

Opgave 1.

- a) Gegeven zijn de onderstaande classes

```
class Naam {
    String s;

    Naam (String naam) {
        s = naam;
    }

    void citeer () {
        s = "\"" + s + "\"";
    }
}

class Auteur {
    Naam naam;

    Auteur (String s) {
        naam = new Naam(s);
    }

    void putNaam (Naam naam) {
        this.naam = naam;
    }
}

en een programma dat de classes Naam en Auteur gebruikt waarin de
volgende declaraties staan

PrintStream out = new PrintStream(System.out);
Naam naam = new Naam("abc");
Auteur auteur = new Auteur("def");

void m1 (Naam par1, String par2) {
    par1.s = par2;
    par2 = "ghi";
}

void m2 (Auteur par1, Naam par2) {
    par1.putNaam(par2);
    par2.citeer();
}

void print () {
    out.printf("%s %s\n", naam.s, auteur.naam.s);
}

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

void doeIets () {
    naam.citeer();
    print();
    m1(naam, "jkl");
    print();
    m2(auteur, naam);
    print();
    m1(naam, "mno");
    print();
    m2(auteur, naam);
    print();
}
```

- b) Schrijf voor $n \geq 1$ de methode : `double sommeerTermen (int x, int n)` die de sommatie uitrekent van de eerste n termen van de onderstaande in wiskunde genoteerde berekening
- $$2x^2 - 3/4x^4 + 4/7x^6 - 5/11x^8 + 6/14x^{10} - \dots$$
- Hierbij staat de notatie a^b voor a tot de macht b .
 Het is niet toegestaan om methodes uit de class `Math` te gebruiken.

- c) Declareer een variabele "matrix" die een twee-dimensiaal array van integers met 23 rijen en 32 kolommen bevat.
 Doe dit gestructureerd, d.w.z. gebruik indien nodig constanten.

Schrijf een methode `isNegatief()` die voor een willekeurig 2-dimensiaal array van integers aangeeft of dit array de eigenschap negatief heeft. Voor deze opgave is gedefinieerd dat een array de eigenschap negatief heeft indien het array meer negatieve dan positieve getallen bevat.

Het array `-1 0 0 0 0` heeft niet de eigenschap negatief omdat er
`0 0 0 0 0` precies evenveel negatieve als positieve
`0 0 0 0 0` getallen in het array staan (van beide 1).
`0 0 0 0 1`

Het array `-2 -3 4 5` is wel negatief omdat er 3 positieve en 6 negatieve
`-1 0 0 0` getallen in staan. -1 0
`-4 -8 -9 9`

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

```
class Opgaveld {
    PrintStream out;
    int a, b;

    Opgaveld() {
        out = new PrintStream(System.out);
        a = 4;
        b = -3;
    }

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

    int m1(int b) {
        a = 2 * b;
        this.b += 1;
        print(a, b);
        return a;
    }

    int m2 (int c, int a) {
        a = b / 3;
        int b = c + 2;
        c = m1(b);
        print(a, b);
        return c;
    }

    void start() {
        b = m1(a);
        a = m2(b, a);
        print(a, b);
    }

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

Geef de uitvoer van dit programma.

Opgave 2.

- a) Gegeven is de onderstaande class welke gebruikt kan worden voor het opslaan van reclameteksten van maximaal 1250 characters

```
class Reclametekst {
    static final int MAX_AANTAL_CHARACTERS = 1250;

    String tekst;
    int aantalCharacters;

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

Voeg aan de class Reclametekst de methode

```
int aantal (String woord)
```

toe die retourneert hoe vaak het meegegeven woord voorkomt in de reclametekst. Om het probleem wat simpeler te maken mag er vanuit gegaan worden dat een reclametekst alleen woorden en spaties bevat.

HINT: Maak een Scanner-object waarmee de woorden uit het bericht gelezen kunnen worden.

- b) Maak een class Tijdschrift waarmee een object gedefinieerd wordt waarin maximaal 75 Reclametekst-objecten kunnen worden opgeslagen.

Voorzie de class Tijdschrift van een methode voegToe() om een Reclametekst-object toe te kunnen voegen aan een Tijdschrift-object.

- c) Voeg aan de class Tijdschrift een methode milieubewust() toe die, in een nieuw Tijdschrift object, alle milieubewuste reclameteksten retourneert. Voor deze opgave is gedefinieerd dat een reclametekst milieubewust is als het woord "milieu" meer dan 3 keer in de tekst staat.

Pak dit gestructureerd aan. Voeg indien er behoefte is aan methodes die iets doen met een Reclametekst-object deze methodes toe aan de class Reclametekst.

Opgave 3.

- a) Schrijf een recursieve body voor de onderstaande methode welke

```
void print2 (int n) {
    // Hier komt de recursieve body
}
```

voor $n \geq 0$ de getallen 0 t/m n twee keer achtereenvolgens op 1 regel afdrukt. De eerste keer dalend en de tweede keer stijgend. Als n negatief is, is er geen uitvoer.

Bij de aanroep `print2(6)` moet de uitvoer zijn

```
6 5 4 3 2 1 0 0 1 2 3 4 5 6
```

- b) Schrijf een recursieve body voor de onderstaande methode welke

```
int aantal (String s, int n, char c) {
    // Hier komt de recursieve body
}
```

telt hoe vaak een meegegeven character c aanwezig is tussen de eerste n characters van een eveneens meegegeven String s .

voorbeeld: `aantal("abcabcabc", 5, 'b') == 2`

- c) Gegevens is de onderstaande methode uit de class String

```
String substring (int beginindex, int eindindex)
```

welke de substring van de gehele String retourneert vanaf de gegeven beginindex tot de gegeven eindindex, m.a.w. van beginindex t/m eindindex-1

Voorbeelden:

```
"12345".substring(0, 3).equals("123")
"12345".substring(2, 3).equals("3")
"12345".substring(1, 4).equals("234")
"12345".substring(0, 5).equals("12345")
"12345".substring(4, 4).equals("")
```

Schrijf nu een recursieve oplossing voor de onderstaande methode `binnensteVooraan()`

```
String binnensteVooraan (String s)
```

die van de meegegeven String s een nieuwe String maakt bestaande uit het middelste character van s (bij een oneven aantal characters) of de twee middelste characters (bij een even aantal characters), gevolgd door de characters die daar voor en achter stonden, gevolgd door de characters die daar weer voor en achter stonden, enz.

Voorbeelden:

```
binnensteVooraan("").equals("")
binnensteVooraan("c").equals("c")
binnensteVooraan("ce").equals("ce")
binnensteVooraan("AdB").equals("dAB")
binnensteVooraan("PceQ").equals("cePQ")
binnensteVooraan("Pc@eQ").equals("@cePQ")
binnensteVooraan("+Pc@eQ+").equals("@cePQ++")
```

Bij het schrijven van de methode `binnensteVooraan()` mag gebruik gemaakt worden van de methode `substring()` uit de class String.

Waardering

Opgave	a	b	c	d	totaal
1.	4	4	4	4	16
2.	5	2	5		12
3.	2	2	4		8
				-- +	36

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