

Belguith Nouha (nouha.belguith@expersi.com)

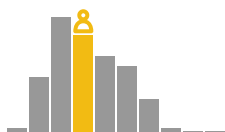
Campagne : Back-end - Java, Spring Framework, SQL - Expert **Domaine(s) :** Java, Language independent, Spring Framework **Langage :** Français

Date : 18/01/2024

MEILLEUR QUE

58%

des professionnels



RANG

1 / 1



DURÉE

0h55 / 1h03



SCORE

620 / 1 600 (39%)

Java 380 / 980pts (39%)

MEILLEUR QUE **61%** des professionnels

Connaissance du langage



340 / 540pts

Fiabilité



0 / 85pts

Modélisation



40 / 40pts

Résolution de problèmes



0 / 315pts

Language independent

0 / 300pts (0%)

MEILLEUR QUE **<1%** des professionnels

Fiabilité



0 / 180pts

Résolution de problèmes



0 / 120pts

Spring Framework

240 / 320pts (75%)

MEILLEUR QUE **>99%** des professionnels

Connaissance du langage



160 / 240pts

Spring Data



40 / 40pts

Spring Web



40 / 40pts

[Accéder au rapport détaillé](#)

Question 1: Opération sur des entiers



Java



00:16 / 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 2: Méthode run() et thread



Java



00:05 / 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 3: Visibilité des attributs privés



Java



00:06 / 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 4: Définition d'une fonction lambda



Java



00:30 / 00:30



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

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 5: Opérateur sur les bits : &



Java



00:20 / 00:20



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

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



Réponse



0010



0000



0001



Résultat



Réponse incorrecte

Connaissance du langage ~~+40pts~~

Question 6: Accès d'une méthode synchronisée par deux threads



Java



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

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 7: Client HTTP2 (java.net.http.*)



Java



00:31 / 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 8: 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 incorrecte

Connaissance du langage ~~+40pts~~



Réponse(s) correcte(s)

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

Question 9: Annotation @FunctionalInterface



Java



00:30 / 00:30



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



Réponse



@Functional



@lambda



@FunctionalInterface



@abstract



Résultat



Réponse incorrecte

Connaissance du langage +20pts

Question 10: Garbage collector



Java



00:20 / 00:20



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

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 11: Opérateur sur les bits : >>



Java



00:17 / 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 12: Méthode stream anyMatch()



Java



00:46 / 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 13: Méthode pour supprimer les espaces



Java



00:40 / 00:40



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

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

Exemple:

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



Réponse



`str.trim()`



`str.strip()`



`str.removeTrailingWhitespace()`



`str.trimWhitespace()`



`str.stripLeading().stripTrailing()`



Résultat



Réponse correcte

Connaissance du langage +40pts

Question 14: String buffers



Java



00:30 / 00:30



0 / 60 pts



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



Question

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



Réponse



Résultat



Réponse incorrecte

Connaissance du langage ~~+60pts~~



Réponse(s) correcte(s)

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

Question 15: Expressions switch



Java



00:42 / 02:30



60 / 60 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 +60pts

Question 16: Utilisation des exceptions



Java



00:15 / 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: Héritage



Java



00:17 / 01:00



20 / 20 pts



Question

Parmi ces deux propositions, laquelle préférez-vous ?

Proposition #1 :

```
interface CapableDeVoler {  
    void vole();  
}  
abstract class Avion implements CapableDeVoler {}  
abstract class Oiseau implements CapableDeVoler {}
```

Proposition #2 :

```
abstract class Avion {  
    abstract void vole();  
}  
abstract class Oiseau extends Avion {}
```



Réponse



La proposition #1



La proposition #2



Résultat



Réponse correcte

Modélisation +20pts

Question 18: Fréquences des mots



Java



10:00 / 10:00



1x (3 sec)



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

Objectif

Dans les tâches de traitement de texte, l'une des premières choses à faire est de déterminer la fréquence d'apparition de chaque mot dans un document donné. Dans cette tâche, vous allez compléter une fonction qui renvoie les fréquences uniques des mots d'un document de mots tokenisés.

