



Accueil > Cours > Programmez avec le langage C++ > Quiz : Expliquez les principes de l'orienté objet en C++

Programmez avec le langage C++

50 heures  Difficile

Mis à jour le 25/03/2019



Expliquez les principes de l'orienté objet en C++

Bravo ! Vous avez réussi cet exercice !

Compétences évaluées



Expliquer les principes de l'orienté objet en C++

Question 1

Que dit la règle de l'encapsulation ?

- ✓ ☒ Tous les attributs d'une classe doivent être privés.
- ☐ Toutes les méthodes d'une classes doivent être privées.
- ☐ Tous les attributs d'une classe doivent être publics.
- ☐ Tous les éléments d'une classe doivent être privés.

Les attributs d'une classe doivent être privés alors que les méthodes peuvent être publiques pour permettre aux utilisateurs de la classe de manipuler les objets créés à partir de cette classe.

Question 2

Si je sépare la déclaration de la classe de l'implémentation des méthodes, quel préfixe dois-je mettre devant le nom des méthodes ?

- ☐ NomDeLaClasse//
- ✓ ☒ NomDeLaClasse::
- ☐ NomDeLaClasse;;
- ☐ NomDeLaClasse.

On utilise `::` pour indiquer l'appartenance d'une méthode à une classe. Pour rappel, `///` est utilisé pour les commentaires et `."` est utilisé pour appeler une méthode ou accéder à un attribut d'un objet.

Question 3

Quel est le rôle du constructeur ?

- ☐ Nettoyer la mémoire quand l'objet est détruit.
- ☐ Déclarer des variables.
- ✓ ☒ Initialiser les attributs de la classe.
- ☐ Indiquer le nom de la classe.

Le principal rôle du constructeur est d'initialiser les attributs. Ceux-ci ont été déclarés dans la définition de la classe (généralement dans le `.h`) mais n'ont aucune valeur par défaut. C'est au constructeur d'en donner une. Le constructeur a toujours le même nom que la classe.

Question 4

Qu'est-ce qu'un constructeur par défaut ?

- ☐ Un constructeur qui renvoie void.
- ✓ ☒ Un constructeur qui n'a aucun paramètre.
- ☐ Un constructeur qui ne contient aucune instruction.
- ☐ Un constructeur qui n'est jamais appelé.

Un constructeur par défaut n'a aucun paramètre. C'est celui qui est appelé si, lors de la création de l'objet, on n'envoie aucun paramètre. Il peut contenir autant d'instructions que l'on souhaite. Notez aussi qu'un constructeur ne renvoie jamais rien, pas même `"void"`.

Question 5

Qu'est-ce qu'un accesseur ?

- ☐ Une méthode qui affiche tout le contenu de la classe.
- ✓ ☒ Une méthode qui permet d'accéder à tous les éléments privés de la classe.
- ☐ Une méthode qui permet de lire / modifier un attribut indirectement.

On utilise les accesseurs pour modifier un attribut précis d'un objet puisque les attributs sont toujours déclarés "privés" dans une classe et par conséquent inaccessibles directement pour les utilisateurs.

Question 6

Quel est le nom de la fonction correspondant à l'opérateur >= ?

- ✓ ☒ operator>=
- ☐ >=
- ☐ operateur_superieurouegal
- ☐ operator_greaterorequal
- ☐ >=operator

Les opérateurs sont toujours nommés en utilisant le mot "operator" suivi du (ou des) symboles qui correspondent à cet opérateur.

Question 7

Quel est le nom du pointeur généré par le compilateur dans toutes les classes et qui pointe vers l'objet actuel ?

- ✓ ☒ this
- ☐ that
- ☐ me
- ☐ self
- ☐ here

C'est this, qui signifie "cela" en français.

Question 8

J'ai une classe X qui contient un pointeur vers un objet Y. Cet objet Y a été alloué dynamiquement dans le constructeur de X. Dois-je supprimer d'abord X puis Y, ou l'inverse, ou ça n'a pas d'importance ?

- ✓ ☒ D'abord Y puis X
- ☐ L'ordre de suppression n'a pas d'importance
- ☐ D'abord X puis Y

Si on supprime d'abord X, qui contient un pointeur vers Y, on perd le pointeur vers Y... donc on ne connaît plus l'adresse de Y, ce qui fait qu'on ne peut plus le supprimer ! Voilà pourquoi il faut procéder dans l'ordre : d'abord on supprime l'objet pointé Y, puis ensuite on peut supprimer X.

Question 9

La portée "protected" empêche l'accès aux méthodes et attributs qui suivent depuis l'extérieur de la classe, sauf...

- ✓ ☒ ... depuis les classes filles.
- ☐ ... depuis la classe mère.
- ☐ ... depuis le main.
- ☐ ... depuis les constructeur d'autres classes.
- ☐ ... depuis les destructeurs d'autres classes.

