

Oefenopgaven hoofdstuk 4

Oefenopgave 1

Schrijf een class met de naam `Persoon`, dat aan de volgende eisen voldoet: Het programma maakt drie strings (naam, achternaam en woonplaats). Maak de constructoren, getters en setters. Maak een methode met een string `toonNAW`, die de inhoud krijgt van de strings (naam, achternaam en woonplaats). Maak een class `TestPersoon`, daarin wordt de string `naw`, naam, achternaam en woonplaats in de output gezet (met behulp van `toonNAW`).

Strings vergelijken

Breid het programma uit om twee strings met elkaar te vergelijken. U maakt in de testclass drie objecten en vergelijkt de twee namen met elkaar (bijvoorbeeld Piet, Piet en Karel). Zet het resultaat in de output.

Breid het programma uit door in de testClass de twee objecten met elkaar te vergelijken met behulp van `==`. Zet in de objecten dezelfde naam, achternaam en woonplaats.

Verwachtte u deze uitkomst? Zie de tekst net boven paragraaf 4.2!

Wat we gedaan hebben, is een voorbeeld van deze code:

```
String var3 = "code";  
String var4 = "code";  
System.out.println(var1.equals(var2));  
System.out.println(var1 == var2);
```

Wat eronder staat, testen we ook even.

```
String var1 = new String("Piet");  
String var2 = new String("Piet");  
String var3 = new String("Karel");  
gelijk = var1 == var2;  
System.out.println("gelijkv = " + gelijk);  
gelijk = var1 == var3;  
System.out.println("gelijkv = " + gelijk);  
gelijk = var1.equals(var2);  
System.out.println("gelijkv = " + gelijk);  
gelijk = var1.equals(var3);  
System.out.println("gelijkv = " + gelijk);
```

Kopieer deze code in de testClass en u ziet dat dit klopt met het boek!

Karakter of een substring in een string zoeken

Er zijn acht methodes voor het vinden van een char of een substring in een String.

Zoek deze methodes op in de Java-bibliotheek en beschrijf wat ze doen. Geef van alle acht de methodes een voorbeeld in een kort programma. Als string kunt u bijvoorbeeld nemen:

```
String voorbeeld = "Deze mooie taal is vernoemd naar de Java koffie";
```

Oefenopgave 2

Maak een class StringUtil. De class heeft twee methodes:

```
public static String fixSqlFieldValue(String value)
public static String encodeHtmlTag(String tag)
```

De eerste methode verdubbelt de single quotes die in een string voorkomen van een SQL-uitdrukking.

Voorbeeld: CREATE TABLE `waarnemingen` wordt CREATE TABLE ``waarnemingen``

In deze methode wordt een stringBuilder gebruikt, waarvan de lengte = 1,1 x lengteString.

Tevens gebruikt u een forlus:

```
for (int i = 0; i < length; i++) {
    char c = value.charAt(i);
```

De tweede methode vervangt speciale tekens, zodat ze op de juiste wijze in de browser zullen getoond worden. De HTML-tag wordt in een StringBuilder geplaatst en elk van de volgende tekens wordt als volgt vervangen: '<' met "<" '>' met ">" '&' met "&" '"' met """ ' ' met " "

Voorbeeld: <HTML> wordt vervangen door <HTML>

Oefenopgave 3a

CBS

Het Centraal Bureau voor de Statistiek heeft enorm veel gegevens tot zijn beschikking die geanalyseerd kunnen worden. In deze opdracht gaat u dat 'in kleine vorm' nabootsen.

Maak de class **Inwoner** met:

- Twee velden: **leeftijd** (≥ 0 en ≤ 100) en **geslacht** (0 = man, 1 = vrouw)
- Een **constructor** met als parameter leeftijd en geslacht.
- Twee accessors **getLeeftijd** en **getGeslacht**.

Maak de class InwonerGegevens

- Maak een veld met als inhoud een fixed size array met ruimte voor 1000 inwoners.
- Realiseer de constructor zonder parameters die het array initialiseert en met de inwoners vult. De leeftijd van elke inwoner dient random tussen 0 en 100 te worden gevuld.
(int)(100*Math.random())
- Het geslacht dient random 0 of 1 te zijn.
- Realiseer de methodes die de aantallen weergeeft van bepaalde leeftijdsgroepen.
 - a. Jeugd (personen t/m 18 jaar)
 - b. Volwassenen (inwoners vanaf 19 t/m 64 jaar)
 - c. Bejaarde mannen (van 65 jaar en ouder en mannelijk geslacht)

Maak een testclass.

Oefenopgave 3b

Maak een tweedimensionaal array met een m-bij-n-tal (bijvoorbeeld 5-bij-8) getallen. Vul dit array random met getallen tussen 0-100.

Maak vier methodes:

- a. void vulRandom(). Een dergelijk getal krijgt u als volgt: (int)(100*Math.random())
- b. Maak een methode die het grootste getal uit het array teruggeeft.
- c. Maak een methode die het kleinste getal uit het array teruggeeft.
- d. Maak een methode waarbij u een getal meegeeft en dan een string met de indexen terugkrijgt.

Toelichting voor hen die geen wiskunde gehad hebben:

Een tweedimensionaal array is een array met rijen en kolommen. Wanneer $m = 2$ en $n = 3$ en de array is random gevuld met getallen, dan komt de array er als volgt uit te zien:

m - n	0	1	2
0	43	55	7
1	19	77	36

Met behulp van de constructor kunt u dat array aanmaken waarbij u de twee indexen m en n meegeeft.

Bij b krijgt u het getal 77 terug.

Bij c krijgt u het getal 7 terug.

Bij d komt de aanroep in main er als volgt uit te zien: `String s = xx.indexen(55);`

U krijgt dan terug "0 1". Wanneer u een getal zou meegeven die niet in het array voorkomt: bijv. `String s = xx.indexen(56);` dan krijgt u "-1 -1" terug.

Oefenopgave 4

Voeg aan de class `testPersoon` uit Oefenopgave 1 een arraylist van Personen toe.

Voeg aan de arraylist 3 personen toe.

Zet in de arraylist op plaats 2 een ander persoon.

Toon alle personen in de output met een forlus en doe het nogmaals met een `ListIterator`.

Oefenopgave 5

Met insert code "override methods" kunt u aan de class `Persoon` de methode `equals` toevoegen.

In deze methode worden alle instance variabelen met elkaar vergeleken.

Maak deze methode. Zie 4.5.2