



# Programmeren1 Practicum

## Herhaling plus Arrays

Week 3a

We oefenen nog een beetje met de dingen die je vorige week hebt leren kennen: de main-functie, het if- en het while-statement. Daarnaast krijg je een nieuw element: array.

### Eerst even aandacht voor



In een practicum hebben we eerst een gezamenlijk deel. Even opletten wat de docent te vertellen heeft dus. We behandelen eerst:

- main, if en while: zijn er nog vragen?
- Een array, wat is dat nu precies?
- Het samenstellen van een String

### SIMPEL



[1] → We gaan kijken naar arrays. De array `studentNames` is een array van Strings. Door rechte haken, [], achter String zie je dat dit een array van String elementen is.

```
String[] studentNames = {"Rob", "Jan"};
```

Bekijk de code hieronder en vul het resultaat in van de stukjes code.

```
String[] studentNames =  
    {"Peter", "Ger", "Jasper", "Bob", "Marieke"};  
  
studentNames[0] =  
  
studentNames[1] =  
  
studentNames[4] =  
  
studentNames[5] =
```


[2a] → Neem het stuk code hiernaast over in je main-functie. Wat is de uitvoer? Wat is er precies gebeurd?

```
String[] studentNames =  
    {"Peter", "Ger", "Jasper", "Bob", "Marieke"};  
  
System.out.println(studentNames[0]);  
System.out.println(studentNames[1]);  
System.out.println(studentNames[4]);  
System.out.println(studentNames[5]);
```

[2b] → Los het probleem dat optreedt op.

[3] → Nog een oefening met lussen: doe de oefening "Mixed Messages", op bladzijde 23 van het boek.

## BASIS



*Maak de BASIS opdrachten in de les als je al klaar bent met de SIMPEL opdracht, deze helemaal begrijpt of wel toe bent aan iets meer uitdaging.*

**[1] →** Maak een nieuw Java-project in Eclipse.

- Maak een Main class met daarin een main-functie.
- Maak in de main-functie een array van 4 elementen groot. De elementen zijn van het type `int`. Zet hierin de waarden -1, 2, 5 en 14.
- Schrijf hierin gebruikmakend van het **while**- en het **if**-statement een programma dat de output geeft die hiernaast getoond wordt.

```
x = -1
x is smaller than 2
x = 2
x is equal to 2
x = 5
x is larger than 2
x = 14
x is larger than 2
```

**[2] →** Bekijk de code hiernaast. Hierin staan twee geneste while-lussen (een while lus in een andere while lus). Voorspel wat de uitvoer gaat zijn van dit stuk code zonder het eerst in Eclipse te zetten.

**[3] →** Controleer je antwoord bij **[2]** door het naar Eclipse in de `main`-functie te copieren en uit te voeren.

```
int x = 0;
int y = 0;

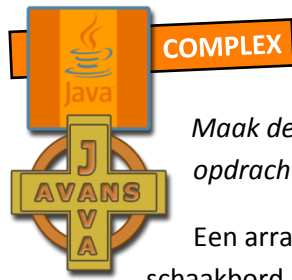
while ( x < 3 ){
    System.out.println( "x = " + x);

    while ( y < 2){
        System.out.println( "y = " + y);
        y++;
    }

    y = 0;
    x++;
}
```

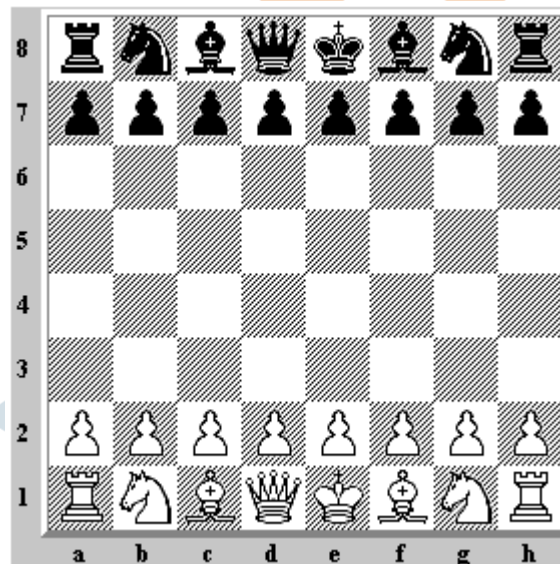
**[4] →** Maak de phrase-o-matic uit je boek door de code van bladzijde 16 in Eclipse in te voeren. Kijk wat het programma doet en bestudeer bladzijde 17. (Waarschijnlijk krijg je deze opdracht niet af in het practicum. Maak het dan thuis af.)

# Java™



*Maak de COMPLEX opdrachten in de les als je al klaar bent met de SIMPEL of BASIS opdracht en die te makkelijk vind of als je nog tijd over hebt.*

Een array is een rij. Een rij is ééndimensionaal. Je zou een speelveld van bijvoorbeeld een schaakbord, kunnen zien als een rij van rijen.



**[1]** → Schrijf een programma dat een speelveld implementeert als een array met arrays. Initialiseer deze matrix met een schaakstand. Gebruik hiervoor de volgende letters:

Witte Koning	- "k"
Witte Koningin	- "q"
Witte Bisschop	- "b"
Wit Paard	- "r"
Witte Toren	- "t"
Witte Pion	- "p"

Zwarte Koning	- "K"
Zwarte Koningin	- "Q"
Zwarte Bisschop	- "B"
Zwart Paard	- "R"
Zwarte Toren	- "T"
Zwarte Pion	- "P"

Neem voor een leeg veld een spatie.

Als het programma gestart wordt print het deze stelling uit. Maak hiervoor gebruik van loops. Of het een wit of zwart veld is waar een stuk op staat mag je buiten beschouwing laten. De beginopstelling wordt bijvoorbeeld afgedrukt als hieronder:

```
TRBQKBRT  
PPPPPPPP
```

```
pppppppp  
trbqkbrt
```

Google

"java initializing arrays"

**[2] →** Mocht je nog tijd hebben, bedenk dan eens of je een functie kan maken waarmee een zet zou kunnen worden gedaan. Deze functie zou er uit kunnen zien als:

```
move(String oldLocation, String newLocation)
```

en gebruikt kunnen worden bijvoorbeeld door een call als:

```
move("a7", "a5");
```

Hiervoor heb je wel String manipulatie functies nodig. Check de String class op de Java API website.

Google

"java manipulating strings", "java api 8 String"

# Java™