

HO GENT

H5 ArrayList en arrays - next level -
Werkcollege

Table of Contents

1. Doelstellingen per oefening	1
2. Oefeningen	1
2.1. Vul een ArrayList op en toon ze daarna	1
2.2. Vul een ArrayList op zonder dubbels	2
2.3. Zoek de fout: declaratie en creatie van een eendimensionale array	2
2.4. Herwerk deze code door gebruik te maken van een array	2
2.5. Getallen uit array die groter zijn dan een extra getal	3
2.6. Hoogste en laagste uit een array + index waarop deze te vinden zijn	4
2.7. Enhanced for gebruiken om arrays te doorlopen	4
2.8. Enhanced for gebruiken om arrays te doorlopen - herwerking vorige oefeningen	5
2.9. Eendimensionale array van Rekeningobjecten - doorlopen met enhanced for	5
2.10. Zoek de fout: declaratie en creatie van een tweedimensionale array	6
2.11. Gemiddelde van de elementen van een tweedimensionale array	6
2.12. Tweedimensionale array van objecten aanmaken, initialiseren en tonen	7

1. Doelstellingen per oefening

- Oefening 2.1: Creatie en gebruik van ArrayList + gebruik enhanced for
- Oefening 2.2: Creatie en gebruik van ArrayList + extra methodes / enhanced for
- Oefening 2.3: Declaratie en creatie van een eendimensionale array
- Oefening 2.4: Inzien wanneer een array gebruikt moet worden, creatie, initialisatie en doorlopen van een eendimensionale array
- Oefening 2.5: Creatie, initialisatie en doorlopen van een eendimensionale array
- Oefening 2.6: Doorlopen van een vooraf door opsomming gedefinieerde eendimensionale array
- Oefening 2.7: Gewone for omzetten naar enhanced for bij arrays
- Oefening 2.8: Gewone for omzetten naar enhanced for bij arrays
- Oefening 2.9: Enhanced for bij array met objecten
- Oefening 2.10: Creatie en initialisatie van een tweedimensionale array
- Oefening 2.11: Doorlopen van een tweedimensionale array van een primitief datatype
- Oefening 2.12: Doorlopen van een tweedimensionale array van objecten

2. Oefeningen

2.1. Vul een ArrayList op en toon ze daarna

De gebruiker geeft bedragen in totdat de gebruiker -1 ingeeft. Alle ingegeven bedragen die een positief getal voorstellen of 0 zijn, worden in een ArrayList bewaard (-1 niet inbegrepen). Druk vervolgens de lijst af.

Indien de lijst leeg is dan wordt de boodschap "de lijst is leeg" weergegeven, anders worden alle elementen van de lijst weergegeven.

Voorbeelduitvoer:

```
Geef een positief bedrag, stop met -1: 4,50
Geef een positief bedrag, stop met -1: 2,29
Geef een positief bedrag, stop met -1: -5,17
Geef een positief bedrag, stop met -1: 12
Geef een positief bedrag, stop met -1: -1
De lijst met bedragen:      4,50      2,29      12,00
```

OF

```
Geef een positief bedrag, stop met -1: -1
De lijst is leeg!
```

2.2. Vul een ArrayList op zonder dubbels

De gebruiker geeft een getal in van 8 cijfers. Dit wordt gecontroleerd en opnieuw gevraagd indien nodig. Daarna worden de individuele cijfers uit het getal gehaald en in een lijst gestopt (in de volgorde waarin ze in het getal voorkomen), waarbij wordt gecontroleerd of een cijfer al in de lijst voorkomt. Tenslotte worden de cijfers uit de lijst getoond.