Implémentation

Implémentez la méthode `countFrequencies(words)` qui prend comme entrée un tableau de chaînes de caractères (`words`), représentant un document de mots tokenisés. Par exemple :

```
['the', 'dog', 'got', 'the', 'bone']
```

Votre méthode `countFrequencies` doit retourner un tableau d'entiers contenant le nombre d'occurrences de chaque mot triés par ordre alphabétique.

Pour l'exemple ci-dessus, la sortie correcte serait :

```
[1, 1, 1, 2] # bone = 1, dog = 1, got = 1, the = 2
```



Réponse

```
1 import java.util.*;
2 import java.io.*;
3 import java.math.*;
4
5 class Solution {
6
7     public static int[] countFrequencies(String[] words) {
8         // Write your code here
9         // To debug: System.err.println("Debug messages...");
10        Map<String, Integer> map = new HashMap<>();
11        for(int i= 0; i<words.length; i++) {
12            map.put(words[i], map.getDefault(words[i], 0)+1);
13        }
14        for(Map.Entry<String, Integer> entery : map)
15            return new int[x];
16    }
17
18    /* Ignore and do not change the code below */
19    public static void main(String args[]) {
20        Scanner in = new Scanner(System.in);
21        int n = in.nextInt();
22        int x = in.nextInt();
23        if (in.hasNextLine()) {
24            in.nextLine();
25        }
26        String[] words = new String[n];
27        for (int i = 0; i < n; i++) {
28            words[i] = in.nextLine();
29        }
30        PrintStream outStream = System.out;
31        System.setOut(System.err);
32        int[] counts = countFrequencies(words);
33        System.setOut(outStream);
34        for (int i = 0; i < x; i++) {
35            System.out.println(counts[i]);
36        }
37    }
38    /* Ignore and do not change the code above */
39 }
```

▶ Voir le code playback

> Résultat

✖ 9 mots
Résolution de problèmes ~~+25pts~~

✖ 7 mots
Résolution de problèmes ~~+25pts~~

✖ 6 mots
Résolution de problèmes ~~+25pts~~

✖ 186 mots
Fiabilité ~~+25pts~~

Question 19: L'art ASCII



Java



08:29 / 10:00



1x (1 sec)



0 / 300 pts

? Question

`AsciiArt.printChar` permet d'afficher un et un seul caractère ASCII de A à Z (inclusif) sur plusieurs lignes et colonnes.

Maintenant, on souhaite faire l'opération dans l'autre sens : obtenir un caractère à partir de sa représentation graphique.

Implémentez la méthode `scanChar(String s)` afin qu'elle retourne le caractère associé à la représentation graphique fournie par `AsciiArt.printChar(char c)`. Si `s` ne correspond pas à un caractère entre A et Z (inclusif), alors `scanChar` devra retourner le caractère `?`.



Réponse

```
1 import java.util.*;
2 import external.*; // imports the AsciiArt class
3
4 class A {
5
6     /**
7      * @return the char that is represented by s
8      */
9     static char scanChar(String s) {
10         for(char c = 'a' ; c <= 'z' ; c++){
11             if(printChar(c).equals(s)){
12                 return c;
13             }
14         }
15         return (char) 0;
16     }
17     static String printChar(char c){
18         return "S";
19     }
20 }
```

▶ Voir le code playback



Résultat



Retourne 'A' si et seulement si la chaîne représente un A
Résolution de problèmes ~~+60pts~~



La solution fonctionne pour tout caractère entre A et Y
Résolution de problèmes ~~+180pts~~



Le programme retourne '?' si aucun caractère ne correspond à s
Fiabilité ~~+30pts~~



La solution fonctionne avec 'Z'
Fiabilité ~~+30pts~~

Question 20: Trouver le plus grand profit



Language independent



19:46 / 20:00



0 / 300 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 Dans cette tâche, on vous donne une liste d'entiers, qui représentent les bénéfices nets mensuels d'une entreprise. L'entreprise souhaite que vous trouviez la plage de mois consécutifs qui a enregistré le plus de bénéfices.

Implémentation

Fonction

Implémenter la méthode **maxProfit**.

