

Imperatief Programmeren, eerste deeltentamen (INFOIMP) 29 september 2006

Schrijf op elk ingeleverd blad je naam.

Schrijf op het eerste blad ook je studentnummers en het aantal ingeleverde bladen.

De lijst met standaardfuncties na afloop graag weer inleveren; de opgaven mag je houden.

Er zijn in totaal 100 punten te behalen.

Opgave 1

(25 punten)

Onderdelen a-d zijn tekstvragen.

Houd het antwoord kort: een of twee zinnen per onderdeel kan al genoeg zijn.

- a) Alice ontwikkelt op een Windows-computer een Java-applet, en zet die op een webpagina. Bob surft met een Mac-computer naar deze pagina en bekijkt de applet.

Beschrijf wie daarbij een compiler en/of een interpreter gebruikt, en welke soorten code daardoor vertaald worden.

- b) Welke relatie bestaat er tussen het begrip *klasse* en het begrip *object*?

En welke tussen het begrip *klasse* en het begrip *methode*?

Wat is het verband tussen die twee relaties?

- c) Wat is een *void* methode?

Waarin wijkt de manier waarop je zo'n methode aanroept af van de aanroep van een niet-void methode?

- d) Wat is een *static* methode?

Waarin wijkt de manier waarop je zo'n methode aanroept af van de aanroep van een niet-static methode?

- e) gegeven zijn de volgende declaraties:

```
int n;  
double d;  
String s;
```

Schrijf onderstaande zes toekennings-opdrachten over, en vul ze waar nodig aan met type-conversies:

```
n = d;  
n = s;  
d = n;  
d = s;  
s = n;  
s = d;
```

Opgave 2

(25 punten)

Het antwoord op deze vraag heeft een vaste vorm: twee kolommen van ieder acht getallen of tekens. Een toelichting is niet nodig.

Hieronder staan 16 programma-fragmenten. Geef voor elk fragment apart aan *hoe vaak* de methode 'iets' wordt aangeroepen. Als de methode oneindig vaak wordt aangeroepen, schrijf dan het symbool ∞ op. Als het fragment een fout bevat die niet door de compiler geaccepteerd wordt, schrijf dan de letter **F** op.

<pre>x=0; while (x<5) { x=x+1; this.iets(); } </pre>	<pre>for (x=0; x<5; x++) { this.iets(); x=x+1; } </pre>
<pre>x=0; while (x<5) { this.iets(); x+1; } </pre>	<pre>for (x=0; x<5; x++) this.iets(); x=x+1; </pre>
<pre>x=0; while (x=5) { this.iets(); x=x+1; } </pre>	<pre>for (x=0; x<x;) this.iets(); </pre>
<pre>x=0; while (x<5); { this.iets(); x=x+1; } </pre>	<pre>for (x=0; x<5) { this.iets(); x=x+1; } </pre>
<pre>x=0; {while (x<5) x=x+1; this.iets(); } </pre>	<pre>for (x=0; x<5; x++) for (y=0; y<5; y++) this.iets(); </pre>
<pre>x=0; {while (x<5) {; this.iets(); x=x+1; }; } </pre>	<pre>for (x=0; x<5; x++) { for (y=0; y<5; y++) this.iets(); this.iets(); } </pre>
<pre>x=0; while (x<5) this.iets(); x=x+1; </pre>	<pre>for (x=0; x<5; x++) this.iets() for (y=0; y<5; y++) this.iets(); </pre>
<pre>x=0; while (x<5); this.iets(); x=x+1; </pre>	<pre>for (x=0; x<5; x++) for (y=0; y<x; y++) this.iets(); </pre>

Opgave 3 en 4 vragen een stukje programma.

Kleine schrijffoutjes (hoofdletters, puntkomma's enz.) worden niet streng afgerekend, maar de elementen die de structuur van het programma bepalen (haakjes, accolades, aanhalingstekens enz.) zijn wel belangrijk. Schrijf die dus duidelijk en op de goede plaats op!

Het is toegestaan (maar niet nodig) om Java-constructies die (nog) niet zijn behandeld toch te gebruiken.

Opgave 3

(25 punten)

Een benadering van 'cosinus hyperbolicus' van een reëel getal x kun je berekenen door:

$$1 + x^2/2! + x^4/4! + x^6/6! + x^8/8! + x^{10}/10! + \dots$$

De notatie '6!' betekent '6 faculteit', dat is alle gehele getallen van 1 tot en met 6 met elkaar vermenigvuldigd.

Schrijf een methode `coshyp` die deze benadering berekent door 20 van deze termen te sommeren, en dat als resultaat oplevert.

Het is hierbij niet toegestaan om de bestaande methodes uit de klasse `Math` te gebruiken. Ook is het niet toegestaan om alle 20 termen helemaal uit te schrijven.

Java heeft geen ingebouwde machtsverhef-operator, dus die kun je ook niet gebruiken.

Je mag wel (maar hoeft niet) zelf extra hulp-methoden definiëren.

Opgave 4

(25 punten)

In deze opgave moet je een complete Java-applet schrijven. Je hoeft alleen de Java-file te schrijven, dus een HTML-file is niet nodig. Verder mag je weglaten:

- de `import`-regels
- methodes die eigenlijk nodig zijn, maar een lege body hebben

Schrijf een applet die een zwart vierkantje van 20×20 beeldpunten laat meebewegen met de bewegingen van de muis. De muis-cursor wijst het midden van het vierkant aan. Zie het eerste plaatje hieronder.

Elke keer als de gebruiker met de muis klikt, wordt het vierkant groter. Het vierkant krijgt er bij elke klik een 'ring' bij, met een steeds lichtere kleur grijs. Het geheel blijft meebewegen met de muis. Het tweede plaatje toont de situatie na 5 muiskliks.

Na 7 kliks is de buitenste ring zeer licht grijs geworden. Bij de 8^e klik begint de volgende ring weer bij zwart, en daarna worden de ringen weer lichter grijs. Het derde plaatje toont de situatie na 11 muiskliks.

Het vierde plaatje toont de situatie na 25 muiskliks. Bij de 8^e, 16^e en 24^e ring verspingt de kleur steeds weer naar zwart. Een deel van het plaatje wordt buiten beeld getekend; dat gaat vanzelf, dus daar hoeft je niet apart rekening mee te houden.

