

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

Opgave 1.

a) Gegeven zijn de onderstaande classes

```
class Naam {
    String s;

    Naam (String s) {
        this.s = s;
    }

    int lengte () {
        return s.length();
    }
}

class V {
    Naam naam;
    int i;

    V () {
        naam = new Naam("start");
        i = naam.lengte();
    }

    void put (Naam n) {
        naam = n;
        i = naam.lengte();
    }
}

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

Output out = new Output();
Naam naam = new Naam("aaa");
V v = new V();

void m1 (V v, Naam n, String s1, String s2) {
    n = new Naam(s1 + s2);
    v.put(n);
    print(v, n);
}

void m2 (V v, Naam n, String s) {
    n.s = s;
    v = new V();
    print(v, n);
}

void print (V v, Naam naam) {
    out.println(v.naam + ", " + v.i);
    out.println(naam);
}
```

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

```
void doelsets () {
    print(v, naam);
    m1(v, naam, "abc", "def");
    print(v, naam);
    m2(v, naam, "XXX");
    print(v, naam);
}
```

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 genoemde berekening

$$2/3 \cdot (x-2) - 3/6 \cdot (x-3) + 4/9 \cdot (x-4) - 5/12 \cdot (x-5) + \dots$$

c)

Declareer een variabele "matrix" die een twee-dimensionaal array van
integers met 9 rijen en 7 kolommen bevat.
Doe dit gestructureerd, d.w.z. gebruik indien nodig constanten.

Schrijf een boolean methode maxKolom() met o.a. een parameter "int k",
die voor een willekeurig twee-dimensionaal array van integers via een
functieresultaat antwoord geeft op de vraag of het getal met kolomindex

voor de voorbeeldmatrix 8 2 3 2 1 en k=0 is de uitkomst 'true'
9 7 5 7 5 bij 1 ≤ k ≤ 4 is de uitkomst 'false'
5 4 3 2 1

d)

Gegeven is het programma in de onderstaande class:

```
class Opgave1d {
    Output out;
    int a = -2;
    b = 3;

    Opgave1d () {
        out = new Output();
    }

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

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

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

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

    public static void main(String argv[]) {
        new Opgave1d().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 studenten.

```
class Student {
    String naam;
    double ectsPerJaar;
    int aantalJaarIngeschreven.
}
```

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

```
class Faculteit {
```

```
    static final int MAX_AANTAL_STUDENTEN = 1500;
```

```
    Student[] studentArray;
```

```
    int aantalStudenten;
```

```
    ...
```

```
}
```

Schrijf een default constructor en een methode voegToe() voor de class Faculteit. De default constructor moet het Faculteit-object initialiseren op een lege faculteit met 0 studenten waaraan maximaal 1500 studenten kunnen worden toegevoegd. De methode voegToe moet de mogelijkheid bieden om 1 Student-object aan een faculteit-object toe te voegen.

- b) Voorzie de class Faculteit van een methode

```
int aantalGoedeStudenten ()
```

welke retourneert hoeveel studenten van de faculteit goed studeren. Voor deze opgave is gedefinieerd dat een student goed studeert als het aantal ects per jaar van die student groter is dan 55.

Indien je een deelprobleem herkent, programmeer dit dan in een aparte methode in de juiste class.

- c) Voeg aan de class Faculteit een methode selectie() toe die, in een nieuw Faculteit object, alle topstudenten retourneert. Voor deze opgave is een topstudent gedefinieerd als een goede student die minimaal 4 jaar is ingeschreven.

Indien je een deelprobleem herkent, programmeer dit dan in een aparte methode in de juiste class.

Opgave 3.

- a) Schrijf een recursieve methode

```
int over (int n, int k) // gegeven n >= k >= 0
```

voor de berekening van over(n, k). De formule over(n, k) is als volgt gedefinieerd

```
over (n, k) = { over(n-1, k)*over(n-1, k-1)  als n>k=1
               { 1                          als n=k (n, k > 0)
               { 1                          als k=0
```

- b)

Gegeven is de onderstaande hulpmethode

```
String aantalKeer (char c, int x) {
    String resultaat = "";
    for (int i = 0; i < x; i++) {
        resultaat += c;
    }
    return resultaat;
}
```

Geef een recursieve oplossing voor de onderstaande methode

```
String voegHaakjesToe (String s, int i)
// s bevat alleen cijfers, 0 <= i <= s.length()
```

die voor een String die alleen cijfers bevat, voor alle cijfers vanaf het cijfer op de positie i, ronde haakjes openen en sluiten toevoegt. Het aantal haakjes openen en sluiten is gelijk aan de waarde van het cijfer (bij het cijfer 6 worden er dus 6 ronde haakjes openen en 6 ronde haakjes sluiten toegevoegd). Alle ronde haakjes sluiten komen aan het eind van de resultaatstring.

Voorbeelden:

de aanroep voegHaakjesToe("0", 0)	levert	"0"
voegHaakjesToe("1", 0)		"(1)"
voegHaakjesToe("2", 0)		"((2))"
voegHaakjesToe("12", 0)		"(1((2)))"
voegHaakjesToe("231", 0)		"((2(((3(1))))))"
voegHaakjesToe("231", 1)		"(((3(1))))"
voegHaakjesToe("231", 2)		"((1))"
voegHaakjesToe("231", 3)		"(((((40)))))"

N.B. De waarde van de variabele i na de assignment

```
int i = '5' - '0';
```

is 5.

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

Waardering

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

Voor MNW-studenten:

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

Voor alle andere studenten:

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