



UITWERKINGEN

Opgave 1.

a) a
a
A
A
b
b
A
b
b
A

```
b) double sommeerTermen (int x, int n) {  
    double resultaat = 0.0;  
    int teken = -1,  
        macht = x;  
  
    for (int i = 2; i <= n+1; i++) {  
        teken = -teken;  
        macht *= x; // macht == x^i  
  
        resultaat += teken * 1.0/i * macht  
    }  
  
    return resultaat;  
}
```

```
c) static final int AANTAL_RIJEN    = 6,  
    AANTAL_KOLOMMEN = 4;  
  
int[][] matrix = new int[AANTAL_RIJEN][AANTAL_KOLOMMEN];  
  
int aantalToppen (int[][] m) {  
    int resultaat = 0;  
  
    for (int i = 0; i < m.length; i++) {  
        for (int j = 1; j < m[0].length-1; j++) {  
            if (m[i][j-1] < m[i][j] && m[i][j] > m[i][j+1]) {  
                resultaat += 1;  
            }  
        }  
    }  
  
    return resultaat;  
}
```

d) 9
1
13
3
4
~

Opgave 2.

```
a)  int aantal (String woord) {
    int resultaat = 0;

    for (int i = 0; i < aantalWoorden; i++) {
        if (woordRij[i].equals(woord) {
            resultaat += 1;
        }
    }

    return resultaat;
}

b)  class Bibliotheek {
    static final int MAX_AANTALBOEKEN = 10000;

    Boek[] boekRij;
    int aantalBoeken;

    Bibliotheek () {
        boekRij = new Boek[MAX_AANTAL_BOEKEN];
        aantalBoeken = 0;
    }

    void voegToe (Boek boek) {
        boekRij[aantalBoeken] = boek;
        aantalBoeken += 1;
    }
}

c)  toevoegen aan de class Boek

    static final String SCIENCE_FICTION_WOORD = "ruimteschip";
    static final int    GRENS_SCIENCE_FICTION =    20,
                        GRENS_ROMAN           = 50000;

    boolean isScienceFictionRoman () {
        return aantal(SCIENCE_FICTION_WOORD) > GRENS_SCIENCE_FICTION
&&
        aantalWoorden > GRENS_ROMAN
    }

    ;

    toevoegen aan de class Bibliotheek

    Bibliotheek scienceFiction () {
        Bibliotheek resultaat = new Bibliotheek();

        for (int i = 0; i < aantalBoeken; i++) {
            if (boekRij[i].isScienceFictionRoman()) {
                resultaat.voegToe(boekRij[i]);
            }
        }

        return resultaat;
    }
}
```

Opgave 3.

```
a)  int aantalZevens (int getal) {
      if (getal == 0) {
          return 0;
      }

      return (getal % 10 == 7 ? 1 : 0) + aantalZevens(getal/10);
  }
```

of

```
int aantalZevens (int getal) {
    if (getal < 10) {
        if (getal == 7) {
            return 1;
        } else {
            return 0;
        }
    }

    if (getal % 10 == 7) {
        return 1 + aantalZevens(getal/10);
    } else {
        return aantalZevens(getal/10);
    }
}

b)  int indexMaximum (int[] r, int l) {
      int resultaat = 0,
          maximum = r[0];

      for (int i = 1; i < l; i++) {
          if (r[i] > maximum) {
              resultaat = i;
              maximum = r[i];
          }
      }

      return resultaat;
  }

void verwissel (int[] r, int index1, int index2) {
    int hulp = r[index1];
    r[index1] = r[index2];
    r[index2] = hulp;
}

void sort (int[] rij, int lengte) {
    if (lengte == 1) {
        return;
    }

    verwissel(rij, indexMaximum(rij, lengte), lengte-1);
    sort(rij, lengte-1);
}
```

N.B. de parameter van het type "int[]" mag in alle drie de methoden

worden weggelaten indien in de oplossing wordt aangegeven dat deze drie methoden in een class `IntRij` staan met een globale variabele van dit type.