

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 M

```
class M {
    String s

    M () {
        s = "K";
    }

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

    void add (String s) {
        this.s += s;
    }

    void verdubbel () {
        s += s;
    }
}
```

en een programma waarin de volgende statements/declaraties staan

```
static final int MAX_LENGTH_RIJ = 4;
PrintStream out = new PrintStream(System.out);
M m = new M();
```

```
String s = "p";
M[] rij = new M[MAX_LENGTH_RIJ];
for (int i = 0; i < rij.length; i++) {
    rij[i] = new M(s);
}

void m1 (M m) {
    m = new M("12");
}

void m2 (M[] r, int i) {
    r[i] = new M("34");
}

void m3 (M m) {
    m.s = "5";
    m.add("6");
}

void println (M m) {
    out.printf("%s\n", m.s);
}
```

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

```
void doeleets () {
    println(m);
    println(rij[2]);
    m1(m);
    println(m);
    println(rij[2]);
    m2(rij, 2);
    println(m);
    println(rij[2]);
    m3(m);
    println(m);
    println(rij[2]);
    m.verdubbel();
    println(m);
    println(rij[2]);
}
```

b) Gegeven is de onderstaande heading van een methode y()

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

Deze methode moet de onderstaande reeks met oneindig veel termen in "aantalTermen" termen benaderen

$$y(x) = x/3 + x^3/(5*21) - x^5/(7*31) + x^7/(9*41) - x^9/(11*51) + \dots$$

Hierbij is x^i de notatie voor " x tot de macht i " en $x!$ de notatie voor " x faculteit".

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 en $x \geq 0$.

Declareer een variabele "matrix" die een twee-dimensionaal array van integers met 13 rijen en 17 kolommen bevat.

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

Schrijf een boolean methode bijzonder() die voor een willekeurig twee-dimensionaal array van positieve integers via het functieresultaat retourneert of dit array de eigenschap bijzonder heeft.

Voor deze opgave heeft een twee-dimensionaal array van integers met k rijen de eigenschap bijzonder indien voor iedere rij i ($1 \leq i \leq k-1$) geldt dat de som van de getallen in rij i kleiner is dan de som van de getallen in rij $i+1$.

Voorbeelden: array 1 4 7 10 is bijzonder en array 1 4 7 10 niet

2 5 8 11	2 5 7 1
3 6 9 12	3 6 9 12

Iet op! De rij met nummer i heeft als index $(i-1)$.

d) Gegeven is het programma in de onderstaande class:

```
import java.io.PrintStream;
class Opgave1d {
    PrintStream out;
    int p, q;

    Opgave1d() {
        out = new PrintStream(System.out);
        p = -3;
        q = 4;
    }

    void print (int a, int b) {
        out.printf("%d\n", a);
        out.printf("%d\n", b);
    }

    void methode1 (int p) {
        int q = p + 5;
        p = -p;
        print(p, q);
    }

    int methode2 (int q) {
        int p = q + 1;
        q -= 1;
        methode1(p);
        print(p, q);
        return q/p;
    }

    void start() {
        methode1(q);
        print(p, q);
        q = methode2(p);
        print(p, q);
    }

    public static void main(String argv[]) {
        new Opgave1d().start();
    }
}

Geef de uitvoer van dit programma.
```

Opgave 2.

a) Gegeven is de onderstaande class welke gebruikt kan worden voor het opslaan van gegevens over tentamens.

```
class Tentamen {
    String naamStudent, naamVak, opleiding;
    int studentnummer;
    double eindcijfer;
    // De constructors en methodes doen er voor deze opgave niet toe.
}
```

Maak een class Tentamenzaal zodanig dat in een instantie van deze class kan worden bijgehouden welke tentamens in een zaal gemaakt zijn (maximaal 100). Verder moet de class Tentamenzaal een default constructor en een methode voegToe() bevatten. De default constructor moet een Tentamenzaal-object initialiseren op een lege tentamenzaal. De methode voegToe() moet de mogelijkheid bieden om 1 Tentamen-object toe te voegen.

b) Voorzie de class Tentamenzaal van een methode

Tentamenzaal opleiding (String o)
welke in een nieuw Tentamenzaal-object de tentamens retourneert van de studenten met opleiding o.

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

c) Gegeven is dat de class Tentamenzaal een methode

Tentamenzaal naamVak (String v)

bevat welke in een nieuw Tentamenzaal-object de tentamens retourneert van de studenten die tentamen hebben gedaan voor vak v.

Voeg nu aan de class Tentamenzaal een methode

Tentamenzaal selectie (String o, String v)

toe, die, gebruik makend van de methodes opleiding() en naamVak() uit de class Tentamenzaal, in een nieuw Tentamenzaal-object de tentamens retourneert van de studenten van opleiding o die tentamen gedaan hebben in vak v.

N.B. De methode opleiding() mag ook gebruikt worden indien deze in onderdeel b) niet geprogrammeerd is.

Opgave 3.

a) Schrijf een recursieve methode "int aantalOnevenCijfers (int n)" welke voor een willekeurig positief geheel getal n retourneert hoeveel oneven cijfers dit getal bevat. Voorbeeld: de getallen 1357931 en 1234567 bevatten respectievelijk zeven en vier oneven cijfers.

b) Schrijf een recursieve methode voor het afdrukken van de volgende rij tekens:

```
p(1) = (a)
p(2) = (a(aa)a)
p(3) = (aaa(aa(a)a)aaa)
p(4) = (aaaa(aa(a(a)aa)aaa)aaaa)
```

De heading van de methode moet er als volgt uitzien:

```
void p (int i) // i > 0
```

Verder is gegeven: `PrintStream out = new PrintStream(System.out);`

```
void printAantalKeer (char c, int x) {
    for (int i = 0; i < x; i++) {
        out.printf("%c", c);
    }
}
```

Waardering				
Opgave	a	b	c	d
1.	4	4	4	4
2.	2	6	4	4
3.	4	4		

Voor MMW-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: $E = T / 4 + 1$