

Opgave 1.

a) Geef de uitvoer van het programma Opgave1.

```
class A {
    int a;

    A () {
        a = 1;
    }

    void m1 (int i) {
        a = a * i;
    }

    public String toString () {
        return "" + a;
    }
}

class B {
    A a;

    B () {
        a = new A();
    }

    void m2 (int a) {
        this.a.a = a;
    }

    public String toString () {
        return a.toString();
    }
}

class Opgave1 {
    A a;
    B b;
    Output out;

    Opgave1 () {
        b = new B();
        a = b.a;
        out = new Output();
    }

    void start () {
        out.println(a);
        out.println(b);
        a.m1(5);
        out.println(a);
        out.println(b);
        b.m2(3);
        out.println(a);
        out.println(b);
        a.a = 7;
        out.println(a);
        out.println(b);
        b.a = a;
        out.println(a);
        out.println(b);
    }

    public static void main (String[] argv) {
        new Opgave1().start();
    }
}
```

b) Gegeven is de onderstaande heading van een methode

```
double benaderLnXPlus1 (double x, int aantalTermen)
```

Deze methode moet $\ln(x+1)$ benaderen m.b.v. de volgende ontwikkeling

$$\ln(x+1) = x - 1/2 \cdot x^2 + 1/3 \cdot x^3 - 1/4 \cdot x^4 + 1/5 \cdot x^5 - \dots$$

Hierbij is x^i de notatie voor "x tot de macht i".

Het aantal uit te rekenen termen is gelijk aan de waarde van de formele parameter aantalTermen.

Programmeer deze methode. Ga uit van aantalTermen >= 1 en $x > 0$.

Declareer een variabele "matrix" die een tweedimensionaal array van integers met 5 rijen en 5 kolommen bevat.

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

Schrijf een methode "gebalanceerdMatrix" die voor een willekeurige vierkante matrix van integers kan controleren of deze matrix de eigenschap "gebalanceerd" heeft.

Voor deze opgave wordt de "linkerdiagonaal" gedefinieerd als de elementen op de diagonaal van linksboven naar rechtsonder.

Evenszins alleen voor deze opgave wordt de eigenschap gebalanceerd gedefinieerd als dat de som van de getallen boven de linkerdiagonaal gelijk is aan de som van de getallen onder de linkerdiagonaal.

d) Welke uitvoer geeft het onderstaande programma?

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

    OpgaveId() {
        out = new Output();
        a = 5;
        b = 3;
    }

    void method1 (int b) {
        int a = b + 1;
        b = a * 2;
        method2(b);
        out.println(a);
        out.println(b);
    }

    void method2 (int a) {
        int b = a - 1;
        a = b + 2;
        b = a + b;
        out.println(a);
        out.println(b);
        return a;
    }

    void start() {
        b = method2(b);
        method1(a);
        out.println(a);
        out.println(b);
    }

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

Opgave 2.

- Maak een class AudioCD voor het opslaan van de volgende gegevens over de muziek op een audio-CD: naamCD, speeltijd en naamuitvoerders. Voorzie de class AudioCD van een constructor met parameters om alle velden van de class AudioCD te initialiseren bij het aanmaken van een nieuw AudioCD object.
- Maak een class CDRek welke een object definieert waarin maximaal 50 AudioCD objecten kunnen worden opgeslagen. Voorzie de class CDRek van een methode voegToe() om een AudioCD object toe te kunnen voegen aan het CDRek object.
- Voeg aan de class CDRek de onderstaande methode zoek() toe die, gegeven de naam van de uitvoerders, in een nieuw CDRek object alle AudioCD objecten retourneert met muziek van de opgegeven uitvoerders.

CDRek zoek (String uitvoerders)

Pak dit gestructureerd aan door o.a. aan de class AudioCD een methode toe te voegen die gebruikt kan worden om te testen of de aan deze methode meegegeven uitvoerders de uitvoerders van de audio-CD zijn.

Opgave 3.

- Schrijf een recursieve methode

double som (int n)

 om de eerste n termen van de onderstaande reeks te sommeren:

$$1/2 + 1/3 + 1/4 + 1/5 + 1/6 + 1/7 + 1/8 + 1/9 + \dots$$

- Gegeven is dat de class String een methode

```
public String substring (int beginIndex, int endIndex)
bevat die de substring van beginIndex t/m endIndex-1 van de string
retourneert.
```

bijvoorbeeld: "abcdef".substring(3,6) levert de String "def" op
 "abcdef".substring(1,4) levert de String "bcd" op
 "abcdef".substring(0,3) levert de String "abc" op
 "abcdef".substring(2,2) levert de String "" op

Schrijf nu een recursieve methode

boolean palindroom(String s)

die true retourneert als de string s een palindroom bevat.

Een palindroom is een rij characters die van links naar rechts en van rechts naar links gelezen dezelfde reeks characters oplevert.

Voorbeelden van String-waarden die palindromen bevatten zijn:

```
" "
"a"
"aa"
"pop"
"lepel"
"parterretrap"
```

Er mag geen gebruik gemaakt worden van zelf gemaakte hulpmethodes.

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

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