

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 afdrucken.

Opgave 1.

a) Gegeven de onderstaande class

```
class X {
    boolean b;
    X () {
        b = false;
    }
}
```

en de onderstaande methodes

```
void m1 (int i) {
    i = 2 * i;
    out.println(i);
}

void m2 (String s) {
    s = s + "abc";
    out.println(s);
}

void m3 (X x) {
    x.b = true;
    out.println(x.b);
}
```

wat is dan de uitvoer van de onderstaande statements?

```
int i = 10;
String s = "123";
X x = new X();

out.println(i);
out.println(s);
out.println(x.b);

m1(i);
m2(s);
m3(x);

out.println(i);
out.println(s);
out.println(x.b);
```

b) Gegeven is dat de inhoud van een bol met straal r bekend kan worden met de formule $(4/3) \times \pi \times r^3$ "derde macht van de straal" en dat er in de class Math een constante π zit met de waarde van π .

Schrijf een methode "inhoudbol" welke voor een willekeurige straal de inhoud van de bol met deze straal uitrekent en retourneert.

double benaderEengeheelDoore (int aantalTermen)

Deze methode moet $1/e$ (e is het getal van Euler) benaderen m.b.v. de volgende ontwikkeling

$$1/e = 1/0! - 1/1! + 1/2! - 1/3! + 1/4! - 1/5! + 1/6! - \dots$$

Hierbij is $n!$ (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.

Declareer een variabele "matrix" die een twee-dimensionaal array van integers met 3 rijen en 6 kolommen bevat.

Doe dit gestructureerd, d.w.z. gebruik indien nodig constanten.

Schrijf een methode "matigGevuld" die voor een willekeurig twee-dimensionaal arrays van integers kan controleren of dit array de eigenschap "matig gevuld" heeft.

Voor deze opgave is gedefinieerd dat een matrix de eigenschap matig gevuld heeft als de elementen van deze matrix voor meer dan 50% bestaan uit nullen.

d) Gegeven is het programma in de onderstaande class:

```
class Opgave_1d {
    Output out;
    int a = 4,
    b = 2;
```

```
Opgave_1d() {
    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_1d().start();
}
```

Welke uitvoer geeft dit programma?

Opgave 2.

Gegeven is de onderstaande class 'Auto'.

```
class Auto {
    String merk,
    int    aantalGeredenkilometers;
}
```

- a) Voeg aan de class Auto een constructor zonder parameters toe die het auto-object initialiseert op een nieuwe merkloze auto, d.w.z. dat het String-object als startwaarde de lege string krijgt en dat de primitieve variabele 'aantalGeredenkilometers' als startwaarde de waarde 0 krijgt.

- b) Voeg nu aan de class Auto een tweede constructor met 1 parameter toe die een nieuwe auto van een bekend merk aanmaakt. de ene parameter bevat de merknaam van de auto.

- c) Maak een class Parkeergarage welke, gebruik makend van de in de onderdelen a) en b) van deze vraag uitgebreide class Auto, een object definieert dat maximaal 500 auto's kan bevatten en een methode voegtoe bevat om een auto aan de parkeergarage toe te voegen.

- d) Voeg aan de class parkeergarage een methode 'aantalNieuwex' toe die het aantal auto's van merk "X" met een minder dan 10000 gereden kilometers telt.
Pak dit gestructureerd aan door eerst aan de class Auto een methode 'nieuwex' toe te voegen.

Opgave 3.

- a) Schrijf een recursieve methode "somcijfers" om de som van de cijfers van een willekeurig positief geheel getal te kunnen bepalen. De stopconditie is dat het getal uit 1 cijfer bestaat. Bedenk zelf wat de recursiestap is.

- b) Schrijf een recursieve methode "palindroom" die retourneert of de waarde van een meegegeven String-object een palindroom bevat. Bedenk zelf welke parameters deze recursieve methode nodig heeft.

Een palindroom is een rij characters die van links naar recht en van rechts naar links gelezen dezelfde reeks characters oplevert.

Voorbeelden van String-waarden die palindromen bevatten zijn:

```
"a"
"aa"
"pop"
"lepel"
"patretrap"
```

Waardering							
Opgave	a	b	c	d	e	f	totaal
1.	3	2	3	4	4		16
2.	2	2	4	4			12
3.	3	5					8
							-- +
							36

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