

Auto-évaluation Concepts et Techniques

00

Document: f0611test.fm

3 septembre 2019

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

The logo for ABIS Training & Consulting. The word "abis" is written in a bold, blue, lowercase sans-serif font. A solid black circle is positioned above the letter 'i'. Below the text is a thick, horizontal grey bar.

TRAINING & CONSULTING

INTRODUCTION AUTO-ÉVALUATION CONCEPTS ET TECHNIQUES OO

Ce test comporte des questions à choix multiples. Pour la plupart des questions, plusieurs réponses correctes sont possibles. Cochez les solutions qui vous semblent bonnes, et comparez vos réponses avec les solutions.

QUESTIONS AUTO-ÉVALUATION CONCEPTS ET TECHNIQUES OO

1. Quelle liste caractérise le mieux la programmation orienté objet?
 - (a) architecture en couches, concurrency, couplage fort
 - (b) encapsulation, héritage, polymorphisme
 - (c) modélisation des données, sous-types, services
 - (d) classes, objets, haute performance
2. Lesquelles parmi ces propositions à propos des **objets** sont vraies? [2 réponses]
 - [a] Un objet est initialisé par redéfinition ('overriding').
 - [b] Un objet est une instance d'une classe.
 - [c] Dans une application, un objet est accédé par son 'object reference'.
 - [d] Un objet définit le comportement de sa classe.
3. Que faut-il faire pour étendre une classe ('**extend**')?
 - (a) attribuer plus de mémoire à la classe.
 - (b) créer une classe plus spécifique.
 - (c) créer des instances supplémentaires de cette classe.
 - (d) créer une super-classe.
4. Que représente le diagramme UML suivant?



- (a) une composition
- (b) une association
- (c) un héritage
- (d) une interaction composée

5. Quelle proposition à propos de l'**héritage** ('inheritance') représente le mieux la philosophie OO?
- (a) L'héritage indique comment les classes varient dans le temps.
 - (b) L'héritage indique comment des message sont transmis entre les classes.
 - (c) L'héritage est une technique de programmation pour réduire la quantité de code.
 - (d) L'héritage permet de modéliser clairement les similarités et les différences.
6. Quelle proposition est correcte à propos des **classes** et des **objets**?
- (a) Chaque classe est instantiée dans 1 ou plusieurs objets.
 - (b) Des objets changent de classe dans le temps à cause du polymorphisme.
 - (c) Une classe fournit le code à exécuter par ses objets.
 - (d) Les classes sont illustrées par les analystes dans des diagrammes conceptuels, puis les objets sont codés par les programmeurs.
7. Quelle proposition est correcte à propos des **classes abstraites** et **concrètes**?
- (a) Une classe abstraite est héritée.
 - (b) Une classe concrète doit avoir un 'use case' concret.
 - (c) Une classe abstraite n'est jamais instantiée.
 - (d) Une classe abstraite n'est pas montrée dans un diagramme de classes.
8. Que signifie le **polymorphisme** dans la programmation OO?
- (a) qu'une classe utilisante ('client class') ne doit pas avoir connaissance de la sous-classe spécifique qui implémente une méthode utilisée.
 - (b) que l'objet, dans son cycle de vie, peut prendre plusieurs 'formes' en changeant de classe.
 - (c) que, lors du stockage, un seul objet peut être représenté (par 'mapping') dans plusieurs tables d'une base de données.
 - (d) qu'une sous-classe peut redéfinir les méthodes de la super-classe à condition qu'elle ne soit pas abstraite.
9. La **hiérarchie** de super-classes et de sous-classes permet de faire quoi? [2 réponses]
- [a] Documenter des associations entre des classes.
 - [b] Réaliser le polymorphisme.
 - [c] Utiliser d'une façon sûre une collection d'objets de différentes classes de la hiérarchie.
 - [d] Changer dynamiquement le comportement d'un objet, en faisant référence à n'importe quelle autre classe dans la même hiérarchie.

