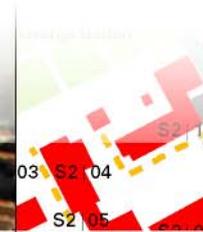


# Anregungen zu Übung 2

Allgemeine Informatik II - SS 2007



Fachgebiet Knowledge Engineering  
Prof. Dr. Johannes Fürnkranz



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT



# Fragen aus den Übungen

Was sind Arrays und wie kann man sie verwenden?

Ein Array ist ein spezieller Datentyp, der mehrere Werte zu einer Einheit zusammenfasst.

- Er ist mit einem Setzkasten vergleichbar, in dem die Plätze durchnummeriert sind.
- Angesprochen werden die Elemente über einen ganzzahligen Index.
- Jeder Platz nimmt immer Werte des gleichen Typs auf.
- Normalerweise liegen die Plätze eines Arrays (seine Elemente) im Speicher hintereinander, doch dies ist ein Implementierungsdetail der virtuellen Maschine, das für Programmierer nicht sichtbar ist.



# Fragen aus den Übungen

Was sind Arrays und wie kann man sie verwenden?

Konkret: Ein Array vom Typ `String` hat mehrere Felder, in denen jeweils ein `String` enthalten ist.

- Arrays erkennt man an eckigen Klammern als Teil des Typnamens.
  - Bei einer strukturellen Deklaration muss die Länge des gewünschten Arrays angegeben werden, in Zeile 2 `String[5]` mit 5 Feldern.
  - Bei expliziter Deklaration mit Inhalten wird ein Array so erstellt, dass für jeden übergebenen Inhalt, hier `String`, ein Feld erstellt wird.

```
1 // Definition eines Arrays mit 5 leeren Feldern
2 String[] stringArray1 = new String[5];
3 // Definition eines Arrays mit 5 Feldern und enthaltenen String
4 String[] stringArray2 = {"Ada", "Beta", "Casa", "Dana", "Emu"};
```



# Fragen aus den Übungen

Was sind Arrays und wie kann man sie verwenden?

Das Array `stringArray2` von der vorherigen Folie kann man sich folgendermaßen vorstellen:

- Die Felder sind von 0 bis Länge-1 nummeriert.
- Auf ein Feld wird über den Bezeichner des Arrays und den Feldindex zugegriffen, z. B. `stringArray2[2]` liefert den String `Casa`.
- Durch Zuweisungen kann der Inhalt der Felder verändert werden, z. B. `stringArray2[2] = "Caesar"`; speicherte den String `Caesar`.

0	1	2	3	4
"Ada"	"Beta"	"Casa"	"Dana"	"Emu"



# Fragen aus den Übungen

Was sind Arrays und wie kann man sie verwenden?

Fortsetzung der vorherigen Folie:

- Der Index eines Arrays ist immer eine Ganzzahl, sprich ein `int`.
- Der Inhalt eines Arrayfelds kann aber etwas ganz anderes sein.
  - `int[] tabelle; String[] nameList; double[] kurse;`

0	1	2	3	4
"Ada"	"Beta"	"Casa"	"Dana"	"Emu"



# Fragen aus den Übungen

Was sind Arrays und wie kann man sie verwenden?

Fortsetzung der vorherigen Folie:

- Aber Vorsicht: Die Länge, also die Anzahl der Felder, kann, nachdem ein Array erstellt wurde, nicht mehr verändert werden.
- Es sollte also schon vorher klar sein, wieviele Felder benötigt werden. Andernfalls bieten sich andere, flexiblere Datentypen an. Dazu später mehr.

0	1	2	3	4
"Ada"	"Beta"	"Casa"	"Dana"	"Emu"



# Fragen aus den Übungen

Wie kann man Arrays verwenden, die nicht bewusst deklariert wurden und über die keine Informationen vorliegen?

Die Anzahl  $n$  der Felder erhält man durch folgende Abfrage:

- Für mein Array `stringArray3` erhalte ich  $n$  als  
`n = stringArray3.length;`
- Die Länge ist immer die Anzahl der Felder. Das letzte Feld ist aber `stringArray3[stringArray3.length-1]`; wobei das erste Feld mit 0 indiziert ist `stringArray3[0]`;

<b>0</b>	<b>1</b>	<b>...</b>	<b>n</b>
<b>"Ada"</b>	<b>"Beta"</b>	<b>...</b>	<b>"nnn"</b>



# Fragen aus den Übungen

Warum wird manchmal `length` und manchmal `length()` verwendet?

Arrays sind Objekte und verfügen immer über die Objektvariable `public final length` in der die Anzahl der Felder hinterlegt ist.

