

1e deeltentamen Imperatief Programmeren

1 december 2009, 9.00-11.00 uur

Schrijf op elk ingeleverd blad je naam.

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

De lijst met standaardfuncties na afloop graag weer inleveren;

de opgaven mag je houden.

Opgave 1

Deze opgave bevat vijf tekstvragen (meestal met twee deelvragen).

Houd het antwoord kort: een zin per onderdeel kan al genoeg zijn.

- a. Bekijk het volgende programmafragment:

```
int n = 5;
double a, b, c, d;
a = n * 1.5;
b = n * 3 / 2;
c = n / 2 * 3;
d = 3 / 2 * n;
```

Wat zijn de waarden van a, b, c en d na het uitvoeren van deze opdrachten?

- b. Wat is de betekenis van het woord `static` in een methode-header?
Hoe wijkt de aanroep van zo'n methode af van de aanroep van methodes die niet `static` zijn?
- c. Variabelen kunnen worden gedeclareerd met een primitief type of met een klasse als type.
Noem 3 primitieve types, en geef bij elk type een voorbeeld van een waarde van dat type.
Wat is de waarde van een variabele met een klasse als type, en wat is het verschil in de manier waarop zo'n waarde wordt opgeslagen vergeleken met de waarde van een variabele met een primitief type?
- d. Schrijf een programmafragment met hetzelfde effect als de opdracht:

```
for (t=0; t<100; t++)
    g.drawString("hoi",10,10*t);
```


maar dan zonder een `for`-opdracht en zonder de `++` operator te gebruiken.
- e. Wat is het verschil tussen een *compiler* en een *interpreter*?
Op welke manier wordt een Java-programma door dit soort tools verwerkt?

Opgave 2

Het antwoord op deze vraag heeft een vaste vorm:

een blok met drie keer vijf letters. Een toelichting is niet nodig.

Hieronder staan 15 fragmenten uit een programma (in een blok van 3 bij 5).

Schrijf op je antwoordblad ook een blok van 3 bij 5 vakjes,

en zet in elk vakje een letter passend bij het overeenkomstige fragment:

- **T** als het programmafragment een **type** is
- **E** als het programmafragment een **expressie** is
- **O** als het programmafragment een **opdracht** is
- **D** als het programmafragment een **declaratie** is
- **H** als het programmafragment een **methode-header** is
- **X** als het programmafragment geen van bovenstaande dingen is

<code>double</code>	<code>double x;</code>	<code>(double)x*x</code>
<code>void x()</code>	<code>x==y+1</code>	<code>y=x!=x;</code>
<code>a%=x;</code>	<code>2xa0</code>	<code>0x2a</code>
<code>Button b=ok;</code>	<code>Button</code>	<code>new Button("ok")</code>
<code>this.add(b);</code>	<code>String ok</code> <code>(Button b)</code>	<code>class OK</code> <code>extends Applet</code>

Opgave 3

Het is in deze opgave **niet** toegestaan om de bestaande methodes uit de klasse `Math` te gebruiken.

a.
De wiskundige notatie '5!' betekent '5 faculteit', dat is alle gehele getallen van 1 tot en met 5 met elkaar vermenigvuldigd. Dus de uitkomst van 5! is 120.
Je kunt de faculteit van elk positief geheel getal uitrekenen. Voor het getal 0 geldt de uitkomst $0! = 1$.

Schrijf een methode `faculteit`, die de faculteit van een niet-negatief geheel getal berekent.

b.
Schrijf een methode `macht`, die x^n uitrekent voor een reëel grondtal x en een gehele, niet-negatieve macht n .

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

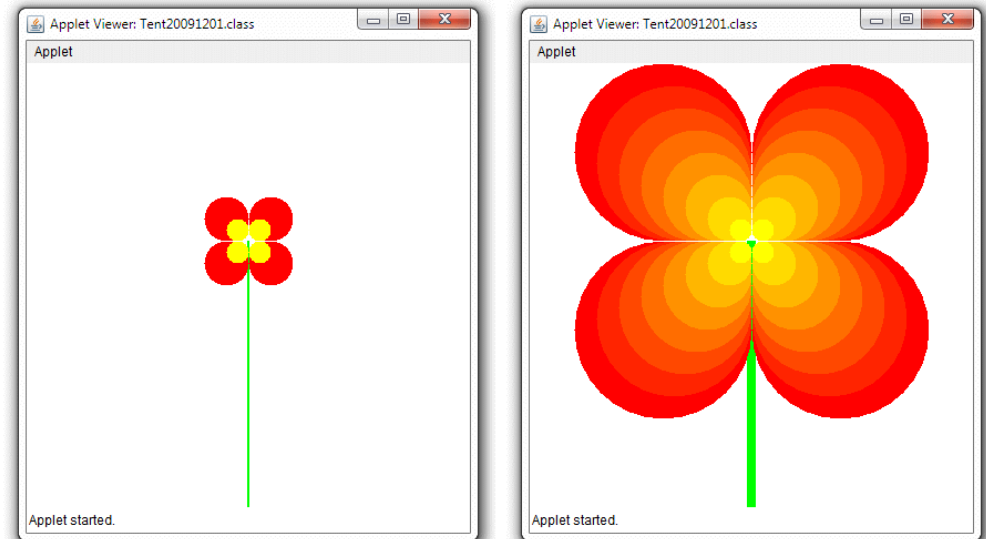
$$1 + x^2 / 2! + x^4 / 4! + x^6 / 6! + x^8 / 8! + x^{10} / 10! + \dots$$

Schrijf een methode `coshyp` die deze benadering berekent door 10 van deze termen te sommeren, en dat als resultaat oplevert.

Het is niet toegestaan om alle 10 termen helemaal uit te schrijven.
Je mag wel de methoden uit onderdeel a en b aanroepen.

Opgave 4

Schrijf een applet die een bloem met steel toont, zoals in de afbeeldingen hieronder.
In de HTML-file wordt vastgelegd waar het midden van de bloem staat.



Aan het begin bestaat elk van de vier bloemblaadjes uit twee cirkels (afbeelding links). Elke keer dat de gebruiker klikt met de muis (op een willekeurige plaats) komt er een cirkel bij. De afbeelding rechts toont de situatie na 6 keer klikken.

De kleinste cirkels hebben een doorsnede van 20 pixels, de volgende cirkel 40 pixels, en door klikken komen er cirkels van 60 pixels, 80 pixels enz. bij.
De kleinste cirkels zijn geel van kleur, de grootste cirkels zijn rood, en de cirkels daartussen hebben de tussenliggende tinten oranje.
De steel is groen, en wordt bij elke klik iets dikker.

Schrijf het Java-programma.

Je mag de import-regels, en methodes die een lege body hebben, weglaten.

Hint: teken de cirkels van groot naar klein, voor het gewenste 'stapel'-effect.

Schrijf ook het ontbrekende deel van de HTML-file:

```
<HTML>
<HEAD><TITLE>Bloem</TITLE></HEAD>
<BODY>
  dit gedeelte schrijven
</BODY>
</HTML>
```