



Apprenez à programmer en Java

40 heures



Difficile

Mis à jour le 05/08/2019



Modélisez une solution objet

Bravo ! Vous avez réussi cet exercice !

Compétences évaluées



Modéliser une solution objet

Question 1

Quel mot-clé doit-on utiliser pour instancier un objet ?

☒ new☐ this☐ public☐ class☐ private☐ void

Pour instancier une classe et ainsi créer un objet, il faut utiliser le mot-clé **new**.

Question 2

Avec quel mot-clé peut-on restreindre l'étendue d'une variable, d'une méthode ou d'une classe ?

- ☐ new
- ☐ public
- ✓ ☒ private
- ☐ this
- ☐ void
- ☐ class

Le mot-clé **private**, associé à divers éléments du langage Java comme des variables d'instances, des méthodes ou des classes permet de restreindre leurs étendues, limitant ainsi leurs accessibilités.

Question 3

Quel mot-clé permet de créer une variable de classe ?

- ☐ Aucun
- ☐ this
- ☐ private
- ☐ public
- ✗ ☒ class
- ✓ ☐ static

Une variable de classe est une variable n'appartenant pas à une instance précise, mais à toute la classe et donc à toutes ses instances. Ce type de variables s'appelle des variables statiques et sont déclarées avec le mot-clé **static**.

Question 4

Avec quel mot-clé peut-on accéder aux méthodes d'une classe mère lorsque vous créez une classe héritée ?

- ☐ this
- ☐ public
- ☐ private
- ☐ static
- ✓ ☒ super
- ☐ class

Lorsque vous créez de l'héritage de classe, il est parfois nécessaire de rappeler du code présent dans la classe mère afin de rajouter un comportement par exemple. Ceci se fait grâce au mot-clé **super**.

Question 5

Quelle caractéristique principale a une méthode polymorphe ?

- ☐ Elle n'a rien de spécial.
- ✓ ☒ Elle a toujours les mêmes caractéristiques, peu importe l'objet qui l'invoque.
- ☐ Elle n'a pas le même nombre de paramètres en fonction de l'objet qui l'invoque.
- ☐ Elle n'a pas le même nom en fonction de l'objet qui l'invoque.

Une méthode polymorphe est une méthode qui permet d'avoir un comportement différent en fonction du type d'objet tout en s'assurant que l'invocation de la méthode est identique à tout le monde. Vous pouvez donc, sans vous soucier du type d'objet utilisé, invoquer une méthode polymorphe.

Question 6

Quels sont les méthodes de la classe Object qui sont (très) souvent redéfinies pour bénéficier des avantages du polymorphisme ?

- ☐ equals(), wait() et notify()
- ☐ equals, toString() et notify()
- ✓ ☒ equals(), toString() et hashCode()
- ☐ toString(), hashCode() et notify()
- ☐ equals(), hashCode() et notify()
- ☐ equals(), hashCode() et toString()

Les méthodes **equals()**, **toString()** et **hashCode()** sont les méthodes que vous aurez à redéfinir le plus souvent car ce sont ces méthodes qui permettent à vos objets d'avoir une identité propre. Si vous ne le faites pas, je vous rappelle que ce sera la méthode de la classe mère qui sera invoqué, donc la classe Object !

Question 7

Quelle nouveauté de Java 7 vous permet de simplifier votre redéfinition des méthodes equals() et hashCode() ?

- ✓ ☒ La classe java.util.Objects
- ☐ La classe java.util.Object
- ☐ La classe java.util.Equals et java.util.Hashcode
- ☐ La classe java.util.Comparable

- ☐ La classe `java.util.objects`

La classe **`java.util.Objects`** contient des méthodes qui vous permettent de redéfinir facilement le comportement des méthodes `equals()` et `hashCode()`. Cette nouveauté permet de gagner en lisibilité en limitant le nombre de lignes de code.

Question 8

De combien de classes une classe fille peut-elle hériter ?

- ✓ ☒ 1
- ☐ 2
- ☐ 3
- ☐ 4
- ☐ 5
- ☐ 6

Une classe ne peut hériter que d'une et une seule classe ! Faire hériter une classe de plus d'une classe entraînera une erreur de compilation.

Question 9

Qu'est-ce qu'un package ?

- ✓ ☒ C'est un endroit où ranger nos classes.
- ☐ C'est une classe du langage Java.
- ☐ C'est un conteneur permettant d'exécuter du code Java.
- ☐ C'est une classe du langage Java permettant de créer plusieurs objets à l'intérieur.