Paramètres

data (**List<Integer>**) : une liste de nombres entiers, représentant une suite de bénéfices mensuels consécutifs

Valeur de retour

months (**List<Integer>**) : une liste de deux entiers contenant les indices des mois délimitant la plage avec le profit maximum

Contraintes

Mémoire RAM disponible : 512 Mo

Durée maximum d'exécution : 1 second




Réponse

JAVA


```
1  import java.util.*;
2  import java.util.stream.Collectors;
3
4  import com.google.gson.Gson;
5  import com.google.gson.GsonBuilder;
6  import com.google.gson.reflect.TypeToken;
7
8  class Solution {
9
10     public static List<Integer> maxProfit(List<Integer> data) {
11         // Write your code here
12         Map<Integer, Integer> map = new HashMap<>();
13         data.forEach(t -> map.put(t, data.indexOf(t)));
14         Map<Integer, Integer> treeMap = map.entrySet().stream()
15             .sorted(Map.Entry.comparingByKey(Comparator.reverseOrder()))
16             .collect(TreeMap::new, (m,e) -> m.put(e.getKey(), e.getValue()));
17         return treeMap.entrySet().stream().limit(2).map(Map.Entry::getValue)
18             .collect(Collectors.toList());
19     }
20
21     /* Ignore and do not change the code below */
22     private static final Gson gson = new GsonBuilder().disableHtmlEscaping().create();
23
24     /**
25      * Try a solution
26      */
27     public static void trySolution(List<Integer> months) {
28         System.out.println("'" + gson.toJson(months));
29     }
30
31     public static void main(String args[]) {
32         try (Scanner in = new Scanner(System.in)) {
33             trySolution(maxProfit(
34                 gson.fromJson(in.nextLine(), new TypeToken<List<Integer>>(){}).getType())
35             ));
36         }
37     }
38     /* Ignore and do not change the code above */
39 }
```

▶ Voir le code playback

Résultat

 20 entiers
Résolution de problèmes ~~+60pts~~

 100 entiers
Résolution de problèmes ~~+60pts~~

 500 entiers
Fiabilité ~~+60pts~~

 1000 entiers
Fiabilité ~~+60pts~~

 2000 entiers
Fiabilité ~~+60pts~~

Question 21: [Web] Cycle de vie des requêtes



Spring Framework



00:30 / 00:30



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

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 22: [Data] find by Id



Spring Framework



00:53 / 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 XXXXX(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

findById

> Résultat



Réponse correcte

Connaissance du langage +20pts

✓ Réponse(s) correcte(s)

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

Question 23: [Data] @ManyToOne(mappedBy = "user")



Spring Framework



01:12 / 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 24: [Validation] Validator



Spring Framework



00:32 / 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: [Core] Beans



Spring Framework



00:07 / 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 26: [Core] Constructeurs Autowired



Spring Framework



00:50 / 00:50



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

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 27: [Core] Application events



Spring Framework



00:25 / 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 28: [AOP] Pointcut



Spring Framework



00:33 / 01:00



20 / 40 pts



Question

Quelles 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 partiellement correcte

Connaissance du langage +20/40pts

Question 29: [Data] findAllByGivenName



Spring Framework



01:00 / 01:00



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

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

findByGivenName

> Résultat



Réponse correcte
Spring Data +20pts

✓ Réponse(s) correcte(s)

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

Question 30: [Web] STOMP



Spring Framework



00:50 / 00:50



0 / 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

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

✎ Réponse

> Résultat



Réponse incorrecte
Connaissance du langage +20pts

✓ Réponse(s) correcte(s)

- STOMP
- Streaming Text Oriented Messaging Protocol

Question 31: [Core] Bean conditionnel



Spring Framework



01:30 / 01:30



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

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

Résultat

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

Réponse(s) correcte(s)

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

Question 32: [Web] PathVariable



Spring Framework



00:51 / 01:00



20 / 20 pts



Question

Avec Spring, un endpoint est défini comme suit :

```
@RequestMapping("/hello/{type}")
```

L'attribut `type` dans l'URL ci-dessus peut être utilisé comme suit :

```
public void getType(@XXX("type") String type) { ... }
```

Quelle annotation devriez-vous écrire à la place de `xxx` ?



Réponse

PathVariable



Résultat



Réponse correcte
Spring Web +20pts



Réponse(s) correcte(s)

- PathVariable
- (?:i).*PathVariable.*

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).