

Studenten MNW krijgen 1:50 uur de tijd voor dit tentamen en maken alleen de opgaven 1 en 2.

Opgave 1.

```
a) Gegeven is de onderstaande class A

class A {
    char c;

    A(char c) {
        this.c = c;
    }
}
```

en een programma waarin de volgende statements/declaraties staan

```
static final int MAX_LENGTH_RIJ = 20;
```

```
Output out = new Output();
A[] rij = new A[MAX_LENGTH_RIJ];
for (int i = 0; i < rij.length; i++) {
    rij[i] = new A('K');
}
StringBuffer sb = new StringBuffer("abc");
A a = new A('A');
```

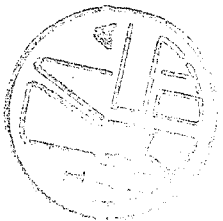
```
void m(A par) {
    par = new A('m');
```

```
void n(A[] par) {
    for (int i = 0; i < par.length; i++) {
        par[i] = new A('n');
    }
}
```

```
void o(A[] par) {
    par = new A[MAX_LENGTH_RIJ];
    for (int i = 0; i < par.length; i++) {
        par[i] = new A('o');
    }
}
```

```
void p(StringBuffer par) {
    sb = new StringBuffer("o");
}
```

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



```
void doarlets () {
    out.println(a.c);
    m(a);
    out.println(a.c);
    out.println(rij[0].c);
    m(rij[0]);
    out.println(rij[0].c);
    n(rij);
    out.println(rij[0].c);
    o(rij);
    out.println(rij[0].c);
    p(sb);
    out.println(sb);
    sb.setCharAt(0, 'x');
    out.println(sb);
}
```

b) Gegeven is de onderstaande heading van een methode

```
double benaderCosinus (double x, int aantalTermen)

Deze methode moet cos(x) (de cosinus van  $x$  radialen) benaderen m.b.v. de volgende ontwikkeling
```

$$\cos(x) = x^0/0! - x^2/2! + x^4/4! - x^6/6! + x^8/8! - \dots$$

Hierbij is x^i de notatie voor " x tot de macht i " en $n!$ (n faculteit) de notatie voor $1 \times 2 \times 3 \times \dots \times (n-1) \times n$. Het aantal uit te rekenen termen is gelijk aan de waarde van de formele parameter `aantalTermen`.

Programmeer deze methode zonder gebruik te maken van methodes uit de class `Math`. Ga uit van: `aantalTermen >= 1`.

c) Declareer een variabele "matrix" die een twee-dimensionaal array van integers met 6 rijen en 11 kolommen bevat. Doe dit gestructureerd, d.w.z. gebruik indien nodig constanten.

Schrijf een boolean methode `som()` die die voor een willekeurig twee-dimensionaal array van integers via een functieresultaat retourneert: -1 indien het array niet vierkant is en de som van de elementen die niet op een diagonaal staan indien het array wel vierkant is.

Voorbeelden:

```
voor de arrays 2 3 2 1 en 1 2 3 4 zijn de uitkomsten respectievelijk
7 5 7 5      5 6 7 8
4 3 2 1      9 0 1 2
              3 4 5 6
```

d) Gegeven is het programma in de onderstaande class:

```
class Opgave1d {
    Output out;
    int a = -4,
        b = 2;

    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(b);
    print(b, a);
    return b;
}

void start() {
    a = methode1(b);
    b = methode2(b, 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 gegevens van producten.

```

class Product {
    String omschrijving;
    double prijsPerStuk;
    int aantalVerkocht;
    char code;

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

```

Maak een class Groothandel zodanig dat een instantie van deze class de gegevens van verschillende producten kan bevatten (maximaal van 500 verschillende producten). Verder moet de class Groothandel een default constructor en een methode voegToe() bevatten. De default constructor moet het Groothandel-object initialiseren op een lege groothandel. De methode voegToe() moet de mogelijkheid bieden om 1 product-object toe te voegen.

- b) Voorzie de class Groothandel van een methode
- ```
double omzet (char code)
```

welke de totale omzet van de groothandel retourneert door van alle producten met de meegegeven code de omzet te sommeren. De omzet van een product is het door de verkoop van dit product ontvangen bedrag.

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

- c) Voeg aan de class Groothandel een methode selectie() toe die, in een nieuw Groothandel object, alle goedlopende producten retourneert. Voor deze opgave is een goedlopend product gedefinieerd als een product waarvan de omzet groter dan 750 is.

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

## Opgave 3.

- a) Schrijf een recursieve methode welke voor een willekeurig positief geheel getal n alle positieve oneven getallen <n afdrukt.
- 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 | totaal |
|--------|---|---|---|---|--------|
| 1.     | 4 | 4 | 4 | 4 | 16     |
| 2.     | 2 | 4 | 6 |   | 12     |
| 3.     | 3 | 5 |   |   | 8      |
|        |   |   |   |   | 36     |

Voor MNW-studenten volgt het eindcijfer E uit het puntentotaal T als volgt:  $E = (T * 9) / 28 + 1$

Voor alle andere studenten volgt het eindcijfer E volgt uit het puntentotaal T als volgt:  $E = T / 4 + 1$