Voorbeelduitvoer:

Geef een getal van 8 cijfers:

123456789

Geef een getal van 8 cijfers:

1234567

Geef een getal van 8 cijfers:

12345678

Alle verschillende cijfers in dit getal (in volgorde van voorkomen) zijn:

1 2 3 4 5 6 7 8

OF

Geef een getal van 8 cijfers:

12121212

Alle verschillende cijfers in dit getal (in volgorde van voorkomen) zijn:

1 2

2.3. Zoek de fout: declaratie en creatie van een eendimensionale array

1. -

```
final int MAX = 3;
int table[MAX] = {1,2,3};
```

2. -

```
int max = 5;
int table1[] = new int [max];
table1[max] = 5;
max++;
double table2[] = new int [max];
```

2.4. Herwerk deze code door gebruik te maken van een array

De opgave was om voor alle maanden van het jaar de gezinsuitgaven in te voeren en het

maandgemiddelde te berekenen. Daarna moest je het nummer afdrukken van die maanden, waarvan de uitgaven groter waren dan het gemiddelde.

Gegeven: de code van de klasse `Gezinsuitgaven.java`

Hierin werd beredeneerd dat pas na de bepaling van het gemiddelde kon nagegaan worden of de uitgaven voor een bepaalde maand groter waren dan het gemiddelde of niet. Vandaar dat er 12 variabelen nodig werden geacht om de maandelijkse gezinsuitgaven bij te houden.

Gevraagd:

Bedenk waarom dit niet zo'n goed idee is (tip: stel je voor dat je dit voor weekuitgaven gaat doen in plaats van maanduitgaven) en herwerk deze oefening en maak gebruik van een array!

De uitvoer blijft vanzelfsprekend dezelfde:

```
Geef de uitgaven in voor maand 1 : 1000
Geef de uitgaven in voor maand 2 : 1000
Geef de uitgaven in voor maand 3 : 1000
Geef de uitgaven in voor maand 4 : 1000
Geef de uitgaven in voor maand 5 : 1000
Geef de uitgaven in voor maand 6 : 1000
Geef de uitgaven in voor maand 7 : 3000
Geef de uitgaven in voor maand 8 : 1000
Geef de uitgaven in voor maand 9 : 2000
Geef de uitgaven in voor maand 10 : 1000
Geef de uitgaven in voor maand 11 : 1000
Geef de uitgaven in voor maand 12 : 2000
In de volgende maanden werd een omzet behaald groter dan het gemiddelde: 7 9 12
```

2.5. Getallen uit array die groter zijn dan een extra getal

Lees 10 gehele getallen in en plaats deze in een array. Lees vervolgens nog een getal `extraGetal` in. Druk alle getallen uit de array af, die strikt groter zijn dan `extraGetal`.

```
Geef getal 1: 48
Geef getal 2: 56
Geef getal 3: 12
Geef getal 4: 15
Geef getal 5: 94
Geef getal 6: 63
Geef getal 7: 57
Geef getal 8: 62
Geef getal 9: 35
Geef getal 10: 81
Geef nog een extra getal: 55
```

Deze getallen zijn strikt groter dan 55:

Index	Waarde
1	56
4	94
5	63
6	57
7	62
9	81

2.6. Hoogste en laagste uit een array + index waarop deze te vinden zijn

Initialiseer een array met de omzet per maand van een bepaald warenhuis. De omzetwaarden zijn respectievelijk: 360, 2100, 450, 1450, 650, 780, 321, 1200, 560, 1850, 960, 420. Druk vervolgens de laagste en de hoogste waarde van de omzet af en het nummer van de corresponderende maand.

```
Hoogste omzet 2100 in maand 2
Laagste omzet 321 in maand 7
```

2.7. Enhanced for gebruiken om arrays te doorlopen

Herschrijf volgende for-statements in **enhanced for**-statements (indien mogelijk):

```
int[] getal = {4,8,2,3,5,17,7,99,3,12};
int kleinste = Integer.MAX_VALUE;
for (int i=0;i<getal.length;i++)
    if (getal[i] < kleinste)
        kleinste = getal[i];
```

```
for (int i=0; i<getal.length;i++)
    getal[i] *= 2;
```

2.8. Enhanced for gebruiken om arrays te doorlopen - herwerking vorige oefeningen

Herwerk de oplossingen van oefening 4, 5 en 6. Ga na welke for-lus je kan vervangen door een enhanced for.

