

a) Als de onderstaande class

```
class A {
    String s;
    int i;

    A () {
        s = new String();
        i = 0;
    }

    A (String string) {
        s = string;
        i = string.length();
    }
}

en de declaraties
```

```
Output out = new Output();
A aObject = new A("tekst");
```

```
void m1 (A a) {
    a.s = a.s + "i";
    a.i = a.i + 1;
    out.println(aObject.i);
    out.println(aObject.s);
    m2(a.s);
}
```

```
void m2 (String s) {
    s = s + "?";
}
```

gegeven zijn, wat wordt er dan door de onderstaande statements afgedrukt?

```
out.println(aObject.i);
out.println(aObject.s);
m1(aObject);
out.println(aObject.i);
out.println(aObject.s);
```

b) Gegeven zijn de onderstaande declaraties

```
static final int MAANDAG = 1,
    DINSDAG = 2,
    WOENSDAG = 3,
    DONDERDAG = 4,
    VRIJDAG = 5,
    ZATERDAG = 6,
    ZONDAG = 7;
```

Gebruik een switch-statement om de onderstaande methode te soortDag() implementeren. Doe dit met zo min mogelijk case labels. Er mag bij het implementeren vanuit gegaan worden dat voor de waarde van de formele parameter 'dag' geldt: MAANDAG <= dag <= ZONDAG.

c)

```
String soortDag (int dag)
/* zaterdag en zondag : weekend
vrijdag : bijna weekend
maandag t/m donderdag : werkdag
*/

Schrijf de onderstaande methode submatrix() die uit de matrix m
een vierkante submatrix met n rijen en kolommen retourneert volgens de
gegeven startRij en startKolom.
```

```
int[][] submatrix (int[][] m, int n, int startRij, int startKolom)
```

Voorbeeld: Bij de matrix

```
1 2 3 4 5
6 7 8 9 1
2 3 4 5 6
7 8 9 0 3
```

zou

```
6 7 8
2 3 4
7 8 9
```

de submatrix met 3 rijen en kolommen en als startRij en startKolom respectievelijk 1 en 0, zijn

d) Gegeven is het programma in de onderstaande class:

```
class Opgeve_1d {

    Output out;
    int a = 2,
    b = 3;

    Opgeve_1d() {
        out = new Output();
    }

    void print (int a, int b) {
        out.println(a);
        out.println(b);
    }

    int methode1(int b) {
        b = b * a;
        a = a - b;
        print(a, b);
        return a;
    }

    int methode2 (int b, int c) {
        int a = c;
        b = a - c;
        c = methode1(c);
        print(a, b);
        return b;
    }

    void start() {
        a = methode1(a);
        b = methode2(b, a);
        print(a, b);
    }

    public static void main(String argv[]) {
        new Opgeve_1d().start();
    }
}
```

welke uitvoer geeft dit programma?

Opgave 2.

- a) Maak een class Postzegel voor het opslaan van de volgende gegevens over een postzegel: naam land, waarde postzegel, wel of niet gestempeld.

Voorzie de class Postregel van een constructor met parameters om alle velden van de class Postregel op zinnvolle waarden te initialiseren bij het aannemen van een nieuw Postregel object.

Denk er aan dat er voor iedere class altijd een default constructor gemaakt moet worden die alle variabelen een initiele waarde geeft.

- b) Maak een class Postzegelverzameling welke een object definieert waarin maximaal 10000 Postzegel objecten kunnen worden opgeslagen.

Voorzie de class `PostzegelVerzameling` van een methode `voegToe()` om een `Postzegel` object toe te kunnen voegen aan het `PostzegelVerzameling` object.

- c) Voeg aan de class PostzegelVerzameling de onderstaande methode zoek() toe die, gegeven de naam van een land, in een nieuw PostzegelVerzameling object, alle Postzegels objecten retourneert van het opgegeven land.

Postzegelverzameling zoek (String land)

Pak dit gestructureerd aan door o.a. aan de class Postregel een methode toe te voegen die gebruikt kan worden om te testen of de postregel van het juiste land is.

Opgave 3.

- a) Geef een recursieve implementatie voor het tellen van het aantal cijfers in een getal.

HINT: Je kunt hierbij gebruik maken van het feit dat een getal van n cijfers gesplitst kan worden in het laatste cijfer en het getal met de resterende $n-1$ cijfers. Zo geldt in Java `"123" / 10 == 12` en `"123" % 10 == 3`.

- b) Geef een recursieve implementatie voor de onderstaande methode `binarySearch()` die m.b.v. `binary search` antwoord geeft op de vraag of de `x` voorkomt in de rij op de indexplaatsen van begin `l` m eind.

```
boolean binarySearch (int[] rij, int begin, int eind, int x)
```

Er mag vanuit gegaan worden dat de methode `binarySearch()` wordt aangeroepen met zinvolle waarden voor 'begin' en 'eind'.

BEKNOPTE BESCHRIJVING BINARY SEARCH

Bij binary search wordt de index 'midden' bepaald van het element dat midden tussen de elementen met de indices 'start' en 'eind' in ligt. Vervolgens wordt gekeken of het gezochte getal gelijk is aan het element met de index 'midden'. Zo nee, dan wordt verdergezocht in de deelrij links of recht van het element met index 'midden'.

Opavge	a	b	c	d	total
1.	4	4	4	4	16
2.	2	4	6		12
3.	3	5			8
					-- +
					36

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