LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Lehrstuhl Bioinformatik • Konstantin Pelz

Erste Java-Programme

(Arrays und Schleifen)

Tutorium Bioinformatik
(WS 18/19)

Konstantin: Konstantin.pelz@campus.lmu.de
Homepage: https://bioinformatik-muenchen.com/studium/propaedeutikum-programmierung-in-der-bioinformatik/

Arrays

Ein Array ist eine einfache Liste.

Dieses wird mit einem Datentypen deklariert, d.h. Alle Werte, die in diesem Array gespeichert werden sollen, müssen von demselben Datentyp sein.

Allgemein: `Datentyp[] arrayname = new Datentyp[Länge]`

Speziell für z.B. ein Integer Array der Länge 10:

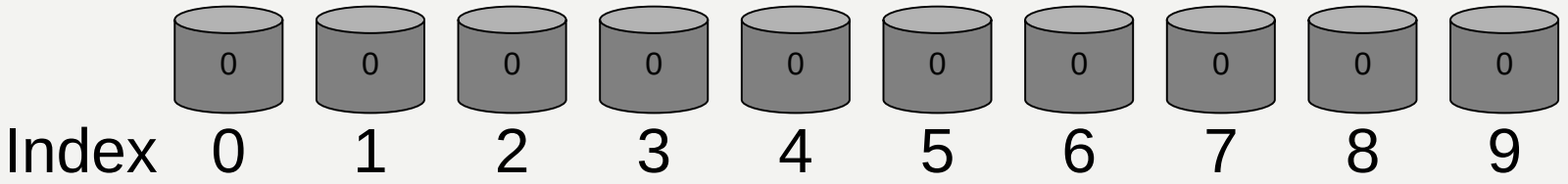
```
1 public class HelloWorld
2 {
3   public static void main(String[] args)
4   {
5
6     int[] firstArray = new int[10];
7
8   }
9 }
```

Arrays

Es erhält eine festgelegte unveränderbare Größe, welche im Attribut *length* gespeichert wird.

```
int[] firstArray = new int[10];
System.out.println(firstArray.length);
```

Schematische Darstellung des *firstArray* :



Wird noch kein spezifischer Wert angegeben, so wird jedes Feld mit dem Standardwert des Datentyps initialisiert.

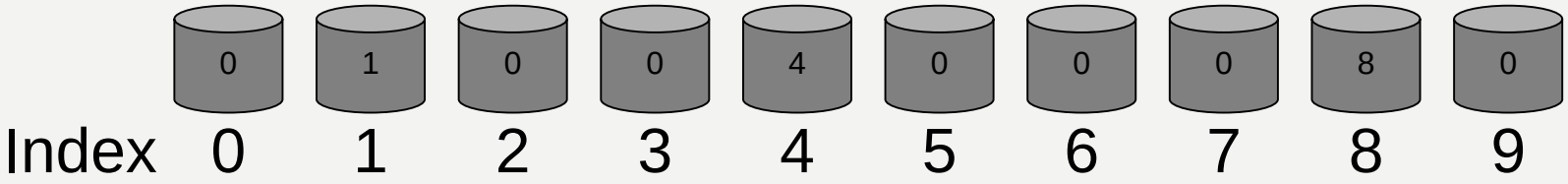
Integer → 0
Boolean → false
String → null

Zugriff auf Arrays

Verändern der Werte in einem Array:

```
int[] firstArray = new int[10];

firstArray[1] = 1;
firstArray[4] = 4;
firstArray[8] = 8;
```



Wichtig: 0 - basiert!

Wird z.B. auf ein 11. Element zugegriffen, `int element = firstArray[10];`
das Array wurde aber mit der Länge 10 initialisiert,
wird eine sog. java.lang.ArrayIndexOutOfBoundsException geworfen.

Zugriff auf Arrays

Verändern der Werte in einem Array:

```
int[] firstArray = new int[10];

firstArray[1] = 1;
firstArray[4] = 4;
firstArray[8] = 8;
```

Wichtig: 0 - basiert!

Wird z.B. auf ein 11. Element zugegriffen, `int element = firstArray[10];`
das Array wurde aber mit der Länge 10 initialisiert,
wird eine sog. java.lang.ArrayIndexOutOfBoundsException geworfen.

Exceptions

Fehler (sogenannte Exceptions) führen in Java direkt zu Programmabstürzen.

Exceptions treten immer erst zur Laufzeit auf, d.h. erst bei der Ausführung des Programms.

Man kann aus einer Exception viele Informationen gewinnen, welche bei der Fehlerbeseitigung helfen können.

Eine Exception liefert in der Regel die Zeilennummer und Klasse, in der der Fehler aufgetreten ist.

