On sait que la probabilité que le point se situe à l'intérieur du quart de disque est égale à $\pi/4$.

Écrivez la méthode `double approx(double[][] pts)` qui va utiliser les points `pts` (tirés au hasard) pour retourner une estimation du nombre π .

Données :

Chaque item de `pts` contient un point. Un point est représenté par un tableau contenant exactement deux nombres, respectivement `x` et `y` tels que $0 \leq x \leq 1$ et $0 \leq y \leq 1$. `pts` n'est jamais null et contient toujours au moins un 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         int hits = 0;
11         double pi = 0.0;
12         for (int i = 0; i < pts.length; i++) {
13             double x = pts[i][0];
14             double y = pts[i][1];
15             if((x*x) + (y*y) <= 1){
16                 hits++;
17             }
18         }
19         pi = (4.0 * hits) / pts.length;
20         return pi;
21     }
22 }
```

 [Voir le code playback](#)

Résultat

 L'estimation de π est valide (liée aux points fournis)
Résolution de problèmes [+257pts](#)

 Le point P(1, 0) est à l'intérieur du quart de disque
Fiabilité [+43pts](#)

Question 21: Du désordre le plus grand gagne



Language independent



05:00 / 05:00



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

Implémentez la méthode `findLargest(numbers)`, afin qu'elle renvoie le plus grand nombre du tableau `numbers`.

Remarques :

`numbers` contient uniquement des entiers. `numbers` contient toujours au moins un élément et ne contient jamais plus de 10 éléments. Les entiers sont compris entre `-2^31` et `+2^31 - 1` (ils tiennent toujours dans un entier signé de 32 bits).

Implémentation

Fonction

Implémenter la méthode `findLargest`.

Paramètres

`numbers` (`List<Integer>`) : Des nombres entiers.

Valeur de retour

`largest` (`int`) : La plus grande valeur parmi les nombres donnés en paramètres.

Contraintes

`1 <= length(numbers) <= 10`

`-2 147 483 648 <= numbers[] <= 2 147 483 647`

Mémoire RAM disponible : 512 Mo

Durée maximum d'exécution : 1 second



Réponse

JAVA

```
1 import java.util.*;
2 import com.google.gson.Gson;
3 import com.google.gson.GsonBuilder;
4 import com.google.gson.reflect.TypeToken;
5
6 class Solution {
7
8     /**
9      * @param numbers Some integer numbers.
10     * @return The largest value among the numbers given in parameters.
11     */
12    public static int findLargest(List<Integer> numbers) {
13        return numbers.stream().sorted((o1, o2) -> ).findFirst().get();
14    }
15
16    /* Ignore and do not change the code below */
17    private static final Gson gson = new GsonBuilder().disableHtmlEscaping().create();
18
19    /**
20     * Try a solution
21     * @param largest The largest value among the numbers given in parameters.
22     */
23    public static void trySolution(int largest) {
24        System.out.println(" " + gson.toJson(largest));
25    }
26
27    public static void main(String args[]) {
28        try (Scanner in = new Scanner(System.in)) {
29            trySolution(findLargest(
30                gson.fromJson(in.nextLine(), new TypeToken<List<Integer>>(){}.getType()
31            )));
32        }
33    }
34    /* Ignore and do not change the code above */
35 }
```

▶ Voir le code playback

▶ Résultat

✗ Valideur simple

Résolution de problèmes ~~+15pts~~

✗ Nombres positifs et négatifs

Résolution de problèmes ~~+10pts~~

✗ Le plus grand nombre est en première position dans le tableau

Résolution de problèmes ~~+15pts~~

✗ Le plus grand nombre est en dernière position dans le tableau, et il y a des doublons

Résolution de problèmes ~~+15pts~~

✗ Que des nombres négatifs

Fiabilité ~~+15pts~~

✗ Un seul nombre, le plus bas possible

Fiabilité ~~+15pts~~

✗ Valideur additionnel

Résolution de problèmes ~~+15pts~~

Question 22: Classification de paquets robotisée

 Language independent

 14:24 / 15:00

 1x (7 sec)

 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

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

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



Réponse

JAVA

