

Imperatief Programmeren (INFOIMP) 26 september 2008

N.B. Bij dit tentamen hoort een lijst met standaardfuncties.

Opgave 1

Deze opgave bevat vijf tekstvragen. Houd het antwoord kort: een of twee zinnen per onderdeel kan al genoeg zijn.

- Wat wordt er in een Java-methode aangeduid met **this**? In welke situatie is het niet toegestaan om **this** te gebruiken?
- Geef drie voorbeelden van een *primitief type* en drie voorbeelden van een *objecttype*. Wat is het verschil tussen de manier waarop een waarde in een variabele met een *primitief type* en in een variabele met een *objecttype* wordt opgeslagen?
- Beschrijf het verschil tussen een variabele die is gedeclareerd in de *header* van een methode en een variabele die is gedeclareerd in de *body* van een methode.
Beschrijf ook het verschil tussen een variabele die is gedeclareerd in de *body* van een *methode*, en een variabele die is gedeclareerd boven in een *klasse*.
- Wanneer is een programmeertaal een *imperatieve* programmeertaal?
- Beschrijf welke rol *broncode* (*source code*) en *bytecode* spelen voor de programmeur, en voor de gebruiker van een Java-programma.

Opgave 2

Het antwoord op deze vraag heeft een vaste vorm: een blok met vijf keer drie woorden. Een toelichting is niet nodig.

Hieronder staan 15 fragmenten uit een programma (in een blok van 5 bij 3). Maak op je antwoordblad ook een blok van 5 bij 3 vakjes, en zet in elk vakje een woord passend bij het overeenkomstige fragment:

- **type** als het programmafragment een **type** is
- **expr** als het programmafragment een **expressie** is
- **opdr** als het programmafragment een **opdracht** is
- **decl** als het programmafragment een **declaratie** is

<code>long s;</code>	<code>long</code>	<code>s.length()</code>
<code>new Color(0,0,0).brighter()</code>	<code>g.setColor(new Color(0,0,0));</code>	<code>Color c1, c2;</code>
<code>b=new Button("");</code>	<code>Button</code>	<code>new Button("b")</code>
<code>Math/PI</code>	<code>double pi;</code>	<code>pi*=3.14159;</code>
<code>x+1/1+x</code>	<code>x==x+1</code>	<code>x++;</code>

Opgave 3

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.

- a) Schrijf een methode `laatste` met een getal als parameter. Je mag zonder controle aannemen dat de parameter 0 of groter is.

De methode moet als resultaat het laatste cijfer opleveren van de parameter. Bijvoorbeeld: de aanroep `this.laatste(123)` levert 3 op.

- b) Alle studenten hebben een studentnummer van 7 cijfers. Maar niet elk getal van 7 cijfers is een geldig studentnummer. Om te controleren of er geen tikfouten zijn gemaakt bij het invoeren van studentnummers, wordt in administratieve software de volgende controle gedaan:

Het eerste cijfer wordt vermenigvuldigd met 7, het tweede cijfer wordt vermenigvuldigd met 6, het derde cijfer met 5, enzovoort, en het laatste cijfer met 1. Alle uitkomsten worden opgeteld. Als het totaal deelbaar is door 11, is het een geldig studentnummer. Bijvoorbeeld: voor de controle van 0304158 wordt uitgerekend: $0 \times 7 + 3 \times 6 + 0 \times 5 + 4 \times 4 + 1 \times 3 + 5 \times 2 + 8 \times 1 = 55$, en dat is inderdaad deelbaar door 11.

Deze controle werkt ook voor 9-cijferige bankrekeningnummers. Nu wordt het eerste cijfer met 9 vermenigvuldigd, het tweede met 8, enzovoort, en het laatste cijfer weer met 1. (De controle werkt alleen niet voor studentnummers die met een 'F' beginnen, en ook niet voor gironummers).

Schrijf een methode `geldig` met een getal als parameter. De methode moet als resultaat opleveren of de parameter een geldig nummer is volgens bovenstaande controle. Het moet zowel voor studentnummers als voor bankrekeningnummers gebruikt kunnen worden.

Opgave 4

In deze opgave moet je een complete Java-applet schrijven. Je mag de import-regels weglaten.

Schrijf een applet met onderstaande specificaties.

Er wordt een patroon getekend zoals in onderstaande afbeelding. (Je kunt het beschouwen als om elkaar heen liggende kaders, of ook als op elkaar gestapelde vierkanten).

Het binnenste vierkant is wit, en is 20x20 beeldpunten groot. De dikte van de kaders eromheen is 10 beeldpunten. Het buitenste kader is zwart, en de kaders daartussen hebben gelijkmatig verlopende grijstinten. Het midden van alle vierkanten wordt gespecificeerd in de HTML-file. Voor de afbeelding hieronder werd gebruikt:

```
<applet code="Tent200809.class" width="300" height="250">
<param name="midden" value="180 120"></param>
</applet>
```

zodat het midden van de vierkanten 180 beeldpunten van de linkerrand staat, en 120 beeldpunten van de bovenrand. De applet moet natuurlijk ook voor andere waarden werken. De twee getallen staan samen in één `<param>`, gescheiden door een spatie.

Boven in het window staat een schuifregelaar.

Met de schuifregelaar bepaalt de gebruiker het aantal kaders.

Aan het begin zijn er 5 kaders zichtbaar (de witte kern meegeteld).

Met de schuifregelaar helemaal naar rechts zijn er 10 kaders.

Met de schuifregelaar helemaal naar links zijn er 2 kaders.