Da es verschiedene Arten von Exception gibt, kann man oft an der Art vermuten was der Fehler ist und wie man ihn beheben kann.

Exceptions

ArrayIndexOutOfBoundsException

Zeile im Code, bei der die Exception entstanden ist

Das Array ist offensichtlich kleiner als 11

TestClass.java

```
1
2  public class TestClass {
3
4
5      public static void main(String[] args) {
6
7          String[] firstArray = new String[10];
8
9          System.out.println(firstArray[10]);
10
11     }
12 }
13
```

Debug | Call Hierarchy | Console | Git Staging

```
<terminated> TestClass [Java Application] /usr/lib64/jdk1.8.0_112/bin/java (14.11.2017, 12:08:12)
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 10
        at TestClass.main(TestClass.java:9)
```

Arrays

Wie befüllt man möglichst einfach ein Array?

Aufgabe: Befülle ein Array der Länge 5 mit den Werten von 1 bis 5.

Arrays

Wie befüllt man möglichst einfach ein Array?

Aufgabe: Befülle ein Array der Länge 5 mit den Werten von 1 bis 5.

```
1  public class HelloWorld
2  {
3    public static void main(String[] args)
4    {
5
6      int[] firstArray = new int[5];
7
8      firstArray[0] = 1;
9      firstArray[1] = 2;
10     firstArray[2] = 3;
11     firstArray[3] = 4;
12     firstArray[4] = 5;
13
14   }
15 }
```

Arrays

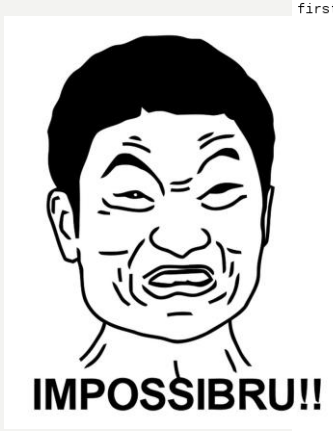
Wie befüllt man möglichst einfach ein Array?

Aufgabe: Befülle ein Array der Länge 5000 mit den Werten von 1 bis 5000.

Arrays

Wie befüllt man möglichst einfach ein Array?

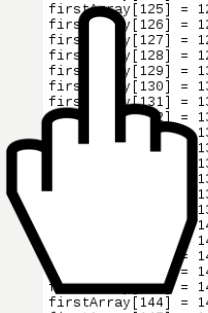
Aufgabe: Befülle ein Array der Länge 5000 mit den Werten von 1 bis 5000.



