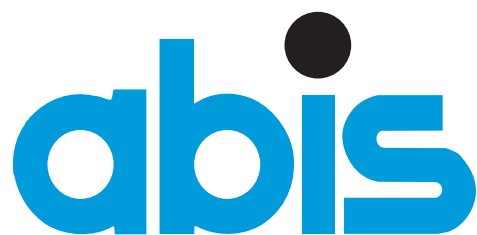


# Zelftest OO concepten en technieken

Document: n0611test.fm

3 september 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

# INTRODUCTIE

Om de voorkennis, nodig voor de cursussen Programmeren in Java en OO analyse en design met UML, te kunnen inschatten, raden wij u aan volgende test uit te voeren. Deze test is gebaseerd op onderwerpen behandeld in de cursus OO concepten en technieken.

Deze test bestaat uit 20 vragen. Voor elke vraag moet u één antwoord aankruisen (tenzij anders vermeld). Een vraag is pas correct beantwoord wanneer alle correcte antwoordmogelijkheden zijn aangekruist.

Reken ca. 10 minuten voor het invullen.

De juiste antwoorden en richtlijnen voor de evaluatie vindt u achteraan dit document.

# VRAGEN

1. Welke lijst typeert het best **Object Oriented** programmeren?
  - (a) gelaagde architectuur, concurrency, hoge koppeling
  - (b) encapsulatie, overerving, polymorfisme
  - (c) data modellering, subtypes, services
  - (d) klassen, objecten, hoge performantie
2. Welke zijn juiste beweringen over **objecten**? [2 antwoorden]
  - [a] Een object wordt geïnitieerd door 'overriding'.
  - [b] Een object is een instance van een klasse.
  - [c] In een toepassing kan een object benaderd worden via een object-referentie.
  - [d] Een object bepaalt het gedrag van zijn klasse.
3. Wat moet je doen om een klasse **uit te breiden** ('extend')?
  - (a) meer geheugen toekennen aan de klasse
  - (b) een meer specifieke klasse maken
  - (c) extra instances creëren van de klasse
  - (d) een superklasse creëren
4. Wat stelt het volgende UML diagram voor?



- (a) een compositie
- (b) een associatie
- (c) een overerving
- (d) een samengestelde interactie

5. Welke uitspraak over **overerving** ('inheritance') geeft het best de OO filosofie weer?
- (a) Overerving toont hoe klassen veranderen in de tijd.
  - (b) Overerving toont hoe boodschappen worden doorgestuurd tussen klassen.
  - (c) Overerving is een programmeertechniek om de hoeveelheid code te reduceren.
  - (d) Overerving laat toe om gelijkenissen en verschilpunten duidelijk te modelleren.
6. Wat is juist i.v.m. **klassen** en **objecten**?
- (a) Elke klasse wordt geïnstantieerd in 1 of meer objecten.
  - (b) Objecten veranderen van klasse in de tijd omwille van polymorfisme.
  - (c) Een klasse levert de programmacode die door haar objecten wordt uitgevoerd.
  - (d) Klassen worden door analisten geïllustreerd in conceptuele diagramma's en objecten worden dan geschreven door programmeurs.
7. Wat is juist i.v.m. **abstracte** en **concrete** klassen?
- (a) Een abstracte klasse wordt overgeërfd.
  - (b) Een concrete klasse heeft een concrete 'use case'.
  - (c) Een abstracte klasse wordt nooit geïnstantieerd.
  - (d) Abstracte klassen worden niet getoond in een class diagram.
8. Bij object georiënteerd programmeren betekent **polymorfisme**
- (a) dat een gebruikende klasse ('client class') niet moet afweten van de specifieke subklasse die een gebruikte methode implementeert.
  - (b) dat een object tijdens zijn levenscyclus verschillende 'vormen' kan aannemen, door van klasse te veranderen.
  - (c) dat één object bij opslag kan worden afgebeeld op meerdere database tabellen.
  - (d) dat een subklasse de methodes van een parent class alleen kan overschrijven als ze niet abstract zijn.
9. De **hiërarchie** van superklassen en subklassen maakt het mogelijk om *[2 antwoorden]*
- [a] associaties te leggen tussen de verschillende klassen.
  - [b] polymorfisme te implementeren.
  - [c] op een veilige manier te werken met verzamelingen van objecten van verschillende klassen in de hiërarchie.
  - [d] object gedrag dynamisch te veranderen, door te verwijzen naar gelijk welke andere class binnen de hiërarchie.