- Über einen Aufruf `arrayBezeichner.length` wird der Inhalt der Variable abgefragt.

Strings sind im Gegensatz zu Arrays veränderlich in der Länge und besitzen keine Konstante, die in Form einer Objektvariablen abgefragt werden kann. Stattdessen gibt es eine Funktion, die die Länge ermittelt und deren Wert zurück gibt.

- Über den Aufruf `stringBezeichner.length()` wird die Länge eines `String` ermittelt.



# Fragen aus den Übungen

## LetterCounter2

Für Aufgaben, deren Lösung sich nicht direkt ergibt, bietet sich ein methodisches Vorgehen an.

Methoden sollten immer in Teile zerlegt und entweder

- von innen nach außen oder
- von außen nach innen implementiert werden.

Das heißt es sollte immer versucht werden, die grundlegende Frage zu finden und diese zu bearbeiten.

- Beispielhaft auf den nächsten Folien ein Vorgehen von innen nach außen.



# Fragen aus den Übungen

## LetterCounter2

### Aufgabe 1 aus Übung 3:

Erstellen Sie eine Klasse `LetterCounter2`. Ihre Klasse soll folgendes enthalten:

- eine Methode `public static void main(String[] args)`
- eine Methode `public static int[] count(String[] text)`
- eine Methode `public static void printCounterArray(int[] letterCounters)`

In der Methode `main` wird die Methode `count` aufgerufen, welche die vorkommenden Buchstaben im `String[] text` zählt und zurückliefert. Dieses Ergebnis wird mittels der Methode `printCounterArray` auf dem Bildschirm ausgegeben.



# Fragen aus den Übungen

## LetterCounter2

Beispielhaft ein Vorgehen von innen nach außen.

Was soll die Methode tun?

- Buchstaben zählen

Wie kann ich das machen?

- `stringBezeichner.charAt(i)` liefert mir den Buchstaben an einer Stelle, `stringBezeichner.length()` die Länge des `String`.
- Mit einer Schleife kann ich jeden Buchstaben des `String` durchlaufen und sein Vorkommen in einem Zählarray speichern.



# Fragen aus den Übungen

## LetterCounter2

Beispielhaft ein Vorgehen von innen nach außen.

Wie komme ich an die `String`?

- Diese sind in einem Array vom Typ `String` gespeichert. Über `stringArray[j]` kann auf den jeweiligen Eintrag zugegriffen werden.
- Über den Aufruf `stringArray.length` erhalte ich die Anzahl der Felder.
- Mit einer Schleife kann ich jeden `String` aus dem Array lesen und ihn weiter verarbeiten.
- Über den Aufruf `stringArray[j].toUpperCase()` wird der Eintrag im Feld `j` in Großbuchstaben umgewandelt.



# Fragen aus den Übungen

## LetterCounter2

Beispielhaft ein Vorgehen von innen nach außen.

Wie können nun aber die Buchstaben gezählt werden?

- Ein Array von Integern mit 26 Feldern repräsentiert mit jedem Feld einen Buchstaben.
  - `integerArray[0]` repräsentiert `'A'`
  - `integerArray[25]` repräsentiert `'Z'`
- Weil für `char` automatisch ein Cast zu `int` durchgeführt wird, kann mit `char` gerechnet werden.
  - `int i = 'A' - 'A'; // Ergebnis in i: 0`
  - `int i = 'Z' - 'A'; // Ergebnis in i: 25`



# Fragen aus den Übungen

## LetterCounter2

Beispielhaft ein Vorgehen von innen nach außen.

Wie können nun aber die Buchstaben gezählt werden?

- Weil für `char` automatisch ein Cast zu `int` durchgeführt wird, kann mit `char` gerechnet werden.
  - `integerArray[c - 'A']++` zählt für `c = 'A'` ein A
  - `integerArray[c - 'A']++` zählt für `c = 'B'` ein B
  - ...
  - `integerArray[c - 'A']++` zählt für `c = 'Z'` ein Z



# Fragen aus den Übungen

## LetterCounter2

Eine optimierte Lösung in 10 Zeilen.

Nur das Array von Integern, was als Rückgabe benötigt wird, wird neu erstellt. Alle übrigen Operationen sind möglich, ohne zusätzliche Variablen zu schaffen.

```
01 public static int[] count (String[] text){
02     int [] intArray = new int[26];
03     for (int i = 0; i < text.length; i++) {
04         text[i] = text[i].toUpperCase();
05         for (int j = 0; j < text[i].length(); j++) {
06             intArray[text[i].charAt(j) - 'A']++;
07         }
08     }
09     return intArray;
10 }
```