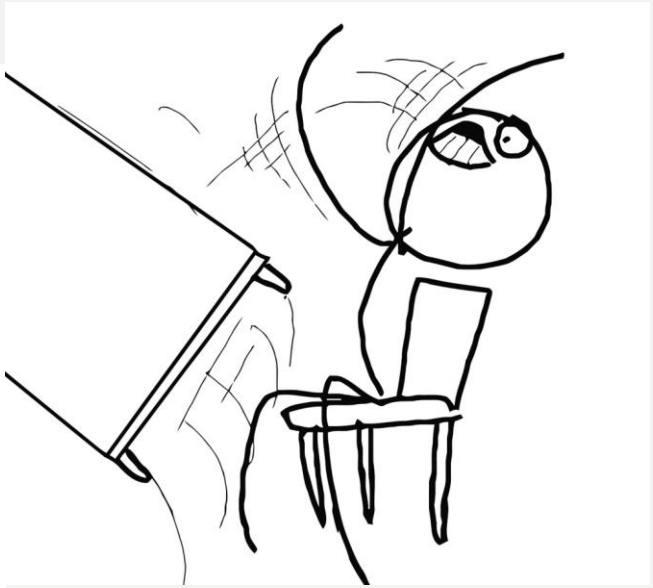

```
firstArray[0] = 1
firstArray[1] = 2
firstArray[2] = 3
firstArray[3] = 4
firstArray[4] = 5
firstArray[5] = 6
firstArray[6] = 7
firstArray[7] = 8
firstArray[8] = 9
      rray[9] = 10
     rray[10] = 11
     rray[11] = 12
     rray[12] = 13
     rray[13] = 14
     rray[14] = 15
     rray[15] = 16
     rray[16] = 17
     rray[17] = 18
     rray[18] = 19
     rray[19] = 20
     rray[20] = 21
     rray[21] = 22
     rray[22] = 23
     rray[23] = 24
     rray[24] = 25
     rray[25] = 26
     rray[26] = 27
     rray[27] = 28
     rray[28] = 29
     rray[29] = 30
     rray[30] = 31
     rray[31] = 32
     rray[32] = 33
     rray[33] = 34
     rray[34] = 35
     rray[35] = 36
firstArray[36] = 37
firstArray[37] = 38
firstArray[38] = 39
firstArray[39] = 40
firstArray[40] = 41
firstArray[41] = 42
firstArray[42] = 43
firstArray[43] = 44
firstArray[44] = 45
firstArray[45] = 46
firstArray[46] = 47
firstArray[47] = 48
firstArray[48] = 49
firstArray[49] = 50
```

```
firstArray[50] = 51
firstArray[51] = 52
firstArray[52] = 53
firstArray[53] = 54
firstArray[54] = 55
firstArray[55] = 56
firstArray[56] = 57
firstArray[57] = 58
firstArray[58] = 59
firstArray[59] = 60
firstArray[60] = 61
firstArray[61] = 62
firstArray[62] = 63
firstArray[63] = 64
firstArray[64] = 65
firstArray[65] = 66
firstArray[66] = 67
firstArray[67] = 68
firstArray[68] = 69
firstArray[69] = 70
firstArray[70] = 71
firstArray[71] = 72
firstArray[72] = 73
firstArray[73] = 74
firstArray[74] = 75
firstArray[75] = 76
firstArray[76] = 77
firstArray[77] = 78
firstArray[78] = 79
firstArray[79] = 80
firstArray[80] = 81
firstArray[81] = 82
firstArray[82] = 83
firstArray[83] = 84
firstArray[84] = 85
firstArray[85] = 86
firstArray[86] = 87
firstArray[87] = 88
firstArray[88] = 89
firstArray[89] = 90
firstArray[90] = 91
firstArray[91] = 92
firstArray[92] = 93
firstArray[93] = 94
firstArray[94] = 95
firstArray[95] = 96
firstArray[96] = 97
firstArray[97] = 98
firstArray[98] = 99
firstArray[99] = 100
```

```
firstArray[100] = 101
firstArray[101] = 102
firstArray[102] = 103
firstArray[103] = 104
firstArray[104] = 105
firstArray[105] = 106
firstArray[106] = 107
firstArray[107] = 108
firstArray[108] = 109
firstArray[109] = 110
firstArray[110] = 111
firstArray[111] = 112
firstArray[112] = 113
firstArray[113] = 114
firstArray[114] = 115
firstArray[115] = 116
firstArray[116] = 117
firstArray[117] = 118
firstArray[118] = 119
firstArray[119] = 120
firstArray[120] = 121
firstArray[121] = 122
firstArray[122] = 123
firstArray[123] = 124
firstArray[124] = 125
firstArray[125] = 126
firstArray[126] = 127
firstArray[127] = 128
firstArray[128] = 129
firstArray[129] = 130
firstArray[130] = 131
firstArray[131] = 132
                = 133
                = 134
                = 135
                = 136
                = 137
                = 138
                = 139
                = 140
                = 141
                = 142
                = 143
                = 144
firstArray[144] = 145
firstArray[145] = 146
firstArray[146] = 147
firstArray[147] = 148
firstArray[148] = 149
firstArray[149] = 150
```



```
firstArray[150] = 151
firstArray[151] = 152
firstArray[152] = 153
firstArray[153] = 154
firstArray[154] = 155
firstArray[155] = 156
firstArray[156] = 157
firstArray[157] = 158
firstArray[158] = 159
firstArray[159] = 160
firstArray[160] = 161
firstArray[161] = 162
firstArray[162] = 163
firstArray[163] = 164
firstArray[164] = 165
firstArray[165] = 166
firstArray[166] = 167
firstArray[167] = 168
firstArray[168] = 169
firstArray[169] = 170
firstArray[170] = 171
firstArray[171] = 172
firstArray[172] = 173
firstArray[173] = 174
firstArray[174] = 175
firstArray[175] = 176
firstArray[176] = 177
firstArray[177] = 178
firstArray[178] = 179
firstArray[179] = 180
firstArray[180] = 181
firstArray[181] = 182
firstArray[182] = 183
firstArray[183] = 184
firstArray[184] = 185
firstArray[185] = 186
firstArray[186] = 187
firstArray[187] = 188
firstArray[188] = 189
firstArray[189] = 190
firstArray[190] = 191
firstArray[191] = 192
firstArray[192] = 193
firstArray[193] = 194
firstArray[194] = 195
firstArray[195] = 196
firstArray[196] = 197
firstArray[197] = 198
firstArray[198] = 199
firstArray[199] = 200
```



Schleifen

for - Schleife:

Allgemein:

```
for ( Initialisierung; Boolscher Ausdruck; Iteration){
  Anweisung;
}
```

Code:

```
for ( int i = 0; i < 10; i++){
  System.out.println(i);
}
```

```
0
1
2
3
4
5
6
7
8
9
```

while - Schleife:

