



Voor alle opgaven in dit tentamen zijn de volgende declaraties
gegeven:

```
Output out = new Output();  
Input in = new Input();
```

Er kan in opgaven waar dit gewenst is dus zonder meer gebruik
gemaakt worden van de variabelen in en out voor inlezen respectievelijk
afdrukken.

Opgave 1.

a) Wat is, gegeven de class

```
class X {  
    String s;  
  
    X () {  
        s = "geinitialiseerd";  
    }  
}
```

en de methodes

```
void m1 (char c) {  
    c = '@';  
    out.println(c);  
}  
  
void m2 (String s) {  
    s = s + ".";  
    out.println(s);  
}  
  
void m3 (X x) {  
    x.s = "veranderd";  
    out.println(x.s);  
}
```

de uitvoer van de onderstaande statements?

```
char ch = '#';  
String tekst = "Dit is een zin";  
X x = new X();  
  
out.println(ch);  
out.println(tekst);  
out.println(x.s);  
  
m1(ch);  
m2(tekst);  
m3(x);
```

```
out.println(ch);  
out.println(tekst);  
out.println(x.s);
```

b) Gegeven is de onderstaande heading van een methode

```
double benaderCosinus (double x, int aantalTermen)
```

Deze methode moet $\cos(x)$ (de cosinus van x radialen) benaderen
m.b.v.
de volgende ontwikkeling

$$\cos(x) = x^0/0! - x^2/2! + x^4/4! - x^6/6! + x^8/8! - \dots$$

Hierbij is x^i de notatie voor "x tot de macht i" en $n!$ (n
faculteit)
de notatie voor $1 \times 2 \times 3 \times \dots \times (n-1) \times n$
Het aantal uit te rekenen termen is gelijk aan de waarde van de
formele parameter `aantalTermen`.
Programmeer deze methode. Ga uit van: `aantalTermen >= 1`.

c) Schrijf een methode "double functie (double x)" voor de berekening
van
de onderstaande formule.

$$(7/8 * \text{"sinus van x"}) / \text{"x tot de derde macht"}$$

De sinus van x kan worden uitgerekend met de static methode
"double sin(double x)" uit de class Math.

d) Declareer een variabele "matrix" die een vierkant twee-dimensiaal
array van integers met 5 rijen en 5 kolommen bevat.
Doe dit gestructureerd, d.w.z. gebruik indien nodig constanten.

Schrijf een methode "diagonalenOK" die voor een willekeurig
vierkant
twee-dimensiaal array van integers kan controleren of dit array
de
eigenschap heeft dat op geen van beide diagonalen in rij i de
waarde i
staat.

VOORBEELD:

De linker van de onderstaande matrices heeft de eigenschap
"diagonalenOK" omdat geen van de diagonalen in een rij het nummer
van
die rij bevat.
De rechter van de twee onderstaande matrices heeft NIET de
eigenschap
"diagonaal" omdat de diagonaal van linksboven naar rechtsonder in
rij 1
getal 1 bevat.

0 1 2	1 2 3
3 4 5	4 5 6
6 7 8	7 8 9

Z.O.Z. voor vervolg opgave 1

e) Gegeven is het programma in de onderstaande class:

```
class Opgave_1e {  
    Output out;  
    int a = 3,  
        b = -2;  
  
    Opgave_1e() {  
        out = new Output();  
    }  
  
    int methode1 (int b, int c) {  
        int a = c;  
        b = a + c;  
        c = b * a;  
        methode2(c);  
        out.println(a);  
        out.println(c);  
        return b;  
    }  
  
    void methode2(int b) {  
        b = b + a;  
        a = a + b;  
        out.println(a);  
        out.println(b);  
    }  
  
    void start() {  
        b = methode1(a, b);  
        methode2(a);  
        out.println(a);  
        out.println(b);  
    }  
  
    public static void main(String argv[]) {  
        new Opgave_1e().start();  
    }  
}
```

welke uitvoer geeft dit programma?

Opgave 2.

