

Opgave 1.

- a) Gegeven zijn de onderstaande classes

```
class Naam {
    String s;

    Naam (String naam) {
        s = '(' + naam + ')';
    }

    void maakLeeg () {
        s = "(LEEG)";
    }
}
```

```
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("XXX");
Auteur auteur = new Auteur("ZZZ");
```

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

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

```
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.maakLeeg();
    print();
    m1(naam, "VVV");
    print();
    m2(auteur, naam);
    print();
    m1(naam, "FFF");
    print();
    m2(auteur, naam);
    print();
}
```

- b) Schrijf een methode

```
int grootsteZevenvoud (int n)
```

die het grootste zevenvoud uitrekent dat niet groter is dan n. Gegeven is dat $n \geq 0$. Een zevenvoud is een getal dat deelbaar is door 7. Maak geen gebruik van de operatie % (modulo) of van een van de methodes uit de class Math.

- c) Declareer een variabele "matrix" die een twee-dimensiaal array van integers met 11 rijen en 9 kolommen bevat. Doe dit gestructureerd, d.w.z. gebruik indien nodig constanten.

Schrijf een methode aantalToppen() die voor een willekeurig twee-dimensiaal arrays van integers kan tellen hoe vaak in dit array een "top" voorkomt.

Voor deze opgave is een "top" gedefinieerd als een getal op een regel dat zowel groter is als het voorgaande getal en alsook groter is als het volgende getal op die regel.

voorbeeld:

```
in de matrix 1 2 3 2 1 zitten 3 "toppen", namelijk de 3 op de 1e regel
              5 7 5 7 5           en beide 7's op de 2e regel.
              5 4 3 2 1
              1 2 3 4 5
```

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

```
class Opgave1d {
    PrintStream out;
    int a, b;

    Opgave1d() {
        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) {
    int a = 2 * b;
    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 Opgave1d().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 1000 characters

```
class Reclametekst {
    static final int MAX_AANTAL_CHARACTERS = 1000;

    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 90 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 agressief() toe die, in een nieuw Tijdschrift object, alle agressieve reclameteksten retourneert. Voor deze opgave is gedefinieerd dat een reclametekst agressief is als het woord "moet" meer dan 2 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) Gegeven is de onderstaande methode

```
void print (int n) {
    if (n <= 10) {
        out.printf("%d", n);
    }

    out.printf("%d", n%10);
    print(n/10);
}
```

Wat wordt er afgedrukt bij de onderstaande aanroep van deze methode?

```
print(89461)
```

- b) 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
```

- c) 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`

- d) Uit de wiskunde is de driehoek van Pascal bekend. Hieronder staan de eerste zes rijen van deze driehoek.

```

              1
            1 1
          1 2 1
        1 3 3 1
      1 4 6 4 1
    1 5 10 10 5 1
```

Voor deze opgave zullen we deze rijen beginnen te nummeren met 1, zodat van het afgedrukte gedeelte van de driehoek de bovenste rij (waar alleen de '1' in staat) de rij met nummer 1 is en de onderste rij de rij met nummer 6 is

Voor een willekeurige rij met nummer i geldt dat deze rij i elementen bevat. Als we `pascal(i,j)` definieren als het j -de getal van de i -de rij, dan geldt dat dit getal gelijk is aan 1 als j het eerste of het laatste getal van de i -de rij aanwijst. Voor ieder getal dat tussen het eerste en laatste getal van een rij inligt, geldt dat dit getal verkregen wordt door uit de voorgaande rij het $(j-1)$ -ste en j -de getal te sommeren.

Schrijf een recursieve methode `pascal` om het j -de getal van de i -de rij te bepalen. De heading van de methode `pascal` moet er als volgt uitzien:

```
int pascal (int i, int j)
```

Bij het schrijven van de methode mag er vanuit gegaan worden dat de aanroep correct zal zijn, m.a.w. aanroepen als `pascal(1, 10)` of met waarden van het argument ≤ 0 , zullen niet voorkomen.

Waardering

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

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