

- het is niet toegestaan een ingeschakelde mobiele telefoon mee te nemen in de tentamenzaal
- het is niet toegestaan ongevraagd de tentamenzaal tussentijds te verlaten voor een w.c. bezoek

Universiteit van Amsterdam  
Opleiding Kunstmatige Intelligentie

# Tentamen Imperatief & Objectgeoriënteerd Programmeren

-- Deeltentamen A --

Cursus 2e semester '09 - '10  
Dinsdag 23 maart 2010, 9.00-12.00, in USC

zaal Universitair Sport Centrum  
De Boelelaan 46, 1082 LR Amsterdam

Er zijn 20 vragen over de tentamenstof, verdeeld over 2 delen:

- deel 1 :10 begripsvragen die betrekking hebben op programmeerconcepten
- deel 2: 10 vragen van over gegeven programmacode in Java (zie BIJLAGE)

Alle vragen zijn van gelijk gewicht bij de beoordeling.

De uitslag is binnen 15 *werkdagen* bekend, het tentamen kan daarna ingekeken worden.  
De uitslag zal zo spoedig mogelijk op de website van de cursus weergegeven worden.

## -tentamen: deel 1-

### Tien begripsvragen die betrekking hebben op programmeerconcepten

*Schrijf het genummerde antwoord op het uitgedeelde antwoordformulier.*

- Vraag 1:** Geef eerst in *pseudocode* en daarna omgezet in Java code een voorbeeld met de noodzakelijke elementen van een herhalingslus, waarbij door een programmegebruiker wordt bepaald hoe vaak de herhaling zal gaan plaatsvinden.
- Vraag 2:** Wat is de functie en werking van een *constructor*? Geef voorbeelden van aanroep (gebruik evt. de Java programmacode van de Bijlage voor het aanwijzen van een voorbeeld). Is er altijd een *default constructor* beschikbaar?
- Vraag 3:** Wat wordt bedoeld met het begrip '*shadowing*' in het kader van de '*scope*' van de definitie van variabelen? Geef een voorbeeld, inclusief een bespreking van de voordelen en de nadelen van '*shadowing*'.
- Vraag 4:** Wat wordt in het kader van het werken met logische operatoren bedoeld met het begrip '*short-circuit evaluation*'? Geef een voorbeeld waarbij van een '*side-effect*' gebruik gemaakt wordt.
- Vraag 5:** Wat wordt beoogd met het software engineering mechanisme '*data hiding*' of '*encapsulation*', en op welke wijze kan dit gerealiseerd worden in de taal Java? Geef in een voorbeeld aan hoe het gerealiseerd wordt binnen de definitie van een klasse, maar ook hoe er gewerkt wordt met die klasse.
- Vraag 6:** Wat wordt bedoeld met het '*principle of least privilege*' en waarom is dit principe van belang voor goede software engineering?
- Vraag 7:** Wat is de reden dat klassemethoden (*static*) geen toegang hebben tot de niet-static variabelen en methoden van de klasse maar omgekeerd wel? Geef een toelichting aan de hand van een eenvoudig voorbeeld.
- Vraag 8:** Welke functionele verschillen zijn er tussen een lusconstructie met een gewoon 'for' statement en een lusconstructie met een 'enhanced-for' statement bij het werken met array's? Geef een voorbeeld van deze verschillen aan de hand van het gebruik van het array: `int array[] = { 7, 5, 10, 23, 3, 52 };`
- Vraag 9:** Op welke wijze vindt *expliciete* en *impliciete* conversie van primitieve typen plaats in Java. Geef een omschrijving en een voorbeeld van beide vormen van conversie.
- Vraag 10:** Wat wordt in het kader van de constructie van complexe computerprogramma's verstaan onder de techniek van '*divide and conquer*'. Waarom leent het ontwerp van de taal Java zich goed voor deze programmeertechniek?

