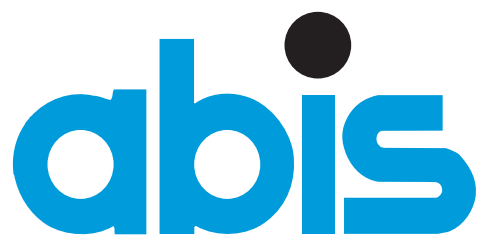


# Auto-évaluation Programmation en Java

Document: f0883test.fm

3 septembre 2019

ABIS Training & Consulting  
Diestsevest 32 / 4b  
B-3000 Leuven  
Belgium



---

TRAINING & CONSULTING

# INTRODUCTION AUTO-ÉVALUATION PROGRAMMATION EN JAVA

Ce test comporte 21 questions à choix multiples. Pour la plupart des questions, seulement une réponse correcte est possible. Il sera indiqué clairement si plusieurs réponses correctes sont possibles. Cochez les solutions qui vous semblent bonnes, et comparez vos réponses avec les solutions (la réponse n'est bonne que si toutes les réponses correctes sont sélectionnées).

Il n'y a pas de "question piège", cependant, lisez attentivement les questions et les propositions. Comptez environ 30 minutes pour répondre à toutes les questions.

Vous trouverez les réponses aux questions ainsi que la grille d'évaluation à la fin de ce document.

# QUESTIONS AUTO-ÉVALUATION PROGRAMMATION EN JAVA

1. Laquelle de ces commandes relatives aux tableaux de variables est syntaxiquement incorrecte?

- (a) `Person[] p = new Person[5];`
- (b) `Person p[5];`
- (c) `Person[] p [];`
- (d) `Person p[][] = new Person[2][];`

2. Observez le morceau de code suivant:

```
public class Test {  
    public static void main(String args[]) {  
        int i = 0, j = 5 ;  
        for( ; (i < 3) && (j++ < 10) ; i++ ) {  
            System.out.print(" " + i + " " + j );  
        }  
        System.out.print(" " + i + " " + j );  
    }  
}
```

quel en sera le résultat?

- (a) `0 6 1 7 2 8 3 8`
- (b) `0 6 1 7 2 8 3 9`
- (c) `0 5 1 5 2 5 3 5`
- (d) erreur de compilation

3. Lesquelles de ces déclarations est correcte? (2 réponses)

- [a] `boolean b = TRUE;`
- [b] `byte b = 255;`
- [c] `String s = "null";`
- [d] `int i = new Integer("56");`

4. Imaginez qu'une classe a comme visibilité public. Dans cette classe, nous définissons une méthode qui est protégée (protected). Laquelle de ces affirmations est dans ce cas correcte?
- (a) Cette méthode est uniquement et seulement disponible dans la classe elle-même et à l'intérieur de toutes les sous-classes.
  - (b) On ne peut pas déclarer dans une classe des méthodes avec une visibilité moindre que la visibilité de la classe elle-même.
  - (c) On n'a pas accès à des méthodes publiques à partir de méthodes protégées.
  - (d) Cette méthode est utilisable à l'intérieur de la classe elle-même et à partir de toutes les classes qui sont définies dans le même package que cette classe.

5. Observez le morceau de code suivant:

```
public class Company{
    public abstract double calculateSalaries();
}
```

Quelle affirmation est dès lors correcte?

- (a) Les mots clés `abstract` et `public` ne peuvent pas être utilisés ensemble.
- (b) La méthode `calculateSalaries` dans la classe `Company` doit avoir un body.
- (c) Une instruction `return` doit se trouver dans la méthode `calculateSalaries`.
- (d) La classe `Company` doit être définie comme abstraite.

6. Observez le morceau de code suivant:

```
public interface Guard{
    void doYourJob();
}

abstract public class Dog implements Guard{}
```

Quelle affirmation est dès lors correcte?

- (a) Ce code ne pourra pas être compilé parce que la méthode `doYourJob()` doit être définie en tant qu'abstraite dans l'interface `Guard`.
- (b) Ce code ne pourra pas être compilé parce que la classe `Dog` doit implémenter la méthode `doYourJob()` à partir de l'interface `Guard`.
- (c) Ce code ne pourra pas être compilé parce que dans la déclaration de la classe `Dog` il faut utiliser le terme `extends` à la place de `implements`.
- (d) Ce code pourra être compilé sans aucun problème.