10. Quelle proposition est vraie à propos de **UML**?
- (a) UML est une approche standardisée pour la modélisation 'use case'.
 - (b) UML fournit des conventions de notation, utilisées lors du développement d'applications.
 - (c) UML est une méthodologie pour la conception et l'entretien d'un système informatique.
 - (d) UML est une alternative pour RUP ou Agile.
11. Quelle technique est proposée par la plupart des méthodes OO pour décrire la **fonctionnalité** et la **portée** ('scope') d'un système?
- (a) use cases
 - (b) un diagramme d'état
 - (c) des flow charts
 - (d) des diagrammes ERD (Entity Relationship Diagram)
12. Que signifie l'**encapsulation** dans l'orienté objet?
- (a) Le code est établi en couches ('layers'), comme par exemple Interface utilisateur, logique métier et persistance.
 - (b) Des données et des procédures logiquement liées sont groupées dans une classe, et le comportement et la structure internes sont cachés pour le monde extérieur.
 - (c) Les objets ne peuvent accéder qu'à leurs propres attributs.
 - (d) L'encapsulation fixe l'implémentation, de sorte qu'après on ne puisse adapter que l'interface publique.
13. Que signifie '**High Cohesion**'?
- (a) que toutes les classes collaborent intensément, ce qui est une bonne idée.
 - (b) que toutes les classes collaborent intensément, ce qui doit être évité.
 - (c) que chaque classe se concentre sur un boulot spécifique, ce qui est une bonne idée.
 - (d) que chaque classe se concentre sur un boulot spécifique, ce qui doit être évité.
14. Quelles propositions sont correctes à propos des '**Design Patterns**'? [2 réponses]
- [a] Les Design Patterns déterminent le design et la mise en page d'une application web.
 - [b] Les Design Patterns sont conçus pour la programmation en Java
 - [c] Les Design Patterns sont devenus populaires par le 'Gang of Four' (GoF).
 - [d] L'utilisation de Design Patterns profite à la maintenabilité d'une application

15. Qu'est-ce qui est vrai à propos du **testing**? [2 réponses]
- [a] Une bonne application OO ne peut être testée que quand tous les composants sont prêts.
 - [b] Le testing fait partie intégrante des méthodes OO comme le RUP et l'Extreme Programming.
 - [c] Le 'black-box' testing d'un composant demande la connaissance de son implémentation concrète.
 - [d] Les use cases s'utilisent pour la rédaction de scénarios de test.
16. Qu'est-ce qui est vrai à propos de **RUP**?
- (a) Selon le RUP, l'implémentation ne peut commencer que quand le design est terminé.
 - (b) Le RUP est une approche itérative et incrémentielle.
 - (c) Selon le RUP, le design ne peut commencer que quand l'implémentation est terminée.
 - (d) Le RUP déconseille l'emploi de dates limites fixes ('time boxing'), parce que cela compromet le contrôle de la qualité et le testing.
17. JDBC, Hibernate, ADO .NET et ADO. NET Entities sont des standards concernant ...
- (a) les design objects
 - (b) les remote network components
 - (c) la persistance
 - (d) l'administration de la base de données
18. Dans les langages OO, la décision sur le code à exécuter n'est prise que durant l'exécution ('at runtime'). Comment s'appelle cette façon de faire?
- (a) static binding
 - (b) dynamic binding
 - (c) dynamic typing
 - (d) strong typing
19. Que fait le **O/R mapping**?
- (a) Il établit les correspondances entre la représentation objet et le stockage dans une base de données relationnelle.
 - (b) Il décrit les relations mutuelles entre les objets d'une application.
 - (c) Il organise le software OO dans une structure de répertoires hiérarchique.
 - (d) Il fait un croquis (à haut niveau) des objets de l'application et de leur fonctionnalités.

20. Quelles propositions sont correctes à propos des **frameworks**? [2 réponses]

- [a] Un framework est basé sur le distributed processing.
- [b] Un framework est un environnement de développement intégré.
- [c] Un framework détermine le cadre et fournit des composants pour la conception d'une application.
- [d] Un framework contribue à la consistance entre les différents projets.

EVALUATION.

Ici sont les réponses correctes:

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

Comptez 1 point par bonne réponse. Pour les questions avec plusieurs bonnes alternatives, 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.

Si votre score est entre 50% et 80%, ce cours vous permettra de compléter vos connaissances.

Si votre score est inférieur à 50%, il vous est vivement conseillé de suivre ce cours.