10. Welk is een juiste bewering over **UML**?
- (a) UML is een gestandaardiseerde aanpak voor 'use case' modellering.
  - (b) UML levert een set van noteringsafspraken, gebruikt bij het ontwerpen van toepassingen.
  - (c) UML is een methodologie voor design en onderhoud van computer systemen.
  - (d) UML is een alternatief voor de RUP en Agile.
11. Welke techniek stellen de meeste OO methodes voor om de **functionaliteit** en de **reikwijdte** ('scope') van een systeem te beschrijven?
- (a) use cases
  - (b) een statechart diagram
  - (c) flow charts
  - (d) Entity Relationship Diagrammen
12. Wat betekent **encapsulatie** in object oriëntatie?
- (a) De code is opgebouwd uit lagen (zoals bijvoorbeeld gebruikersinterface, business logica en persistentie).
  - (b) Logisch gerelateerde data en procedures worden gebundeld in een klasse, terwijl intern gedrag en gegevensstructuur verborgen worden.
  - (c) Objecten kunnen uitsluitend hun eigen attributen benaderen.
  - (d) Encapsulatie legt de implementatie vast, waardoor nadien alleen nog de publieke interface kan worden aangepast.
13. Wat betekent '**High Cohesion**'?
- (a) Het betekent dat alle klassen zeer nauw samenwerken, wat een goed principe is.
  - (b) Het betekent dat alle klassen zeer nauw samenwerken, wat moet vermeden worden.
  - (c) Het betekent dat elke klasse 1 ding zeer goed doet, wat een goed principe is.
  - (d) Het betekent dat elke klasse 1 ding zeer goed doet, wat moet vermeden worden.
14. Welke zijn juiste beweringen over '**Design Patterns**' ? [2 antwoorden]
- [a] Design Patterns bepalen de grafische vormgeving van de pagina's in een web applicatie.
  - [b] Design Patterns zijn specifiek bedacht voor Java programmatie.
  - [c] Design Patterns werden populair door de 'Gang of Four' (GoF).
  - [d] Het gebruik van Design Patterns verbetert de onderhoudbaarheid van een applicatie.

15. Wat is juist i.v.m. **testen**? [2 antwoorden]
- [a] Een goede OO applicatie kan pas getest worden als alle onderdelen af zijn.
  - [b] Testing is een integraal onderdeel van OO methodes als RUP en Extreme Programming.
  - [c] Black-box testing van een component vereist kennis van de concrete implementatie.
  - [d] Use cases worden gebruikt voor het opstellen van test scenario's.
16. Wat is juist i.v.m. de **RUP**?
- (a) Volgens de RUP kan de implementatie pas beginnen als de design klaar is.
  - (b) De RUP is een iteratieve en incrementele benadering.
  - (c) Volgens de RUP kan de design pas beginnen als de implementatie klaar is.
  - (d) De RUP ontmoedigt het gebruik van vaste deadlines ('time boxing'), omdat daardoor kwaliteit zorg en testen in het gedrang komen.
17. JDBC, Hibernate, ADO .NET en ADO.NET Entities zijn standaarden voor
- (a) design objects
  - (b) remote network components
  - (c) persistence
  - (d) database administration
18. In OO talen wordt de beslissing over de uit te voeren code vaak pas gemaakt tijdens de uitvoering van het programma. Hoe noemt men dit principe?
- (a) static binding
  - (b) dynamic binding
  - (c) dynamic typing
  - (d) strong typing
19. Waarover gaat **O/R mapping**?
- (a) Het verband leggen tussen de object-voorstelling en de opslag in een relationele database.
  - (b) Het in kaart brengen van de onderlinge relaties tussen de objecten.
  - (c) Het organiseren van de OO software in een hiërarchische folderstructuur.
  - (d) Een ruwe schets maken van de applicatie-objecten en hun functionaliteit.

20. Welke zijn juiste uitspraken over **frameworks**? [2 antwoorden]

- [a] Frameworks zijn gebaseerd op distributed processing.
- [b] Een framework is een gestandaardiseerde ontwikkelomgeving.
- [c] Een framework bepaalt het stramien of levert onderdelen voor de opbouw van een applicatie.
- [d] Frameworks zorgen voor consistentie tussen verschillende projecten.

# EVALUATIE.

Hier zijn de correcte antwoorden op alle vragen:

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

Geef uzelf 1 punt voor elke correct beantwoorde vraag (d.w.z. alle correcte antwoordmogelijkheden aangeduid en geen enkele andere).

Heeft u minder dan 50% van de vragen correct beantwoord, dan raden wij u ten zeerste aan eerst de cursus OO concepten en technieken te volgen.

Heeft u tussen de 50% en 75% van de vragen correct beantwoord, dan weet u al heel wat, maar u zult zeker nog bijleren tijdens de cursus OO concepten en technieken.

Heeft u meer dan 75% correct, dan zult u tijdens de cursus OO concepten en technieken niet veel meer bijleren. U kunt dan onmiddellijk starten met de cursussen Java concepten en Programmeren in Java of de cursus OO analyse en design met UML.