7. Observez les classes suivantes:

```
public class Person{
    public void talk(){ System.out.print("I am a Person "); }
}
public class Student extends Person {
    public void talk(){ System.out.print("I am a Student "); }
}
```

```
}
```

Quel sera le résultat du morceau de code qui suit:

```
public class Test{
    public static void main(String args[]){
        Person p = new Student();
        p.talk();
    }
}
```

- (a) I am a Person
- (b) I am a Student
- (c) I am a Person I am a Student
- (d) I am a Student I am a Person

8. Observez le morceau de code suivant:

```
public class Person{
    private String firstName;
    public Person(String fn){ firstName = fn; }
}

public class Student extends Person{
    private String studentNumber;
    public Student(String number) { studentNumber = number; }
}
```

Quelles affirmations sont dès lors correctes? (2 réponses)

- [a] Ce code pourra être compilé si l'on définit dans la classe `Person` un constructeur du type no-argument.
- [b] Ce code pourra être compilé si l'on définit dans la classe `Student` un constructeur du type no-argument.
- [c] Ce code pourra être compilé dans le cas où l'on code l'instruction qui suit à la première ligne dans le constructeur de `Student`:  
`super();`
- [d] Ce code pourra être compilé si l'on fait appel au constructeur de `Person` à partir du constructeur de `Student`.

9. Indiquez les caractéristique correctes d'un 'enumeration type' (2 réponses)

- [a] `enum` peut définir des champs et méthodes statiques
- [b] `enum` peut contenir un constructeur publique
- [c] `enum` peut implémenter des interfaces
- [d] `enum` est une référence vers une série variable de constantes

10. Observez le morceau de code suivant:

```
class Person { public int number; }
```

```

public class Test{
    public void doIt(int i , Person p){
        i = 5;
        p.number = 8;
    }

    public static void main(String args[]){
        int x = 0;
        Person p = new Person();
        new Test().doIt(x, p);
        System.out.println(x + " " + p.number);
    }
}

```

Quel en sera le résultat en sortie?

- (a) 0 8
- (b) 5 0
- (c) 0 0
- (d) 5 8

11. Observez le morceau de code suivant:

```

class SalaryCalculationException extends Exception{}

class Person{
    public void calculateSalary() throws SalaryCalculationException {
        //...
        throw new SalaryCalculationException();
        //...
    }
}

class Company{
    public void paySalaries(){
        new Person().calculateSalary();
    }
}

```

Quelles affirmations sont dès lors correctes? (2 réponses):

- [a] Ce code pourra être compilé sans problème.
- [b] Ce code pourra être compilé si l'on fait renvoyer `paySalaries()` une valeur booléenne à la place de `void`.
- [c] Ce code pourra être compilé si l'on rajoute dans `paySalaries()` un `try-catch-block`.
- [d] Ce code pourra être compilé si l'on rajoute dans la signature de la méthode `paySalaries()` "`throws SalaryCalculationException`".

12. A propos des méthodes statiques, quelles affirmations sont correctes? (2 réponses)

- [a] les méthodes statiques sont difficiles à entretenir par leur implémentation qui ne peut plus être modifiée.
- [b] les méthodes statiques peuvent être utilisées à travers une référence objet à un objet de la classe dans laquelle les méthodes sont définies.
- [c] les méthodes statiques sont toujours du type public parce qu'elles sont définies au niveau de la classe.
- [d] les méthodes statiques n'ont pas d'accès direct aux méthodes qui ne sont pas statiques et qui sont définies dans la même classe.

13. Observez le morceau de code suivant:

```
class Person{ public void talk(){} }

public class Test{
    public static void main(String args[]){
        Person p = null;
        try{
            p.talk();
        } catch(NullPointerException e){
            System.out.print("There is a NullPointerException. ");
        } catch(Exception e){
            System.out.print("There is an Exception. ");
        }
        System.out.print("Everything went fine. ");
    }
}
```

Laquelle de ces affirmations est juste.

- (a) L'exécution de ce programme donnera ceci:  
There is a NullPointerException. Everything went fine.
- (b) L'exécution de ce programme donnera ceci:  
There is a NullPointerException.
- (c) L'exécution de ce programme donnera ceci:  
There is a NullPointerException. There is an Exception.
- (d) Ce code ne pourra pas être compilé parce que Java ne connaît pas les pointeurs.

