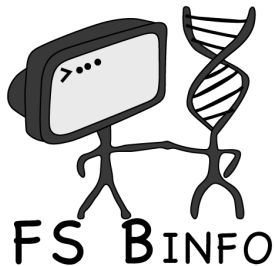


# Umgang mit Arrays, Schleifen und Dateien



Chrissi Seitz

Fachschaft Bioinformatik

17. November 2009

# Zeitplan

---

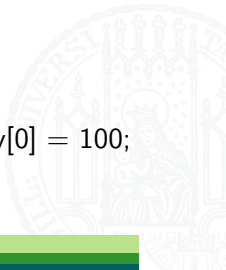
- 27.10 Einführung in Linux und die Shell
  - 03.11 Arbeiten mit Dateien auf der Shell
  - 10.11 Einführung in Java
  - 17.11 **Umgang mit Arrays, Schleifen und Dateien**
  - 24.11 Einführung in objekt-orientierte Programmierung
  - 01.12 Rekursion und Iteration
  - 08.12 GUI Entwicklung
  - 15.12 Debuggen mit Eclipse
- 
- 12.01 Taschenrechner in Java I
  - 19.01 Taschenrechner in Java II
  - 26.01 Wiederholung

Webseite: <http://www.bioinformatik-muenchen.com/bioinfocom/informatik-tutorium>

## Arrays - was ist das?

---

- Array auch genannt Feld, Reihung, Container
- nimmt Objekte gleichen Typs auf und verwaltet diese
- Zugriff auf Objekte über Position im Array  
VORSICHT! Arrays beginnen mit Zählung bei 0! (siehe Beispiel)
- Deklaration mit [ ] und Angabe der Anzahl enthaltener Elemente
- Beispiel: int Array mit 10 Elementen:
  - `int [ ] myArray = new int [10];` *oder*
  - `int myArray [ ] = new int [10];`
  - Zugriff auf erstes Element: `myArray[0]`
  - Zugriff auf letztes somit `myArray[9]`
  - Matrix: `int [ ] [ ] myMatrix = new int [2] [3]`
- Wertezuweisung wie bei normalen Variablen: `myArray[0] = 100;`



## Aufgaben - einfache Arrays

---

- Erstelle ein Array mit 5 beliebigen Zahlen und lass dir diese Ausgeben
- Erstelle ein Array mit 5 beliebigen Strings und lass dir diese Ausgeben
- Erstelle eine Matrix mit den Zahlen 1 - 6 und lass dir diese Ausgeben (mit Position in Klammern; 2 Zeilen, 3 Spalten)  
*Tipp: zur Ausgabe benutze `System.out.println(myArray[Pos]);`*
- Was fällt auf?

## Arrays befüllen mit Hilfe von Schleifen (Loops)

---

- da Befüllen der Arrays von Hand sehr aufwändig, Schleifen zu Hilfe nehmen
- verschiedene Arten von Schleifen:
  - for-Schleife
  - while-Schleife
  - do-while Schleife
- Schleifen (Loops) führen beinhaltenden Code nach gewissen Bedingungen aus
- Verlassen immer auch mit `break`; möglich



## die For-Schleife

---

- Syntax: for (Initialisierung; Schleifenbedingung; Inkrement)  
{ Anweisung; }
- Beispiel (Befüllen unseres Arrays):  

```
for (int i = 0; i < myArray.length; i++){  
    System.out.println(myArray[i]);  
}
```

VORSICHT! myArray.length gibt bei Beispiel von vorher 10 zurück, Positionen gehen aber von 0 bis 9!
- Endlosschleife möglich! `for(int i =0; ; i++){ ... }`

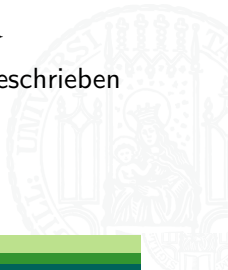
# die While-Schleife

---

- Syntax: `while (Bedingung) { Anweisung; }`
- erst Prüfen der Bedingung, dann evtl. Ausführen der Anweisung
- Beispiel (Befüllen unseres Arrays):

```
int i = 0;
while(i < myArray.length){
    System.out.println(myArray[i]);
    i++;
}
```

- Endlosschleife möglich! z.B.: `while(true){ ... }`
- For-Schleife kann in gleichwertige while-Schleife umgeschrieben werden