## -tentamen: deel 2-

### Tien vragen over gegeven programmacode (zie BIJLAGE).

- Zie voor de volgende vraag de bijlage met programmacodes:
  - fig. 3.13: **Account.java**
  - fig. 3.14: **AccountTest.java**

**Vraag 11:** Welke bedragen komen er in de laatste twee regels balans-uitvoer te staan van het testprogramma als bij eerste invoer het bedrag -123,456 (minus!) wordt ingevoerd en bij de tweede invoer het bedrag: 123,456 (N.B., gebroken getallen als invoer met drie digits achter de komma)

- Zie voor de volgende twee vragen de bijlage met programmacodes:
  - fig. 4.9 : **Gradebook.java**
  - fig. 4.10: **GradebookTest.java**

**Vraag 12:** In de methode **determineClassAverage()** komt een *cast* voor in het if-statement. Wat zijn de gevolgen als we deze cast niet opnemen. Levert dit een programmafout op? Zo ja, waarom en welke soort fout (logisch, compilatie, run-time). Zo nee, waarom niet. Geef een toelichting.

**Vraag 13:** In de methode **determineClassAverage()** komt een herhaling voor van:

```
// prompt for input and read grade from user
System.out.print( "Enter grade or -1 to quit: " );
grade = input.nextInt();
```

Kan deze herhaling op een eenvoudige manier teruggebracht worden tot één keer, met precies dezelfde werking van het programma? Zo nee, waarom niet, zo ja, op welke wijze?

- Zie voor de volgende twee vragen de bijlage met programmacodes:
  - fig. 6.11: **Scope.java**
  - fig. 6.12: **ScopeTest.java**

**Vraag 14:** Welke getalsmatige uitvoer wordt door het testprogramma bij de uitvoering van de methode *begin()* gegenereerd? Geef een toelichting bij de uitkomst.

**Vraag 15:** Stel dat we de methode-header van de definitie van *begin()* met een parameter *x* uitbreiden: `public void begin( int x )`

Wordt dat toegestaan door de compiler, zo nee, waarom niet en zo ja, is dit een vorm van 'shadowing'? Geef een toelichting.

- Zie voor de volgende vraag de bijlage met programmacode:
  - o fig. 8.7: **Date.java**,
  - o fig. 8.8: **Employee.java**
  - o fig. 8.9: **EmployeeTest.java**

**Vraag 16:** Wanneer in de 'main' in plaats van: `Date birth = new Date( 7, 24, 1949 );` de volgende datum komt te staan: `Date birth = new Date( 6, 31, 1949 );` wat verandert er dan in de uitvoer?

- Zie voor de volgende twee vragen de bijlage met programmacode:
  - o fig. 7.9 : **Card.java**,
  - o fig. 7.10: **DeckOfCards.java**
  - o fig. 7.11: **DeckOfCardsTest.java**

**Vraag 17:** Beschrijf in *pseudocode* het mechanisme van het 'schudden' van de stapel kaarten.

**Vraag 18:** In de methode `dealCard()` van de klasse `DeckOfCards` wordt onder voorwaarde een stapel kaarten teruggegeven, waarbij de het nummer van de volgende kaart die aan de beurt is wordt opgehoogd met een *postfix increment*:

`return deck[currentCard++];`. Maakt het voor de correcte werking van het programma iets uit of die ophoging de vorm heeft van een *prefix increment*: `return deck[++currentCard];` of het *postfix increment*? Geef een toelichting.

- Zie voor de volgende vraag de bijlage met programmacode:
  - o fig. 7.13: **PassArray.java**

**Vraag 19:** Stel dat we het eerste statement in de methode `main` vervangen door het volgende:

```
int array[] = { 1, 2, 3 };
```

Kunnen we het programma normaal laten uitvoeren, zo ja welke uitvoer is er? Zo nee, treden er fouten op en welke dan (compilatie, logische, runtime) en hoe ziet de uitvoer er uit?

- Zie voor de volgende vraag de bijlage met programmacode:
  - o fig. 8.12: **Employee.java**
  - o fig. 8.13: **EmployeeTest.java**

**Vraag 20:** Wat is de rol van de *garbage collector* in het het testprogramma nadat de instanties `e1` en `e2` op `null` zijn gezet? Stel dat we vóór het afsluiten van de `main` methode nog een keer een telling van de aanwezige `employee` objecten opvragen, wat zal dan de uitkomst zijn? Geef een toelichting.