Gegeven is de onderstaande class 'Werknemer'.

```
class Werknemer {  
    String naam,  
        double salaris;  
    int aantalJarenInDienst;  
}
```

a) Voeg aan de class Werknemer een constructor zonder parameters toe die het werknemer-object initialiseert op een nieuwe naamloze werknemer, d.w.z. dat het String-object als startwaarde de lege string krijgt en

dat de primitieve variabelen 'salaris' en 'aantalJarenInDienst' respectievelijk de startwaarden 0.0 en 0 krijgen.

b) Voeg nu aan de class Werknemer een tweede constructor met 2 parameters toe die een nieuwe werknemer met een bekende naam, een salaris > 0.0 en 0 dienstjaren aanmaakt. De eerste parameter bevat de naam van de werknemer en de tweede parameter diens salaris.

c) Maak een class Bedrijf welke, gebruik makend van de in de onderdelen a) en b) van deze vraag uitgebreide class Werknemer, een object definieert dat maximaal 2500 werknemers kan bevatten en een methode voegToe bevat om een werknemer aan het bedrijf toe te voegen.

d) Voeg aan de class Bedrijf de onderstaande methode toe

```
int aantalWerknemersDatSalarisverhogingKrijgt ()
```

Deze methode telt hoeveel werknemers in aanmerking komen voor een salarisverhoging. Een werknemer komt in aanmerking voor een salarisverhoging indien het salaris minder is dan 50.000 en het aantal dienstjaren meer is dan 10 of (ongeacht het salaris) als het aantal jaren in dienst groter is dan 25.

Pak dit gestructureerd aan door eerst aan de class Werknemer een methode 'krijgtSalarisverhoging' toe te voegen. Denk ook aan het gebruik van constanten.

Opgave 3.

a) Schrijf een recursieve methode codeer() die een stuk tekst inleest dat begint met een '(' en eindigt met een ')' en de characters tussen de ronde haakjes codeert in een int. De tekst zal alleen bestaan uit hoofdletters en ronde haakjes. Haakjes komen altijd in paren voor en kunnen genest zijn. Bijvoorbeeld "(AB(CD)E)". Gegeven is dat de leeswijzer bij aanroep van de methode een '(' aanwijst.

Voor de codering van een hoofdletters wordt de het volgnummer van die hoofdletter in het alfabet gebruikt. Vervolgens wordt dit volgnummer vermenigvuldigt met het volgnummer binnen het haakjespaar. De sommatie van alle zo verkregen coderingen van hoofdletters levert de codering van het haakjespaar op.

Voorbeeld: De codering van (FDE) is:

$$\begin{aligned}
 \text{code (FDE)} &= 1 \cdot \text{code F} + 2 \cdot \text{code D} + 3 \cdot \text{code E} \\
 &= 1 \cdot 6 + 2 \cdot 4 + 3 \cdot 5 \\
 &= 29
 \end{aligned}$$

Als een haakjespaar 1 een ander haakjespaar 2 bevat, wordt dit haakjespaar 2 op de al beschreven manier gecodeerd en vermenigvuldigd

met het volgnummer ervan binnen haakjespaar 1. Hierbij telt haakjespaar 2, net zoals een hoofdletter, voor 1 volgnummer.

Voorbeeld: De codering van (GED(FDE)A) is:

$$\begin{aligned}
 \text{code (GED(FDE)A)} &= 1 \cdot 7 + 2 \cdot 5 + 3 \cdot 4 + 4 \cdot \text{code (FDE)} + 5 \cdot 1 \\
 &= 1 \cdot 7 + 2 \cdot 5 + 3 \cdot 4 + 4 \cdot 29 + 5 \cdot 1 \\
 &= 150
 \end{aligned}$$

En de code van (ZZ(GED(FDE)A)(FDE)) is:

$$\begin{aligned}
 \text{code (ZZ(GED(FDE)A)(FDE))} &= 1 \cdot 26 + 2 \cdot 26 + 3 \cdot \text{code (GED(FDE)A)} + 4 \cdot \text{code (FDE)} \\
 &= 1 \cdot 26 + 2 \cdot 26 + 3 \cdot 150 + 4 \cdot 29 \\
 &= 644
 \end{aligned}$$

Waardering

Opgave	a	b	c	d	e	f	totaal
1.	4	4	2	4		18	
2.	2	2	4	4		10	
3		8					8
							-- +
							36

Het eindcijfer E volgt uit het puntentotaal T als volgt : $E = T / 4 + 1$