C'est un endroit où ranger nos classes. Un package est en fait un répertoire où nos fichiers Java sont rangés et où vous pouvez définir les droits d'utilisations en fonction de la portée de nos classes.

Question 10

Cette classe sera-t-elle utilisable dans un autre package ?

```
package com.openclassroom.certification.java; class TestPackage{ }
```

- ☐ Oui
- ☐ Non, le nom du package est trop long.
- ☐ Non, la classe doit avoir un constructeur.

✓ ☒ Non, la portée de la classe n'est pas bonne.

Non, la classe n'a pas de portée public qui lui permettrait de pouvoir être utilisée à l'extérieur du package. Sans être précédée du mot-clé public, la portée d'une classe dans un package est, par défaut, restreint à l'intérieur du package.

Question 11

Qu'est-ce qu'une classe abstraite ?

- ✓ ☒ C'est une classe déclarée avec le mot-clé abstract, ne pouvant pas être instanciée et pouvant contenir des méthodes abstraites, donc sans corps de méthode.
- ☐ C'est une classe ne pouvant pas être héritée.
- ☐ C'est une classe ne pouvant pas avoir de constructeur.
- ☐ C'est une classe du langage Java permettant de créer plusieurs objets en son sein.

Une classe abstraite est une classe ne pouvant pas être instanciée mais rien ne l'empêche d'avoir des méthodes, un constructeur ou tout autre chose qu'une classe normale peut avoir.

Question 12

Dans le langage Java, qu'est-ce qu'une interface ?

- ☐ Une fenêtre permettant à l'utilisateur d'interagir avec le programme.
- ☐ Une classe présente dans un package du langage Java servant de base à tous les objets du langage.
- ☐ Une classe abstraite
- ✓ ☒ Une classe 100% abstraite permettant de créer un nouveau super type et jouir du polymorphisme.

*Une classe abstraite peut contenir des méthodes non abstraites alors qu'une interface ne définit que des entêtes de méthodes, donc sans corps. Ceci permet de créer un nouveau super type : les classes implémentant une ou plusieurs interfaces pourront alors être instanciées et stockées dans un conteneur du type de l'interface. C'est une alternative à la restriction d'héritage multiple imposé par Java. **Important :** Depuis Java 8, les interfaces peuvent contenir des définitions de méthodes par défaut. Les interfaces ne seront donc plus abstraites à 100%. La raison est que ce mécanisme pourra permettre de rajouter facilement des méthodes dans une interface sans avoir à les redéfinir dans toutes les classes les implémentant.*

Question 13

Qu'offre Java avec les interfaces java.util.Collection et java.util.Map ?

- ✓ ☒ Les interfaces dont héritent les collections présentes.
- ☐ Rien, ces interfaces n'existent pas.

- ☐ Rien, ce ne sont pas des interfaces mais des classes.
- ☐ Rien, ces interfaces n'ont pas d'implémentation.

Ces interfaces forment la première pierre à l'édifice des classes permettant de gérer des collections d'objets en Java. Ces objets implémentant ces interfaces vous offrent la possibilité de gérer, plus finement qu'avec des tableaux classiques, des paquets d'objets : certains sont triés automatiquement, certains n'autorisent pas les doublons, etc.

Question 14

Qu'est-ce que la généricité ?

- ☐ Un concept permettant d'avoir une classe, une méthode ou une collection figée pour chaque utilisation.
- ☐ Un concept permettant d'avoir un code appelé dans chaque classe de façon identique.
- ☒ Un concept permettant de ne pas spécifier de type précis pour une classe, une collection ou une méthode afin d'avoir du code réutilisable.
- ☐ Un concept permettant à une classe de ne pas avoir de classe fille.

La généricité permet d'avoir une entité Java (classe, méthode ou collection) pouvant être utilisée avec plusieurs type d'objets, sans avoir à recoder entièrement la classe, la méthode ou la collection.

[◀ CRÉEZ UNE JVM MODULAIRE AVEC JAVA 9](#)[ACTIVITÉ : RÉALISEZ UN GARAGE ▶](#)

Le professeur

Cyrille Herby

Spécialiste en développement Java et curieux insatiable d'informatique et de programmation web.
Actuellement auditeur en sécurité.

Découvrez aussi ce cours en...



Livre



PDF

L'entreprise

Alternance

Forum

Blog

Nous rejoindre

Entreprises

Employeurs

En plus

Devenez mentor

Aide et FAQ

Conditions Générales d'Utilisation

Politique de Protection des Données Personnelles

Nous contacter

 Français 



Télécharger dans
l'App Store