14. A propos de l'usage des **Generics**, quelles affirmations sont correctes? (2 réponses)

- [a] Generics sont des sous-classes typées des classes du framework des Collections
- [b] Generics sont utilisés pour paramétrer les collections afin de vérifier les types des objets de la collection au niveau de la compilation.
- [c] Generics peuvent être utilisés pour vérifier les types des objets d'une collection au niveau d'exécution.
- [d] Generics rendent possible d'itérer facilement sur la collection totale par une boucle 'enhanced for'.

15. Quelle classe de collection associe des valeurs avec des clés et permet de trier ces clés dans un ordre naturel?
- (a) `java.util.HashSet`
  - (b) `java.util.LinkedList`
  - (c) `java.util.TreeMap`
  - (d) `java.util.SortedSet`
16. Quelle affirmation en rapport avec les composants GUI est fausse?
- (a) Swing existe depuis la version 1.2 de JDK.
  - (b) AWT veut dire Abstract Window Toolkit
  - (c) On peut placer des composants AWT dans des containers Swing.
  - (d) Les classes AWT sont obsolètes (deprecated).
17. Quelles affirmations en rapport avec les événements sont justes (2 réponses)
- [a] Les objets `Event` sont placés sur une file d'attente, de laquelle ils sont retirés par des subscribers (des objets de classes qui implémentent l'interface `Subscriber`).
  - [b] Un écouteur (listener) d'un événement doit toujours implémenter la méthode `public void listen(EventObject obj)`.
  - [c] Chaque objet `Event` doit être un objet d'une sous-classe de `EventObject`.
  - [d] Chaque écouteur (listener) de `Event` est en mesure de savoir quelle est la source de l'Event en appelant la méthode `getSource()` de l'objet `Event`.
18. Comment peut-on sérialiser un objet?
- (a) On doit implémenter l'interface `Serializable` dans la classe de l'objet.
  - (b) On doit faire appel à la méthode `serializeObject()` (qui est héritée de la classe `Object`) pour l'objet.
  - (c) On doit appeler la méthode statique `serialize(Object obj)` de la classe `Serializer` avec comme argument l'objet que l'on veut sérialiser.
  - (d) On ne doit absolument rien faire car tous les objets peuvent être par défaut sérialisés.
19. Quelles affirmations en rapport avec les IO sont justes (2 réponses)?
- [a] `OutputStream` est la super-classe abstraite de toutes les classes qui représentent un flot de bytes en sortie.
  - [b] Les sous-classes de la classe `Reader` sont utilisées pour lire des flots de caractères
  - [c] Pour écrire des caractères vers un flot de sortie, on doit utiliser la classe `CharacterOutputStream`.
  - [d] Pour écrire un objet sur un fichier on utilise la classe `ObjectFileWriter`



20. Observez le morceau de code suivant:

```
public class MyThread extends Thread{
    public String text;
    public void run(){
        System.out.print(text);
    }
}

public class Test{
    public static void main(String args[]){
        MyThread t1 = new MyThread(); t1.text = "one ";
        MyThread t2 = new MyThread(); t2.text = "two ";
        t1.start();
        t2.start();
        System.out.print("three ");
    }
}
```

Laquelle de ces affirmations est dès lors correcte:

- (a) Lorsque l'on exécute ce programme, on reçoit toujours le résultat suivant:  
one two three
- (b) Lorsque l'on exécute ce programme, on reçoit toujours le résultat suivant:  
three one two
- (c) Le résultat de ce programme est incertain.
- (d) La compilation de ce programme va échouer.

21. Laquelle de ces commandes relatives aux expressions Lambda est syntaxiquement correcte?

- (a) (a, b) -> a.getName()
- (b) a, b -> a.startsWith("hello")
- (c) (int y, z) -> {int x=1; return y+5; }

# EVALUATION.

Ici sont les réponses correctes:

1. b
2. a
3. c d
4. d
5. d
6. d
7. b
8. a d
9. a c
10. a
11. c d
12. b d
13. a
14. b d
15. c
16. d
17. c d
18. a
19. a b
20. c
21. a

Comptez 1 point par bonne réponse. Pour les questions avec plusieurs bonnes réponses, comptez 1 point seulement si vous avez coché toutes les bonnes alternatives.

Si votre score atteint 80% ou plus, il ne vous est pas nécessaire de suivre ce cours [Programmation en Java](#).

Si votre score est entre 50% et 80%, ce cours [Programmation en Java](#) vous permettra de compléter vos connaissances.

Si votre score est inférieur à 50%, il vous est vivement conseillé de suivre ce cours [Programmation en Java](#).