

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 X {
    char c;

    X () {
        c = '$';
    }

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

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

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

class Y {
    X x;
    String s;

    Y () {
        x = new X();
        s = x.toString();
    }

    Y (char c) {
        x = new X(c);
        s = x.toString();
    }

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

en de declaraties

```
Output out = new Output();
Y y = new Y('a');
```

```
void m1 (Y y) {
    X localX = y.x;
    localX.c = 'Z';
    out.println(localX.c);
    out.println(y.x.c);
}
```

```
y.insert('Y');
out.println(localX.c);
out.println(y.x.c);
```

```
y.s = concat(y.s, y.s);
out.println(y.s);
out.println(y.x.toString());
```

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

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

```
out.println(y.x.toString());
out.println(y.s);
m1(y);
out.println(y.x.toString());
out.println(y.s);
```

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

$$1/3 \cdot x - 1/4 \cdot x^2 + 1/5 \cdot x^3 - 1/6 \cdot x^4 + 1/7 \cdot x^5 - \dots$$

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 12 rijen en 8 kolommen bevat. Doe dit gestructureerd, d.w.z. gebruik indien nodig constanten.

Schrijf een methode aantalToppen() die voor een willekeurig twee-dimensionaal array van integers kan tellen hoe vaak in dit array een "top" voorkomt. Voor deze opgave is een "top" gedefinieerd als een getal in een kolom dat zowel groter is als het voorgaande getal en alsook groter is als het volgende getal in die kolom.

voorbeeld:

```
in de matrix 1 2 3 2 1 zitten 4 "toppen", namelijk de 7 in de 2e kolom
              5 7 5 7 5           de 5 in de 3e kolom
              5 4 3 2 1           de 7 in de 4e kolom
              1 2 9 4 8           de 5 in de 5e kolom
```

d) Gegeven is het programma in de onderstaande klasse.

```

class OpgaveId {
    Output out;
    int a = -2;
    b = -3;

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

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

    int method1(int b) {
        a = a - b;
        b = b / a;
        print(a, b);
        return a;
    }

    int method2 (int b, int c) {
        int a = b;
        c = a - c;
        b = method1(c);
        print(c, a);
        return b;
    }

    void start() {
        a = method1(b);
        b = method2(a, b);
        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 de tekst van boeken met niet meer dan 60.000 woorden.

```
class Boek {
    static final int MAX_AANTAL_WOORDEN = 60000;
    String[] woordrij;
    int aantalWoorden;

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

Voeg aan de class Boek de methode

```
int langWoorden ()
```

toe welke telt hoeveel "lange" woorden er in het boek staan en dit aantal retourneert. Voor deze opgave is een lang woord gedefinieerd

als een woord met meer dan 10 letters.

b) Maak een .ass Boekenplank waarmee een object gedefinieerd wordt waarin maximaal 40 Boek-objecten kunnen worden opgeslagen.

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

c) Voeg aan de class Boekenplank een methode `MoestijlkeBoeken()` toe die, in een nieuw Boekenplank object, alle "moestijlke" boeken retourneert. Voor deze opgave is gedefinieerd dat een boek moestijlke is als meer dan 10% van de woorden een lang woord is.

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) Schrijf een recursieve methode `printXreeks()` om de X-reeks voor gehele getallen $n \geq 1$ af te drukken, waarbij de X-reeks van een getal gedefinieerd is als:

```
X-reeks(n) = 1 + X-reeks(n / 2) * X-reeks(n / 2)
als n > 1
```

Voorbeelden: n x-reeks

[illegible]

b) Programmeer een recursieve boolean methode aanwezig() welke m.b.v. binary search van een willekeurig array van het type int[], dat monoton niet-dalend gesorteerd is, kan zeggen of een gezochte int waarde aanwezig is.

Bedenk zelf welke heading deze methode aanwezig() nodig heeft..

Wardering

Opgrave	a	b	c	d	total
1.	4	4	4	4	16
2.	4	2	6		12
3.	3	5			8
					-
					+
					36

Voor MNW-studenten:

Het eindcijfer E volgt uit het puntentotaal T als volgt : $E = T \cdot 9 / 28 + 1$

Voor alle andere studenten:

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