

Voor studenten die het tentamen Inleiding Programmeren II uit de
opleiding Medische Natuurwetenschappen doen geldt:

- het tentamen bestaat uit de opgaven 1 en 2, opgave 3 hoeft niet gemaakt te worden
 - de tijdsduur is 2:30 uur
- Voor alle andere studenten geldt:
- het tentamen bestaat uit alle drie de opgaven
 - de tijdsduur is 3:00 uur

Opgave 1.

a) Gegeven zijn de onderstaande classes

```
class P {
    char c;

    P () {
        c = '1';
    }

    P (char c) {
        this.c = c;
    }

    void insert (char c) {
        this.c = c;
    }

    public String toString () {
        return "" + c;
    }
}

class Q {
    P p;
    String s;

    Q () {
        p = new P();
        s = p.toString();
    }

    Q (char c) {
        p = new P(c);
        s = p.toString();
    }

    void insert (char symbol) {
        p.insert(symbol);
        s = p.toString();
    }
}
```

en de declaraties

```
Outly    out = new Output();
Q q = new Q('a');
```

```
void m1 (Q q) {
    P localeP = q.p;
    q.insert('A');
    out.println(localeP.c);
    out.println(q.p.c);
}
```

```
localeP.c = 'b';
out.println(localeP.c);
out.println(q.p.c);
```

```
concat(q.s, "?");
out.println(q.s);
out.println(q.p.toString());
```

```
void concat (String s1, String s2) {
    s1 = s1 + s2;
}
```

Wat wordt er bij dit gegeven door de hierna volgende statements afgedrukt?

```
out.println(q.p.toString());
out.println(q.s);
m1(q);
out.println(q.p.toString());
out.println(q.s);
```

b) Schrijf voor $n \geq 1$ de methode : double sommeerTermen (int x, int n) die de sommatie uitrekent van de eerste n termen van de onderstaande in wiskunde genoteerde berekening

$$1/2 \cdot x^2 - 1/3 \cdot x^3 + 1/4 \cdot x^4 - 1/5 \cdot x^5 + 1/6 \cdot x^6 - \dots$$

Hierbij staat de notatie a^b voor a tot de macht b. Het is niet toegestaan om methodes uit de class Math te gebruiken.

Declareer een variabele "matrix" die een twee-dimensionaal array van integers met 6 rijen en 4 kolommen bevat. Doe dit gestructureerd, d.w.z. gebruik indien nodig constanten.

Schrijf een methode aantalToppen() die voor een willekeurig twee-dimensionaal arrays van integers kan tellen hoe vaak in dit array een "top" voorkomt .

Voor deze opgave is een "top" gedefinieerd als een getal op een regel dat zowel groter is als het voorgaande getal en alsook groter is als het volgende getal op die regel.

voorbeeld:

in de matrix 1 2 3 2 1 zitten 2 "toppen", namelijk de 3 op de 1e regel
5 7 5 7 5 en beide 7's op de 2e regel.
5 4 3 2 1
1 2 3 4 5



d) Gegeven is het programma in de onderstaande class:

```
class OpgaveId {
    Output out;
    int a = 4;
    b = 5;

    OpgaveId() {
        out = new Output();
    }

    void print (int a, int b) {
        out.println(a);
        out.println(b);
    }

    int methode1(int b) {
        a = a + b;
        b = a / b;
        print(a, b);
        return a;
    }

    int methode2 (int b, int c) {
        int a = b;
        c = a - c;
        b = methode1(c);
        print(c, a);
        return b;
    }

    void start() {
        a = methode1(b);
        b = methode2(a, b);
        print(a, b);
    }

    public static void main(String argv[]) {
        new OpgaveId().start();
    }
}
```

Opgave 2.

a) Gegeven is de onderstaande class welke gebruikt kan worden voor het opslaan van de tekst van boeken met niet meer dan 100.000 woorden.

```
class Boek {
    static final int MAX_AANTAL_WOORDEN = 100000;
    String[] woorden;
    aantalwoorden;

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

Voeg aan de class Boek de methode

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

toe welke telt hoe vaak het meegegeven woord voorkomt in het boek en dit aantal retourneert.

b) Maak een class Bibliotheek waarmee een object gedefinieerd wordt waarin maximaal 10000 Boek-objekten kunnen worden opgeslagen.

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

c)

Voeg aan de class Bibliotheek een methode ScienceFiction() toe die, in een nieuw Bibliotheek object, alle science fiction romans retourneert. Voor deze opgave is gedefinieerd dat een boek een roman is als het aantal woorden groter is dan 50.000 en dat een boek een science fictionboek is als het woord ruimeschip er meer dan 20 keer in staat.

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

Opgave 3.

a) Geef een recursieve implementatie van de onderstaande methode

```
int aantalZevens (int getal)
```

die retourneert hoe vaak het cijfer 7 voorkomt in het meegegeven getal.

```
voorbeelden:   aantalZevens(777)   == 3
               aantalZevens(1357531) == 1
               aantalZevens(3)    == 0
```

b)

Programmeer een recursieve methode sort() welke m.b.v. insertion sort een willekeurig array van het type int[] monotoon niet-dalend kan sorteren.

schematische beschrijving insertion sort:

In een ongesorteerde rij getallen met lengte n wordt 1 getal op de juiste plaats gezet wat resulteert in een gesorteerde deelrij met lengte 1 en een ongesorteerde deelrij met lengte n-1. Vervolgens wordt de hiervoor beschreven aanpak herhaalt voor de ongesorteerde deelrij, d.w.z. 1 getal ervan wordt op de juiste plaats gezet, wat resulteert in een ongesorteerde deelrij met lengte n-2 en een gesorteerde deelrij met lengte 2. Het herhalen van deze aanpak gaat door totdat de hele rij gesorteerd is.

Waardering				
Opgave	a	b	c	d
1.	4	4	4	4
2.	4	2	6	2
3.	3	5		8
				-- +
				36

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