## die do-while Schleife

---

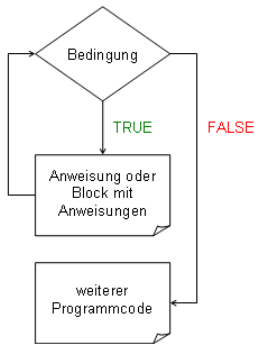
- Syntax: do Anweisung while Bedingung;
- erst Ausführen der Anweisung, dann Prüfen der Bedingung  
VORSICHT! die Anweisung wird mindestens einmal ausgeführt!
- Beispiel (wird die 0 noch mit ausgegeben?):

```
int x =5;
do {
    System.out.println("x= "+ x);
    x--;
} while (x >=1);
```

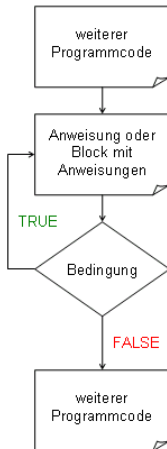


## nochmal: Unterschied while und do-while Schleifen

While-Schleife



Do-While-Schleife



Was passiert bei  
`while (b)`, wenn  
`boolean b = false;`  
?

## Aufgaben - Arrays und Schleifen

---

- Erstelle ein Array mit 80 Zahlen und lass dir diese ausgeben
- Lass dir alle Teiler von 1000 ausgeben  
*Tipp: modulo (%) benutzen*
- Erstelle eine 10x10 Matrix mit den Zahlen 1 - 100 und gib sie mit Positionsangabe aus
- Schreibe ein Programm, welches das kleine 1x1 ausgibt  
*Beispielausgabe: 1x1 = 1; 2x1 = 2; .... ; 3x9 = 27; ...*

## Einfaches File I/O

---

- benötigt `import java.io.*;`
- benötigt `FileReader` / `FileWriter` für Zugriff auf Datei
- zeichen- oder zeilenweise möglich
  - zeichenweise mit Streams
  - zeilenweise mit Buffern (`BufferedReader`, `BufferedWriter`)
- immer Abfangen von Exceptions (sonst gehts nicht!):

```
try{eigentlicher Code}
catch (IOException e){
    throw new RuntimeException ( e )
}
```
- Überprüfen, ob Datei vorhanden mit `file.exists()`;



## Beispiel: File einlesen - zeichenweise

---

```
try{
    File file = new File(filename); //muss existieren!
    FileReader fr = new FileReader(file);
    String gelesen;
    char[] temp = new char[(int) file.length()];
    fr.read(temp);
    gelesen = new String(temp);
    System.out.println(gelesen);
    fr.close();
}
catch (IOException e){
    throw new RuntimeException( e );
}
```



## Beispiel: Erstellen und Schreiben einer Datei

---

- ```
try {
    File datei = new File(filename); //wenn vorhanden,
wirds überschrieben
    BufferedWriter out = new BufferedWriter(new
FileWriter(datei));
    out.write("Inhalt der neuen Datei\n in zwei
Zeilen");
    out.close();
    System.out.println("Dateigroesse = " +
datei.length());
}
catch (IOException e)
    throw new RuntimeException( e ); }
}
```



## Aufgabe: File erstellen und einlesen mit Buffern

---

- nimm das Beispiel von vorher, um ein beliebiges File zu erstellen
- wo wurde die Datei erstellt? Wie sieht der Inhalt aus?
- lies diese Datei zeilenweise ein (siehe Beispiel) und gib den Inhalt aus
- lies diese Datei mit `BufferedReader` ein und gib den Inhalt aus
- *Tipp: zeileweise einlesen mit `String inhalt = in.readLine();`  
`in` ist der Name des neu-erstellten `BufferedReader`*
- *Tipp: mit `while ((content = in.readLine()) != null)` alle Zeilen einlesen (stoppt wenn keine mehr da)  
Nicht vergessen `content` vorher als `String` zu initialisieren!*

# Übungen für Zuhause - Arrays und Schleifen

---

- rausfinden was "Schleifen mit Sprungzielen" sind
- rausfinden was for-each loops sind
- Schreibe ein Programm welches die Primzahlen bis 100 ausgibt
- Realisiere folgende "Wetterstation" Aufgabe: <https://wiki.freitagrunde.org/Javakurs2007/Wetterstation>

## Übungen für Zuhause - File I/O

---

- Weitere File-I/O Dinge (z.B Löschen von Dateien):  
`http://www.java-wi.de/quellen/I0.java`
- Schreibe in ein File "numbersTo1000.txt" die Zahlen 1 - 1000;
- Öffne diese Datei und speichere die Daten in ein Array  
*Tip: um aus einem String ein Integer zu machen benutze*  
`Integer.parseInt("Zahl");`
- Verdopple alle Elemente des Arrays und schreibe diese in ein neues File "numbersTo2000.txt"  
*Tip: um aus einem Integer einen String zu machen benutze*  
`Integer.toString(Zahl);`