

Opgave 1.

a) Gegeven zijn de onderstaande classes A en B

```
class A {
    int i;

    A () {
        i = 5;
    }

    void add (int x) {
        // Zorg dat x uit precies 1 cijfer bestaat.
        x = x % 10;

        i = i*10 + x;
    }
}

class B {
    A a;
    String s;

    B (A a) {
        this.a = a;
        s = new String();
    }

    boolean isEenCijfer (char c) {
        return '0' <= c && c <= '9';
    }

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

        for (int i = 0; i < s.length(); i++) {
            if (isEenCijfer(s.charAt(i))) {
                int cijfer = s.charAt(i) - '0';
                a.add(cijfer);
            }
        }
    }
}
```

en een programma waarin de volgende statements/declaraties staan

```
A a = new A();
B b = new B(a);
PrintStream out = new PrintStream(System.out);

void println (B y) { // print, gescheiden door een '-', y.s en y.a.i
    out.printf("%s-%d\n", y.s, y.a.i);
}

void doeIets () {
    println(b);
    b.add("3");
    println(b);
    a.i = 7;
    println(b);
    a.add(1);
    println(b);
}
```

```
a = new A();
println(b);
a.add(8);
println(b);
b.add("Piet wordt 20 jaar.");
println(b);
b.a.add(0);
println(b);
}
```

a) Als na de declaratie

```
A x1 = new A();
```

de opdracht

```
x1.add(123)
```

wordt uitgevoerd, wat dan direct na de opdracht de waarde van x1.i?

Wat wordt er met de expressie s.charAt(i) - '0' (in de methode add() van de class B) uitgerekend?

Is een expressie als 'B' - 1 correct Java. Verklaar je antwoord.

Is een expressie als "B" + 1 correct Java. Verklaar je antwoord.

b) Wat is de uitvoer van de methode doeIets()?

c) Declareer een variabele "matrix" die een twee-dimensionaal array van integers met 66 rijen en 111 kolommen bevat. Doe dit gestructureerd, d.w.z. gebruik indien nodig constanten.

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

Voor deze opgave heeft een twee-dimensionaal array van integers de eigenschap normaal indien er geen negatieve getallen aanwezig zijn en bij geen enkele rij het aantal nullen groter is dan het aantal positieve getallen.

Voorbeelden: array 1 4 7 10 is normaal; array 0 0 0 10 en -1 2 niet

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

d) Gegeven is het programma in de onderstaande class:

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

    Opgaveld() {
        out = new PrintStream(System.out);
        p = -7;
        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 / 3;
    p = -p;
    print(p, q);
}

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

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

public static void main(String argv[]) {
    new Opgave1().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 planten.

```

class Plant {
    String stam, klasse, clade, orde, familie, geslacht, soort;
    boolean windverspreiding;
    int registratienummer;
    double hoogte;
    // De constructors en methodes doen er voor deze opgave niet toe.
}

```

Maak een class BotanischeTuin zodanig dat in een instantie van deze class kan worden bijgehouden welke planten in de botanische tuin staan. (maximaal 10000). Verder moet de class BotanischeTuin een default constructor en een methode voegToe() bevatten. De default constructor moet een BotanischeTuin-object initialiseren op een lege BotanischeTuin. De methode voegToe() moet de mogelijkheid bieden om 1 Plant-object toe te voegen.

- b) Voorzie de class BotanischeTuin van een methode

```

    BotanischeTuin familie (String f)

```

welke in een nieuw BotanischeTuin-object de planten uit de botanische tuin retourneert welke van de familie f zijn.

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

- c) Gegeven is dat de class BotanischeTuin een methode

```

    BotanischeTuin windverspreiding ()

```

bevat welke in een nieuw BotanischeTuin-object de planten retourneert waarvan de zaden via de wind verspreid worden.

Voeg nu aan de class BotanischeTuin een methode

```

    BotanischeTuin selecteerWindverspreiding (String f)

```

toe, die, gebruik makend van de methodes familie() en windverspreiding() uit de class BotanischeTuin, in een nieuw BotanischeTuin-object de planten uit de botanische tuin retourneert die van de familie f zijn en waarvan de zaden via de wind verspreid worden.

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

Opgave 3.

- a) Gegeven is de iteratieve methode

```

int som (int[] rij) {
    // Initialiseer resultaat op de meest negatieve integer
    // zodat de oplossing ook werkt voor arrays met 0 elementen.
    int resultaat = Integer.MIN_VALUE;

    for (int i = 0; i < rij.length; i++) {
        resultaat += rij[i];
    }

    return resultaat;
}

```

Programmeer deze methode nog een keer maar nu met een recursieve oplossing. Let op, de parameterlijst van de recursieve variant hoeft niet identiek te zijn aan die van de iteratieve oplossing.

- b) Programmeer een recursieve methode die voor een willekeurig getal n (n >= 1) kan bepalen hoeveel gehele delers >= 1 dit getal heeft.

Laat zien hoe je methode moet worden aangeroepen voor het getal 10.

Waardering

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

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