

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

Opgave 1.

a) Gegeven zijn de onderstaande classes

```
class G {
    String s;
    int i;

    G (String s) {
        this.s = s;
        i = s.length();
    }

    void put (String s) {
        this.s = s;
        i = s.length();
    }
}

class H {
    G g

    H () {
        g = new G("noot");
    }

    H (G g) {
        this.g = g;
    }

    void put (String s) {
        g.put(s);
    }
}

en een programma dat de classes G en H gebruikt waarin de
volgende declaraties staan

Output out = new Output();
G var1 = new G("aap");
H var2 = new H();
var3 = new H(var1);

void m1 (String s) {
    s = "wim";
}

void m2 (G g) {
    g.s = "zus";
}

void print (G par1, H par2, H par3) {
    out.println(par1.s);
    out.println(par2.g.s);
    out.println(par3.g.s);
}
```

```
void doeiets () {
    print(var1, var2, var3);
    var3.put("mies");
    print(var1, var2, var3);
    m1(var2.g.s);
    out.println(var2.g.s);
    m2(var2.g);
    out.println(var2.g.s);
    var1.put("jet");
    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

$$(x^2)/(2!) - (x^3)/(3!) + (x^4)/(4!) - (x^5)/(5!) + \dots$$

Hierbij staat " x^n " voor " x tot de macht n " en " $n!$ " voor " n faculteit" oftewel " $1 \cdot 2 \cdot 3 \cdot 4 \cdot \dots \cdot (n-1) \cdot n$ ".

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

Declareer een variabele "matrix" die een twee-dimensionaal array van integers met 4 rijen en 11 kolommen bevat.

Doe dit gestructureerd, d.w.z. gebruik indien nodig constanten.

Schrijf een boolean methode maxRijgemiddelde() die voor een willekeurig twee-dimensionaal array van integers via een functieresultaat het maximum van de rijgemiddeldes retourneert. Een rijgemiddelde is het gemiddelde van de getallen in een rij. Zo is het rijgemiddelde van de rij 6 6 7 7 7 8 6 6 gelijk aan 6,625

Doe dit gestructureerd. Programmeer de problemen in een hulpmethode.

Voorbeeld: van de matrix 8 2 3 2 1 zijn de rijgemiddeldes 3,2 9 7 5 7 5 5 4 3 2 1 3,0

Het maximum van de rijgemiddeldes bij dit voorbeeld is dus: 6,6

d) Gegeven is het programma in de onderstaande class:

```
class Opgave1d {
    Output out;
    int a, b, c;

    Opgave1d() {
        out = new Output();
        a = 4;
        b = -3;
        c = -5;
    }

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



```

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

int methode2 (int c, int a) {
    c = c + a;
    b = b - b;
    a = 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 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 boeken.

```

class Boek {
    String schrijver;
    double verkoopprijs; // in euro
    int aantalbladzijden.
}

```

```

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

```

```

class Boekenkast {
    static final int MAX_AANTAL_BOEKEN = 800;
    Boek[] boekenArray;
    int aantalBoeken;
}

```

```

}

Schrijf een default constructor en een methode voegToe() voor de
class Boekenkast. De default constructor moet het Boekenkast-object
initialiseren op een lege boekenkast met 0 boeken waaraan maximaal
800 boeken kunnen worden toegevoegd. De methode voegToe moet de
mogelijkheid bieden om 1 Boek-object aan een Boekenkast-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 Boekenkast van een methode
 int aantalGoedkopeBoeken ()

welke retourneert hoeveel boeken in de boekenkast goedkoop zijn.
 Voor deze opgave is gedefinieerd dat een boek goedkoop is als de gemiddelde prijs per bladzijde minder dan euro 0,015 bedraagt.

- c) Voeg aan de class Boekenkast een functie zoek() toe. Deze methode krijgt via een parameter de naam van een schrijver gegeven en moet in een nieuw Boekenkast object alle boeken van die schrijver die in de boekenkast staan retourneren.

- d) Voorzie de class Boekenkast van een methode

```

int aantalGoedkopeBoeken (String schrijver)

```

welke retourneert hoeveel boeken in de boekenkast, die door de via de parameter gegeven schrijver geschreven zijn, goedkoop zijn.

- e) Hoe heet het principe dat het mogelijk maakt dat de class Boekenkast twee verschillende methodes met naam "aantalGoedkopeBoeken" bevat?

Opgave 3.

- a) Schrijf een recursieve methode voor de berekening van het n-de driehoeksgetal (n>0). De driehoeksgetallen zijn als volgt gedefinieerd

```

driehoeksgetal(1) = 1
driehoeksgetal(2) = 1 + 2
driehoeksgetal(3) = 1 + 2 + 3
driehoeksgetal(4) = 1 + 2 + 3 + 4
....
driehoeksgetal(n) = 1 + 2 + 3 + 4 + ... + (n-1) + n

```

- b) Geef een recursieve implementatie van de onderstaande methode zoek(). Deze methode geeft m.b.v. linear search antwoord op de vraag of het meegegeven getal x aanwezig is in het eveneens meegegeven array a.

Bedenk zelf welke de derde parameter is die deze recursieve methode nodig heeft.

```

boolean zoek (int[] a, int x, ....)

```

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-boeken:

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