

Opgave 1.

- a) Gegeven zijn de onderstaande classes

```
class Y1 {
    String s;

    Y1 (String s) {
        this.s = s;
    }
}

class Y2 {
    Y1 y1;
    int i;

    Y2 (String s, int i) {
        y1 = new Y1(s); this.i = i;
    }

    void put (String s) {
        y1.s = s; i = s.length();
    }
}
```

en een programma dat de classes Y1 en Y2 gebruikt waarin de volgende declaraties/statements staan

```
static final int MAX_AANTAL = 7;
String a = "a";
Y1[] rij = new Y1[MAX_AANTAL];
for (int i = 0; i < MAX_AANTAL; i++) {
    rij[i] = new Y1(a);
    a += "a";
}
Y2 y2 = new Y2(rij[4].s, 5);

void print (Y2 p1, Y1 p2) {
    out.printf("%d, %s\n%s\n", p1.i, p1.y1.s, p2.s);
}

void m1 (Y1 y1, String s) {
    y1 = new Y1(s);
}

void m2 (Y2 y2, String s, int i) {
    y2.y1 = new Y1(s);
    y2.i = i;
}
```

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

```
void doeIets () {
    m1(rij[4], "c");
    print(y2, rij[4]);
    rij[4] = y2.y1;
    print(y2, rij[4]);
    y2.put("abc");
    print(y2, rij[4]);
    m2(y2, "d", 12);
    print(y2, rij[4]);
    y2 = new Y2(rij[4].s, 14);
    print(y2, rij[4]);
}
```

- b) Gegeven is de onderstaande heading van een methode cosinus()

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

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

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

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

Het aantal uit te rekenen termen is gelijk aan de waarde van de 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 tweedimensionaal array van integers met 8 rijen en 8 kolommen bevat. Doe dit gestructureerd, d.w.z. gebruik indien nodig constanten.

Schrijf een methode "palindroom" die voor een willekeurige matrix van char's kan controleren of alle rijen in deze matrix palindromen van characters zijn.

Een rij van characters is een palindroom als zowel van links naar recht gelezen als van rechts naar links gelezen dezelfde string van characters ontstaat. Zo is de rij characters "a b c d c b a" een palindroom en de rij "a b c d e f" niet.

Doe dit gestructureerd, programmeer eventuele deelproblemen in een aparte methode.

- d) Gegeven is het programma in de onderstaande class:

```
import java.io.PrintStream;
class Opgave1d {
    PrintStream out;
    int p = 5;
    q = 10;

    Opgave1d() {
        out = new PrintStream(System.out);
    }

    void methode1(int p) {
        this.p = p + 2;
        q = 4 * p;
        out.printf("%d\n%d\n", p, q);
    }

    int methode2 (int r, int p) {
        this.q = r - 3;
        methode1(q);
        p = q / this.p;
        out.printf("%d\n%d\n", q, p);
        return r;
    }

    void start() {
        methode1(q);
        q = methode2(p, q);
        out.printf("%d\n%d\n", p, q);
    }

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

Wat drukt dit programma af?

Opgave 2.

- a) Gegeven zijn de onderstaande classes welke gebruikt kunnen worden voor het opslaan van gegevens van boeken.

```
class Boek {
    String titel,
        genre,
        auteur;
    double prijs;
    int registratienummer; // uniek nummer voor ieder boek

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

class Boekhandel {

    static final int MAX_AANTAL_BOEKEN = 11250;

    Boek[] boekArray;
    int aantalBoeken;

    ...
}
```

Schrijf een default constructor en een methode voegToe() voor de class Boekhandel. De default constructor moet het Boekhandel-object initialiseren op een lege boekhandel met 0 boeken waaraan maximaal 11250 boeken kunnen worden toegevoegd. De methode voegToe moet de mogelijkheid bieden om 1 Boek-object aan een Boekhandel-object toe te voegen.

- b) Voorzie de class Boekhandel van een methode

Boekhandel goedkopeBoeken ()

welke in een nieuw Boekhandel-object de goedkope boeken retourneert. Voor deze opgave is gedefinieerd dat een boek goedkoop is als het minder dan euro 17,50 kost.

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

- c) Gegeven is dat de class Boekhandel een methode

Boekhandel genre (String s)

bevat. Deze methode retourneert, in een nieuw Boekhandel-object, alle boeken die als genre de meegegeven string hebben.

Programmeer, gebruik makend van de methodes goedkopeBoeken() en genre() en zonder gebruik te maken van een herhalingsopdracht, een methode

Boekhandel populaireBoeken ()

die, in een nieuw Boekhandel-object, alle populaire boeken retourneert. Voor deze opgave is gedefinieerd dat een boek populair is als het een goedkoop boek van het genre "Detective" is.

- d) Schrijf voor de class Boekhandel een methode

void verwijder (int registratienummer)

waarmee een verkocht boek uit het Boekhandel-object verwijderd kan worden.

Opgave 3.

- a) Schrijf een recursieve methode minimum() die voor een willekeurig array van int's het minimum kan bepalen.

Houd er rekening mee dat een array 0 elementen kan bevatten.

Bedenk zelf welke parameters de methode minimum() nodig heeft.

Er mag worden aangenomen dat het array volledig gevuld is.

HINT1: Het minimum van 0 getallen is oneindig, oftewel, in Java, Integer.MAX_VALUE

HINT2: Het minimum van twee integers kan worden bepaald met de methode min() uit de class Math.

Zo geldt: "Math.min(-5, 16) == -5" en "Math.min(10, 34) == 10".

- b) 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	totaal
1.	4	4	4	4	16
2.	2	4	2	4	12
3.	4	4			8
					-- +
					36

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