Avec une portée protected, seule les classes filles auront le droit d'accès aux éléments. On ne peut toujours pas accéder à ces éléments depuis l'extérieur, notamment depuis le "main" ou dans n'importe quelle autre méthode d'autres classes.

Question 10

Ce code est-il légal (compilera-t-il) ?

```
Truc *monTruc(0);  
Bidule *monBidule = new Bidule();  
  
monTruc = monBidule;
```

- ☐ Oui, quel que soient Bidule et Truc.
- ☐ Oui, si Truc hérite de Bidule.

- ✓ ☒ Oui, si Bidule hérite de Truc.
- ☐ Non, jamais.

Si Bidule hérite de Truc, on dit que "Bidule EST UN Truc". On peut donc parfaitement affecter un pointeur sur un Bidule à un pointeur sur un Truc.

Question 11

Qu'est-ce que le masquage ?

- ☐ Utiliser l'opérateur de résolution de portée pour utiliser la fonction héritée.
- ✓ ☒ Le fait de cacher une fonction membre héritée en écrivant une fonction de même nom dans la classe fille.
- ☐ Mettre tous les attributs en privé pour cacher l'implémentation aux utilisateurs.
- ☐ Avoir deux fonctions de noms presque semblables pour faire deux actions presque semblables.

On masque une méthode lorsque l'on crée une méthode de même nom dans une classe fille. Cette méthode remplace la version héritée. Il faut alors utiliser l'opérateur de résolution de portée (.👤) pour accéder à la version de la méthode définie dans la classe mère.

Question 12

Quelles méthodes ne peuvent pas être virtuelles ?

- ✓ ☒ Les constructeurs
- ☐ Les méthodes avec plus d'un argument
- ☐ Les opérateurs
- ☐ Le destructeur
- ☐ Les méthodes renvoyant void

Les seules méthodes ne pouvant pas être virtuelles sont les constructeurs. Les destructeurs, eux, doivent toujours être virtuels lorsque l'on utilise le polymorphisme.

Question 13

Combien d'instances de classes abstraites peut-on créer ?

- ☐ Une seule à la fois.

- ☐ Autant qu'il y a de classes dérivées.
- ☐ Autant que l'on veut.
- ✓ ☒ On ne peut pas créer d'instance à partir d'une classe abstraite.
- ☐ Une seule qui existe dès le début du programme et qui est détruite à la fin.

Il n'est pas possible d'instancier une classe abstraite. Le but est d'empêcher l'appel aux méthodes virtuelles pures qui n'ont pas de corps.

Question 14

Quels sont les ingrédients nécessaires à la résolution dynamique des liens ?

- ✓ ☒ Une fonction virtuelle ainsi qu'un pointeur ou une référence.
- ☐ Une fonction virtuelle et un pointeur.
- ☐ Une classe abstraite et un pointeur.
- ☐ Une fonction virtuelle pure et une référence.
- ☐ Une fonction virtuelle et une référence.

Pour que la résolution dynamique des liens se fasse, il faut appeler une méthode virtuelle en utilisant un pointeur ou une référence.

Question 15

À quels constituants d'une classe, une fonction amie a-t-elle accès ?

- ☐ À toutes les méthodes
- ✓ ☒ À tous les attributs et méthodes
- ☐ À tous les attributs
- ☐ À toutes les méthodes publiques
- ☐ À toutes les méthodes protected

Une fonction déclarée friend d'une classe a un accès complet à cette dernière. Elle peut donc lire et modifier tous les attributs et utiliser toutes les méthodes peu importe leurs portées.

Question 16

Quelle est la meilleure manière de créer un opérateur + pour une classe "Euro" représentant une quantité d'argent ?

- ☐ Créer un opérateur + comme méthode de la classe.
- ☐ Créer un opérateur + comme fonction externe utilisant les accesseur de Euro.
- ✓ ☒ Créer un opérateur + comme fonction externe utilisant l'opérateur += de Euro.
- ☐ On ne peut pas créer un tel opérateur.

La meilleure manière de créer un opérateur + est de définir un opérateur += pour notre classe (en général une méthode) et ensuite de créer un opérateur + comme fonction externe utilisant l'opérateur += lors des calculs.

◀ **UTILISEZ LES ÉLÉMENTS STATIQUES ET
L'AMITIÉ**

INITIEZ-VOUS À QT ▶

Les professeurs

Mathieu Nebra

Entrepreneur à plein temps, auteur à plein temps et co-fondateur d'OpenClassrooms :o)

Matthieu Schaller

Chercheur en astrophysique et cosmologie. Spécialiste en simulations numériques de galaxies sur superordinateurs.

Découvrez aussi ce cours en...



Livre



PDF

OpenClassrooms

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 ▼