```
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         if(!isEncombrant(width, height, length) && !isMass(mass)){
9             return "STANDARD";
10        }
11        else if(isEncombrant(width, height, length) && isMass(mass)){
12            return "REJECTED";
13        }
14        else {
15            return "SPECIAL";
16        }
17    }
18
19    public static boolean isMass(int mass) {
20        return mass >= 20 ;
21    }
22    public static boolean isEncombrant (int width, int height, int length) {
23        return width*height*length >= 1000000 || width >= 150 || length >= 150 ;
24    }
25
26
27    /* Ignore and do not change the code below */
28    public static void main(String args[]){
29        Scanner in = new Scanner(System.in);
30
31        // game loop
32        while (true) {
33            int width = in.nextInt();
34            int height = in.nextInt();
35            int length = in.nextInt();
36            int mass = in.nextInt();
37            PrintStream outStream = System.out;
38            System.setOut(System.err);
39            String action = solve(width, height, length, mass);
40            System.setOut(outStream);
41            System.out.println(action);
42        }
43    }
44    /* Ignore and do not change the code above */
45 }
```

▶ Voir le code playback

❯ 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 23: [Web] Cycle de vie des requêtes

 Spring Framework  00:05 / 00:30  20 / 20 pts

❓ Question

Lorsqu'un objet requête quitte le client, son premier arrêt lorsqu'il entre dans une application Spring est :

📝 Réponse

- ViewResolver
- DispatcherServlet
- La page JSP index
- Le premier Spring Controller qu'il peut trouver

❯ Résultat

- Réponse correcte
Spring Web +20pts

Question 24: [Validation] Validator

 Spring Framework

 00:14 / 01:15

 20 / 20 pts

Question

Quelle est le nom de l'interface Spring implémentée ci-dessous, utilisée pour valider un business model depuis plusieurs modules Spring ?

```
public class UserValidator implements XXXXXX {  
  
    public boolean supports(Class clazz) {  
        return User.class.equals(clazz);  
    }  
  
    public void validate(Object obj, Errors e) {  
        ValidationUtils.rejectIfEmpty(e, "name", "name.empty");  
        User p = (User) obj;  
        if (user.getAge() < 0) {  
            e.rejectValue("age", "negative");  
        }  
    }  
}
```

Réponse

Validator

Résultat

 Réponse correcte
Connaissance du langage +20pts

Réponse(s) correcte(s)

- Validator
- org.springframework.validation.Validator

Question 25: [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 26: [Core] Application events

 Spring Framework  00:06 / 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 27: [Core] Beans

 Spring Framework



00:05 / 00:30



20 / 20 pts

Question

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

Réponse

beans

Résultat

 Réponse correcte
Connaissance du langage +20pts

Réponse(s) correcte(s)

- beans
- Spring beans
- (?i).*bean.*

Question 28: [Core] Constructeurs Autowired

 Spring Framework



00:06 / 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 29: [Core] Lazy

 Spring Framework



00:21 / 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 30: [Rest] RestTemplate

 Spring Framework



00:16 / 00:45



40 / 40 pts

Question

Quelle classe client ReST Spring fournit-elle pour faire des appels aux API externes ?

Réponse

RestTemplate

Résultat

 Réponse correcte
Connaissance du langage +40pts

Réponse(s) correcte(s)

- RestTemplate
- (?).*RestTemplate.*
- (?).*WebClient.*

Question 31: [AOP] Pointcut

 Spring Framework



00:19 / 01:00



40 / 40 pts

Question

Quelle méthodes vont être interceptées par ce pointcut ?

```
@Pointcut("execution(public * find*(..))")  
private void someOperation() {}
```

Plusieurs réponses attendues.

Réponse

- public String findName()
- public char[] findPassword(boolean cache)
- protected String findAddress()
- public void createOrFindAccount(String name, char[] pass)

Résultat

 Réponse correcte
Connaissance du langage +40pts

Question 32: [Data] Avantages de HQL



Spring Framework



00:45 / 00:45



7 / 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

Quels sont les avantages de l'utilisation du langage de requête Hibernate (HQL) par rapport au langage de requête natif dans un référentiel Spring Data ?

Réponse

- Des requêtes plus courtes avec l'option d'omettre des déclarations comme `SELECT *` et la partie `ON` des déclarations de jointure
- Validation des requêtes au démarrage de l'application
- Migration transparente vers un autre langage de base de données relationnelle
- Interopérabilité entre les bases de données SQL et NoSQL

Résultat

Réponse partiellement correcte
Spring Data +7/20pts

Question 33: [Data] findAllByGivenName

 Spring Framework

 00:21 / 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.*

Question 34: [Web] STOMP



Spring Framework



00:29 / 00:50



20 / 20 pts

Question

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

Réponse

STOMP

Résultat



Réponse correcte

Connaissance du langage +20pts

Réponse(s) correcte(s)

- STOMP
- Streaming Text Oriented Messaging Protocol

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).