2.9. Eendimensionale array van Rekeningobjecten - doorlopen met enhanced for

1. domeinklasse **Rekening**:

Rekening
<<Property>> -rekeningnummer : long
<<Property>> -saldo : double
<<Property>> -houder : String
+Rekening()
+Rekening(rekeningnummer : long)
+Rekening(rekeningnummer : long, houder : String)
+stortOp(bedrag : double) : boolean
+haalAf(bedrag : double) : boolean

Implementeer de klasse Rekening die je in oefening 1 van hoofdstuk 3 ontworpen hebt. Gebruik de gegeven testklasse (zie Chamilo) om de logica in deze klasse te controleren.

2. applicatieklasse **RekeningApplicatie**:

Maak een eendimensionale array van Rekeningobjecten met volgende saldo's: 100, 777.77, 287.15, -350, 399.99, 123.45, 987.65, -68.18, 413.26. De rekeningnummers zijn opeenvolgende getallen beginnend bij 0 en de houders bestaan telkens uit 1 letter, beginnend bij A.

Doorloop na het aanmaken van de rekeningen de array met een enhanced for om op die manier alle gegevens van de rekeningen in een uitvoerstring te zetten en het gemiddelde saldo te berekenen. Let op: negatieve saldo's zijn volgens het domein niet toegelaten, dus het saldo van die rekeningen zal automatisch op 0 gezet worden!

Deze gegevens (de uitvoerstring en het gemiddelde) worden tenslotte getoond.

De uitvoer ziet er dus uit als volgt:

```
Rekening 0 van A bevat 100,00
Rekening 1 van B bevat 777,77
Rekening 2 van C bevat 287,15
Rekening 3 van D bevat 0,00
Rekening 4 van E bevat 399,99
Rekening 5 van F bevat 123,45
Rekening 6 van G bevat 987,65
Rekening 7 van H bevat 0,00
Rekening 8 van I bevat 413,26
```

Het gemiddelde saldo van deze 9 rekeningen bedraagt: 343,25 euro

2.10. Zoek de fout: declaratie en creatie van een tweedimensionale array

Vind de fouten in volgend stukje code:

```
final int MAX = 10;
int[][] table = new int [MAX][MAX];
for (int i= 0; i<= table.length; i++)
    for (int j= 0; j<= table[j].length; j++)
        table[i,j] = i+ j;
```

2.11. Gemiddelde van de elementen van een tweedimensionale array

Gegeven:

```
public class OefTheorie2Dim
{
    public static void main(String arg[])
    {
        double gem;
        double table[][] = new double [3][2];
        // TO DO
        String uitvoer = String.format("gemiddelde is %.1f", gem);
        System.out.print(uitvoer);
    }
}
```

Gevraagd:

Werk de main-methode verder uit en gebruik enhanced for waar mogelijk:

- de gebruiker geeft alle elementen van de array in.
- bereken het gemiddelde van alle elementen van de array.

Mogelijke uitvoer:

```
Geef getal op rij 1 en kolom 1: 5
Geef getal op rij 1 en kolom 2: -1
Geef getal op rij 2 en kolom 1: 4
Geef getal op rij 2 en kolom 2: 3
Geef getal op rij 3 en kolom 1: -2
Geef getal op rij 3 en kolom 2: 1
gemiddelde is 1,7
```

2.12. Tweedimensionale array van objecten aanmaken, initialiseren en tonen

Creëer de twee dimensionale array “rekening” van type Rekening (zie oef 9).

De array bestaat uit 3 rijen.

Per rij vraag je aan de gebruiker het aantal kolommen.

Per element van de array creëer je een object van Rekening en vul je het saldo op volgens onderstaande voorbeelduitvoer.

```
Geef aantal kolommen in voor rij 1: 2
Geef aantal kolommen in voor rij 2: 4
Geef aantal kolommen in voor rij 3: 3
10,00    20,00
11,00    21,00    31,00    41,00
12,00    22,00    32,00
```