

- N.B. MMW-studenten maken de opgaven 1 en 2 en krijgen hiervoor 2 uur de tijd -

Opgave 1.

a) Gegeven zijn de onderstaande classes

```
class D {
    StringBuffer sb;

    D (char c) {
        sb = new StringBuffer();
        add(c);
    }

    void add (char c) {
        sb.append(c);
    }

    }

class E {
    D d;

    E () {
        d = new D('A');
    }

    E (D d) {
        this.d = d;
    }

    void add (char c) {
        d.add(c);
    }

    }

}

en een programma dat de classes D en E gebruikt waarin de
volgende declaraties staan

Output out = new Output();
D var1 = new D('X');
E var2 = new E(),
var3 = new E(var1);

void m1 (StringBuffer sb) {
    sb = new StringBuffer("new1");
}

void m2 (D d) {
    d.sb = new StringBuffer("new2");
}

void m3 (StringBuffer sb) {
    sb.append('Z');
}

void print (D par1, E par2, E par3) {
    out.println(par1.sb);
    out.println(par2.d.sb);
    out.println(par3.d.sb);
}
```

```
void doeiets () {
    print(var1, var2, var3);
    var3.add('B');
    print(var1, var2, var3);
    m1(var2.d.sb);
    out.println(var2.d.sb);
    m2(var2.d);
    out.println(var2.d.sb);
    m3(var1.sb);
    print(var1, var2, var3);
}
```

Wat wordt er afgedrukt als in het programma de methode doeiets() wordt aangeroepen?

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

$$0 - (1+1)/(x^3) + (2+3)/(x^6) - (3+4)/(x^9) + (4+5)/(x^{12}) + \dots$$

Hierbij staat " x^n " voor " x tot de macht n " en " $n!$ " voor " n faculteit" ofwel " $1*2*3*4*...*(n-1)*n$ ".

Er mag geen gebruik gemaakt worden van methodes uit de class Math.

c) Declareer een variabele "matrix" die een twee-dimensionaal array van integers met 11 rijen en 33 kolommen bevat. Doe dit gestructureerd, d.w.z. gebruik indien nodig constanten.

Schrijf een int methode maxDiagonaal() die voor een willekeurig twee-dimensionaal array van positieve gehele getallen via een functieresultaat retourneert:
 - het maximum van de getallen op de twee diagonalen indien de matrix vierkant is;
 - het getal 0 indien de matrix niet vierkant is.

Voorbeeld: van de matrix

8	2	3
1	7	5
9	4	3

d) Gegeven is het programma in de onderstaande class:

```
class OpgaveId {
    Output out;
    int a, b, c;

    OpgaveId() {
        out = new Output();
        a = 2;
        b = -2;
        c = -1;
    }

    void print (int p, int q, int r) {
        out.println(p);
        out.println(q);
        out.println(r);
    }

    void methode1(int c, int b) {
        a = b + c;
        b = c + a;
        c = a + b;
        print(a, b, c);
    }
}
```

```

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

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

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

```

Geef de uitvoer van dit programma.

Opgave 2.

- a) Gegeven zijn de onderstaande classes welke gebruikt kunnen worden voor het opslaan van gegevens van tentamens.

```

class Tentamen {
    String naamStudent,
        naamVak;
    double eindcijfer;

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

```

```

class Onderwijsbureau {
    static final int MAX_AANTAL_TENTAMENS = 1000;
    Tentamen[] tentamenArray;
    int aantalTentamens;
    ...
}

```

Schrijf een default constructor en een methode voegToe() voor de class Onderwijsbureau. De default constructor moet het Onderwijsbureau-object initialiseren op leeg d.w.z. op 0 geregistreerde tentamens. De methode voegToe moet de mogelijkheid bieden om 1 Tentamen-object aan een Onderwijsbureau-object toe te voegen.

N.B. Voor de onderdelen b) t/m d) van deze opgave geldt dat deelproblemen in een aparte methode en in de juiste class geprogrammeerd dienen te worden.

- b) Voorzie de class Onderwijsbureau van een methode
- ```
int aantalOnvoldoendes ()
```

welke retourneert voor hoeveel tentamens een onvoldoende gehaald is. Voor deze opgave is gedefinieerd dat voor een tentamen een onvoldoende gehaald is als het eindcijfer lager dan 5,5 is.

- c) Voeg aan de class Onderwijsbureau een functie zoek() toe. Deze methode krijgt via een parameter de naam van een vak gegeven en moet in een nieuw Onderwijsbureau object alle voor dat vak gedane tentamens retourneren.

- d) Voorzie de class Onderwijsbureau van een methode

```
int aantalOnvoldoendes (String naamVak)
```

welke retourneert voor hoeveel van de op het onderwijsbureau geregistreerde tentamens voor het vak 'naamVak' een onvoldoende gehaald is. Implementeer deze methode met de voor onderdeel b) en c) geprogrammeerde methodes.

- e)

Hoe heet het principe dat het mogelijk maakt dat de class Onderwijsbureau twee verschillende methodes met de naam "aantalOnvoldoendes" bevat?

## Opgave 3.

- a) Schrijf een recursieve methode welke voor een willekeurig positief geheel getal n alle positieve even getallen <=n afdruckt.

- b)

Het spel "Jump It" bestaat uit een rij met n positieve gehele getallen voorafgegaan door een 0, bijvoorbeeld (n=5): 0 - 3 - 80 - 6 - 57 - 10

Het doel van het spel is om met een zo laag mogelijke som van de gepasseerde getallen van de 0 naar het laatste getal te lopen.

Bij iedere stap naar een nieuw getal kan er voor gekozen worden om naar het volgende getal te gaan (bijv. van de 0 naar de 3) of om het volgende getal over te slaan (bijv. van de 0 naar de 80).

In de gegeven rij kan er dus o.a. van de 0 naar het laatste getal (10) gelopen worden door steeds het volgende getal te nemen, de som is dan

$$0 + 3 + 80 + 6 + 57 + 10 = 156$$

Een tweede manier van lopen is die waarbij de 80 wordt overgeslagen:

$$0 + 3 + 6 + 57 + 10 = 76$$

Ook kunnen zowel de 80 als de 57 worden overgeslagen:

$$0 + 3 + 6 + 10 = 19$$

Enz. enz.

Geef een recursieve implementatie van de onderstaande methode jumpIt() welke, volgens de hierboven geschetste spelregels van Jump It, de minimale som van alle mogelijke manieren om van het eerste naar het laatste getal te lopen retourneert.

```
int jumpIt (int[] rij)
```

## Waardering

| Opgave | a | b | c | d | e | totaal |
|--------|---|---|---|---|---|--------|
| 1.     | 4 | 3 | 4 | 4 |   | 15     |
| 2.     | 2 | 4 | 4 | 3 | 1 | 14     |
| 3.     | 3 | 4 |   |   |   | 7      |

-- +  
36

Voor MNW-studenten:

Het eindcijfer E volgt uit het puntentotaal T als volgt :  $E = T * 9 / 29 + 1$

Voor alle andere studenten:

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