Allgemein:

```
while (Boolscher Ausdruck){
  Anweisung;
}
```

Code:

```
int i = 0;

while (i < 10){
  i++;
  System.out.println(i);
}
```

```
0
1
2
3
4
5
6
7
8
9
```

Schleifen

Was passiert?

```
1  public class HelloWorld
2  {
3    public static void main(String[] args)
4    {
5
6      int i = 0;
7
8      while (i < 10){
9        System.out.println(i);
10     }
11
12   }
13 }
```

Schleifen

Was passiert?

```
1  public class HelloWorld
2  {
3    public static void main(String[] args)
4    {
5
6      int i = 0;
7
8      while (i < 10){
9        System.out.println(i);
10     }
11
12   }
13 }
```

Ist immer wahr → Schleife bricht nie ab

→ Programm terminiert nicht

Schleifen

Wie befüllt man möglichst einfach ein Array?

Aufgabe: Befülle ein Array der Länge 5000 mit den Werten von 1 bis 5000.

```
int[] firstArray = new int[5000];

for(int i = 0; i < firstArray.length; i++){
  firstArray[i] = i+1;
}
```

oder

```
int[] firstArray = new int[5000];

int i = 0;

while (i < firstArray.length){
  firstArray[i] = i+1;
  i++;
}
```

Arrays

Mehrdimensionale Arrays:

```
int[][] secondArray = new int[6][5];
```

Verschachtelte *for* - Schleife:

```
for (int row = 0; row < secondArray.length; row++){
  for (int col = 0; col < secondArray[row].length; col++){
        secondArray[row][col] = row+col;
  }
}
```

0,0	0,1	0,2	0,3	0,4
1,0	1,1	1,2	1,3	1,4
2,0	2,1	2,2	2,3	2,4
3,0	3,1	3,2	3,3	3,4
4,0	4,1	4,2	4,3	4,4
5,0	5,1	5,2	5,3	5,4

Arrays

Mehrdimensionale Arrays:

```
int[][] secondArray = new int[6][5];
```

Verschachtelte *for* - Schleife:

```
for (int row = 0; row < secondArray.length; row++){
  for (int col = 0; col < secondArray[row].length; col++){
        secondArray[row][col] = row+col;
  }
}
```

0,0	0,1	0,2	0,3	0,4
1,0	1,1	1,2	1,3	1,4
2,0	2,1	2,2	2,3	2,4
3,0	3,1	3,2	3,3	3,4
4,0	4,1	4,2	4,3	4,4
5,0	5,1	5,2	5,3	5,4

Arrays

Mehrdimensionale Arrays:

```
int[][] secondArray = new int[6][5];
```

Verschachtelte *for* - Schleife:

```
for (int row = 0; row < secondArray.length; row++){
  for (int col = 0; col < secondArray[row].length; col++){
        secondArray[row][col] = row+col;
  }
}
```

row = 0

0,0	0,1	0,2	0,3	0,4
1,0	1,1	1,2	1,3	1,4
2,0	2,1	2,2	2,3	2,4
3,0	3,1	3,2	3,3	3,4
4,0	4,1	4,2	4,3	4,4
5,0	5,1	5,2	5,3	5,4

Exkurs: Methoden

Wie sieht eine Methode allgemein aus?

```
public static Rückgabewert methodenname(Übergabeparameter){
  // Anweisungen
  return Rückgabewert;
}
```

Wo im Code stehen Methoden?

```
1  public class HelloWorld {
2
3    public static Rückgabewert methodenname(Übergabeparameter){
4
5      return Rückgabewert;
6
7    }
8
9    public static void main(String[] args){
10
11     |
12
13   }
14 }
```

Was sind Rückgabewerte?

- int, double, String, char, ... → Alle Datentypen
- void → Dann gibt die Methode nichts zurück

```
public static void nutzlos(){
  return;
}
```

Exkurs: Methoden

Eine Methode namens *getArray* soll als Übergabeparameter eine Zahl *n* erwarten, ein Array der Größe *n* erzeugen, mit Zahlen von 1 bis *n* füllen und zurückgeben.

```
public static int[] getArray(int n){

  int[] array = new int[n];

  for (int i = 0; i < array.length; i++){
      array[i] = i+1;
  }

  return array;
}
```

Wie benutzt man diese Methode?

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

  int[] array = getArray(10);

}
```

Exkurs: String

```
String s = "Bioinformatik";
int len = s.length();
char a = s.charAt(0);
char b = s.charAt(len-1);
String c = s.substring(2, 5);
System.out.println(a+c+b);
```

Länge des Strings

Character an einer Stelle im String

Substring von / bis
von inklusive
bis exklusive

Selbst überlegen ;)