

Opgave 1.

- a) Gegeven zijn de onderstaande classes

```
class Naam {
    String s;

    Naam (String naam) {
        s = naam;
    }

    void maakLeeg () {
        s = "LEEG";
    }
}

class Auteur {
    Naam naam;

    Auteur (String s) {
        naam = new Naam(s);
    }

    void putNaam (Naam naam) {
        this.naam = naam;
    }
}

en een programma dat de classes Naam en Auteur gebruikt waarin de
volgende declaraties staan

PrintStream out = new PrintStream(System.out);
Naam naam = new Naam("XXX");
Auteur auteur = new Auteur("ZZZ");

void m1 (Naam par1, String par2) {
    par1.s = par2;
    par2 = "RRR";
}
```

```
void m2 (Auteur par1, Naam par2) {
    par1.putNaam(par2);
    par2.maakLeeg();
}
```

```
void print () {
    out.printf("%s - %s\n", naam.s, auteur.naam.s);
}
```

Wat is nu de uitvoer van een aanroep van de hierna gegeven methode  
doelsets() die in hetzelfde programma staat als de gegeven declaraties?

```
void doeIets () {
    naam.maakLeeg();
    print();
    m1(naam, "VVV");
    print();
    m2(auteur, naam);
    print();
    m1(naam, "FFF");
    print();
    m2(auteur, naam);
    print();
}
```

- b) Schrijf voor  $n > 1$  de methode : double sommeerTermen (int x, int n)  
die de sommatie uitrekt van de eerste n termen van de onderstaande  
in wiskunde genoteerde berekening
- $$1/4 \cdot x^2 - 1/6 \cdot x^4 + 1/8 \cdot x^6 - 1/10 \cdot x^8 + 1/12 \cdot x^{10} - \dots$$
- Hierbij staat de notatie  $a^b$  voor a tot de macht b.  
Het is niet toegestaan om methodes uit de class Math te gebruiken.
- c) Declareer een variabele "matrix" die een twee-dimensinaal array van  
integers met 15 rijen en 9 kolommen bevat.  
Doe dit gestructureerd, d.w.z. gebruik indien nodig constanten.
- Schrijf een methode isPositief() die voor een willekeurig 2-dimensinaal  
array van integers aangeeft of dit array de eigenschap positief heeft.  
Voor deze opgave is gedefinieerd dat een array de eigenschap positief  
heeft indien het array meer positieve dan negatieve getallen bevat.
- Het array -1 0 0 0 heeft niet de eigenschap positief omdat er  
0 0 0 0 precies evenveel negatieve als positieve  
0 0 0 0 0 getallen in het array staan (van beide 1).  
0 0 0 0 1
- Het array 2 3 4 5 is wel positief omdat er 7 positieve en 2 negatieve  
-1 0 0 0 getallen in staan.  
-4 8 9 9
- d) Gegeven is het programma in de onderstaande class:
- ```
class OpgaveId {
    PrintStream out;
    int a, b;

    OpgaveId() {
        out = new PrintStream(System.out);
        a = 3;
        b = -4;
    }

    void print (int x, int y) {
        out.printf("%d\n%d\n", x, y);
    }

    int m1(int b) {
        int a = 2 * b;
        b += 1;
        print(a, b);
        return a;
    }

    int m2 (int c, int a) {
        a = b / 3;
        int b = c + 2;
        c = m1(b);
        print(a, b);
        return c;
    }

    void start() {
        b = m1(a);
        a = m2(b, a);
        print(a, b);
    }

    public static void main(String argv[]) {
        new OpgaveId().start();
    }
}
```
- Geef de uitvoer van dit programma.

## Opgave 2.

- a) Gegeven is de onderstaande class welke gebruikt kan worden voor het opslaan van smsberichten van niet meer dan 150 tekens.

```
class SmsBericht {
    static final int MAX_AANTAL_TEKENS = 150;

    String bericht;
    int aantalTekens;

    // De constructors en methodes doen er voor deze opgave niet toe.
}
```

Voeg aan de class SmsBericht de methode

```
int aantal (String woord)
```

toe die retourneert hoe vaak het meegegeven woord voorkomt in het smsbericht. Om het probleem wat simpeler te maken mag er vanuit gegaan worden dat een smsbericht alleen woorden en spaties bevat.

HINT: Maak een Scanner-object waarmee de woorden uit het bericht gelezen kunnen worden.

- b) Maak een class MobileTelefoon waarmee een object gedefinieerd wordt waarin maximaal 50 SmsBericht-objekten kunnen worden opgeslagen.

Voorzie de class MobileTelefoon van een methode voegToe() om een SmsBericht-object toe te kunnen voegen aan een MobileTelefoon-object.

- c) Voeg aan de class MobileTelefoon een methode spam() toe die, in een nieuw MobileTelefoon object, alle spamberichten retourneert. Voor deze opgave is gedefinieerd dat een smsbericht een spambericht is als het woord "koop" meer dan 4 keer in het bericht staat.

Pak dit gestructureerd aan. Voeg indien er behoefte is aan methodes die iets doen met een SmsBericht-object deze methodes toe aan de class SmsBericht.

## Opgave 3.

- a) Schrijf een recursieve methode maximum() die voor een willekeurig array van int's het maximum van de elementen kan bepalen.

Houd er rekening mee dat een array 0 elementen kan bevatten.

Bedenk zelf welke parameters de methode maximum() nodig heeft.

Er mag worden aangenomen dat het array volledig gevuld is.

HINT1: Het maximum van 0 getallen is -oneindig, oftewel, in Java,

```
Integer.MIN_VALUE
```

HINT2: Het maximum van twee integers kan worden bepaald met de methode max() uit de class Math.

Zo geldt: "Math.max(-5, 16) == 16" en "Math.max(10, 34) == 34".

- b) Gegeven is dat de class String een methode substring() als volgt bevat:

```
String substring (int beginindex, int eindindex)
```

welke de substring van de gehele String retourneert vanaf de gegeven beginindex tot de gegeven eindindex, m.a.w. van beginindex t/m eindindex-1

Voorbeelden:

```
"12345".substring(0, 3).equals("123")
"12345".substring(2, 3).equals("3")
"12345".substring(1, 4).equals("234")
"12345".substring(0, 5).equals("12345")
"12345".substring(4, 4).equals("")
```

Schrijf nu een recursieve oplossing voor de onderstaande methode binnensteVooraan()

```
String binnensteVooraan (String s)
```

die van de meegegeven String s een nieuwe String maakt bestaande uit het middelste character van s (bij een oneven aantal characters) of de twee middelste characters (bij een even aantal characters), gevolgd door de characters die daar voor en achter stonden, gevolgd door de characters die daar weer voor en achter stonden, enz.

Voorbeelden:

```
binnensteVooraan("").equals("")
binnensteVooraan("c").equals("c")
binnensteVooraan("ce").equals("ce")
binnensteVooraan("AdB").equals("dAB")
binnensteVooraan("PceQ").equals("cePQ")
binnensteVooraan("Pc@eQ").equals("@cePQ")
binnensteVooraan("+Pc@eQ+").equals("@cePQ++")
```

Bij het schrijven van de methode binnensteVooraan() mag gebruik gemaakt worden van de methode substring() uit de class String.

Waardering

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

Voor alle andere studenten:

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