

Bioinformatik Tutorium

Arrays und Schleifen

Daniel Bader

Übersicht

- Kurze Wiederholung der Java Einführung
 - Eclipse: Project, Class, main
 - Typen, if-else, Kommentare
- Datentyp: Array
 - eindimensional
 - zweidimensional
- Schleifen
 - for-Schleife
 - while-Schleife

Java mit Eclipse

- Wie geht's los?
 - ein Project erstellen
 - neue Klasse erstellen
 - zum Ausführen: Standard-„main“-Methode schreiben:

```
public static void main(String[] args) {  
}
```

- mehr Infos dazu im Kapitel „Vererbung“

Typen

- **boolean**: **true** oder **false**
- **int**: ganze Zahl ($\pm 2^{31}$, ca. 2 Mrd.)
- **double**: Gleitkommazahl ($\pm 2^{63}$)
- **char**: ein Buchstabe, z.B. 'a'
- **String**: Text beliebiger Länge, z.B. "Daniel hält Tut.\n"

Abfrage von Bedingungen

```
if(condition){
    System.out.print(''It was true. Yeah! \n'');
}else if(condition2){
    //check something else
    System.out.println(''Cond2=true. '');
}else{
    /*
     *when condition and condition2 are false
     */
    System.out.println(''Cond2=true. '');
}
```

1D Arrays

- Speicher für Objekte des gleichen Typs

- Aufbau 1D:

- `array a = ['a', 'b', 'c']`
- Größe wird festgelegt und ist fix, hier 3
- Zugriff auf die einzelnen „Felder“ per Index

felder	[]	[]	[]	[]	[]
indices	0	1	2	3	4

- Ändern von Einträgen:
`array[index] = neuerWert`

2D Arrays

00	01	02
10	11	12
20	21	22

- `array b =`

```
[ [ 'a', 'b', 'c' ]  
  [ 'd', 'e', 'f' ]  
  [ 'g', 'h', 'i' ] ]
```
- a.k.a. „Array of Arrays“
- Indizes:
 - Zeilen → erster Index
 - Spalten → zweiter index

```
31  /*
32  * How to initialize an array?
33  */
34      //empty initialization of 1D array
35      int[] emptyArray = new int[3];
36
37      //empty initialization of 2D array
38      int[][] empty2Darray = new int[3][];
39      empty2Darray = new int[emptyArray.length][emptyArray.length];
40
41      //initialization with defined content of 1D array
42      int[] arrayA = new int[]{1, 2, 3, 4, 5};
43      int[] arrayB = new int[]{6, 7, 8, 9, 10};
44
45      //change of an entry
46      arrayB[2] = 100;
47
48      //initialization of 2D array
49      int[][] sudokuArray = { {1,2,3}, {4,5,6}, {7,8,9} };
50
```

Zusammenfassung: Arrays in Java

- Initialisierung
 - leer
 - mit festgelegtem Inhalt
 - durch ein bereits erstelltes Array
- Zugriff auf Elemente über Indizes

for – Schleife

- Syntax:

```
for(Init; Bedingung; Veränderung){...}
```

- Eigenschaften:

- **Init**: Initialisierung der Schleifenvariablen
- **Veränderung**: Operation, die nach jedem Durchlauf auf der Variable ausgeführt wird
- **Bedingung**: booleschen Ausdruck mit der Variable, wird vor jedem Durchlauf geprüft
 - true: „mach' weiter!“, false: „Schleifenende“
- **Stopkriterium muss** (irgendwann) **erfüllt werden**, sonst Endlosschleife

```
60 //print content of arrayA to console by your own|
61 printer("\n");
62 printer("arrayB: ");
63 for(int index =0; index < arrayB.length; index ++)
64 {
65     printer("[ "+arrayB[index]+" ]");
66 }
67 printer("\n\n");
68
69 //print 2D array to console by your own
70 printer("empty 2D array: \n");
71 for(int verticalIndex=0; verticalIndex < empty2Darray.length; verticalIndex++)
72 {
73     printer("[ ");
74     for(int horiIndex=0; horiIndex < empty2Darray[verticalIndex].length; horiIndex++)
75     {
76         printer("[ "+ empty2Darray[verticalIndex][horiIndex] +" ] ");
77     }
78     printer("]\n");
79 }
80
```

while-Schleife

- Syntax:
`while(Bedingung){...}`
- Eigenschaften
 - **keine** festgelegte Schleifenvariable
 - **Bedingung** boolescher Ausdruck, wird vor jedem Durchlauf geprüft
 - Stop Kriterium!
→ Bedingung **muss** irgendwann „**false**“ werden

```
83 //while loop
84 int indexWhile = 0;
85 int counter = emptyArray.length;
86 while(indexWhile != emptyArray.length)
87 {
88     emptyArray[indexWhile] = counter;
89     counter--;
90     indexWhile++;
91 }
92 printer("\n");
93 printer("new defined array: "+Arrays.toString(emptyArray)+"\n");
```

Das solltet ihr mal gehört haben...

- in Java gibt's Typen
- if-else Abfragen
- Arrays zum Speichern von Elementen
- for-Schleifen
- while-Schleifen

Aufgaben

- Addiere alle Zahlen in einem int-Array und gib sowohl das Array als auch den Mittelwert seiner Einträge auf der Konsole aus.
- Addiere von 2 Arrays, die Einträge, die am gleichen Index stehen (z.B. `arrayA[0] + arrayB[0]`). Der Output soll wie folgt aussehen:
„Die Summe an Position 2 beträgt 5.“
 - a) gleich große Arrays
 - b) unterschiedlich große Arrays
 - c) bei einem geraden Index Summe, bei ungeradem Differenz

Aufgaben 2

- Bilde die beiden Diagonalsummen von einem 2D Array
- Schreibe eine Methode die Zahlen immer an das Ende eines Array einfügt. Dieses soll vergrößert werden, wenn es schon voll ist, aber noch weitere Zahlen eingefügt werden sollen.
- Ändere bei allen Ergebnissen deinen Ansatz:
for ↔ while Schleife