

Java - Part 2

Katja Wegner

katja.wegner@eml-r.villa-bosch.de

Überblick

⇒ Referenzdatentypen

- Zeichenketten
- Felder
 - * Eindimensional
 - * Mehrdimensional
- Vektoren

Parameterübergabe in Methoden

call-by-value

Wert/Inhalt der Variable an Methode
übergeben

primitive Datentypen

Beispiel:

int x;

Wert von x

3

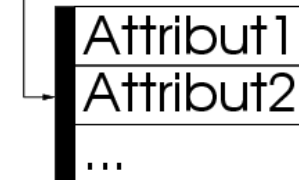
call-by-reference

Referenz/Speicherplatzposition
übergeben

Objekte und Felder (Arrays)

Klasse k;

Referenz



Referenzdatentypen

- ⇒ alle nicht primitiven Datentypen
- ⇒ Initialisierung: **new**
z.B. Klasse bezeichner = **new** Klasse();
- ⇒ Zuweisung von Werten:
 - = (Zuweisungsoperator) - nur Übergabe der Referenz (Inhalt nicht kopiert)
 - **clone()** - erzeugt Kopie des Inhalts des Objekts
- ⇒ Vergleich von Objekten
 - == / != - testet ob Referenz gleich oder ungleich ist (nicht Inhalt)
 - **equals()** - inhaltlicher Vergleich

Zeichenketten

⇒ Zeichenketten (Strings) sind Objekte → Referenzdatentypen

⇒ Zwei Klassen von Strings:

- **String:**
unveränderbar / feste Länge nach Initialisierung
- **StringBuffer:**
können verkürzt / verlängert werden

Die Klasse String

Deklaration:

```
String text, text2;
```

Wertzuweisung:

```
text = 'a';
```

```
text2 = 'Hallo';
```

+ Initialisierung

```
text = new String('Anton');
```

Verbinden

```
text = text + 'Meier';
```

```
text = text2 + text;
```

oder Methode *concat()*

```
text2 = text2.concat(text);
```

Stringlänge

```
int length = text.length();
```

Die Klasse StringBuffer

für umfangreiche Zeichenketten-Operationen

bessere Performance als `string + string2 + string3`

Deklaration:

```
StringBuffer myBuffer;
```

Initialisierung:

```
myBuffer = new StringBuffer();
```

Wertzuweisung:

```
myBuffer.append("Hallo Lisa"); // String oder  
myBuffer.append(2); // primitive Datentypen anhängen  
myBuffer.insert(6, "Mona"); // = Hallo Mona Lisa
```

Umwandlung in String:

```
myBuffer.toString();
```

2.1. Übung

Schreiben Sie ein Programm, das zwei als Kommandozeilenargumente (`args[0]` und `args[1]`) gegebene Strings lexikografisch vergleicht und geben Sie die Relation ($>$, $<$, $=$) zwischen den beiden Strings aus. Benutzen Sie Methoden aus der Klasse `String` (siehe auch Java API der Klasse `String`).

Eindimensionale Felder

0	-1	-2	-3	-4	-5	-6	-7
0	1	2	3	4	5	6	7

Deklaration:

```
Typ [] myArray;
```

Initialisierung:

```
String[] myArray = new String[3]
```

```
double[] myArray2 = new double[10];
```

```
String[] myArray3 = { "alpha", "beta", "gamma" };
```

Wertzuweisung:

```
myArray[0] = "alpha";
```

```
myArray[1] = "beta";
```

```
myArray[2] = "gamma";
```

Länge eines Arrays:

```
int length = myArray.length;
```

2.2. Übung

Erstellen Sie eine Klasse mit dem Namen `TestArray.java`. Definieren Sie in der Methode `main` zwei eindimensionale Arrays mit der Länge zehn, das erste vom Typ `int` und das zweite vom Typ `double`. Weisen Sie dem ersten Array die Werte 2, 4, 6, ..., 20 zu und kopieren Sie diese Werte in das zweite Array.

2.3. Übung

Erstellen Sie eine Klasse mit dem Namen `TestArray2.java`. Definieren Sie in der Methode `main` ein Array vom Typ `double` mit der Länge 100.000. Weisen Sie diesem innerhalb einer Schleife die Werte 0.0, 0.5, 1.0, 1.5, ... zu.

Schreiben Sie eine zweite Schleife um die erste Schleife herum, um diese Wertzuweisung 10.000 mal durchzuführen.

Mehrdimensionale Felder

	0	1	2	3	4
0	v	w	x	y	z
1	a	b	c	d	e

Deklaration: `Typ [][] myMultiArray;`

Initialisierung: `String [][] myMultiArray = new String[3][10]`
`double [][][] myMultiArray2 = new double[10][5][];`
`int [][] myMultiArray3 = new int[5][];`

Wertzuweisung: `myMultiArray3[0] = new int[4];`
`myMultiArray3[2][0] = 9; // [zeile][spalte]`

9				

2.4. Übung

Schreiben Sie eine Klasse `SchuelerNoten`, die ein eindimensionales Feld (1x5) mit allen Namen enthält und ein mehrdimensionales Feld mit den Noten der entsprechenden Schüler (5x10). Vergeben Sie an die Felder Werte und geben Sie den Namen und die Durchschnittsnote aus.

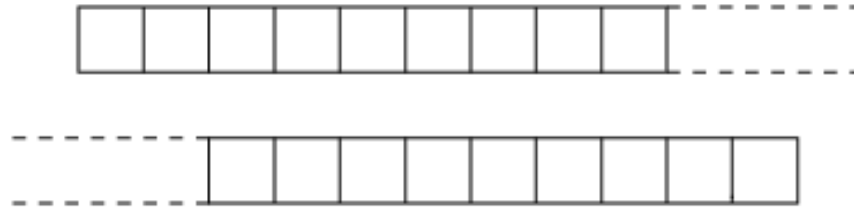
0	Tina Müller
1	Sandra Lehmann
2	Sven Geiger
...	

Namensfeld

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

Notenfeld

Vektoren



⇒ Arrays:

- wenn die Länge einmal gesetzt ist, ist sie fest
- alle Elemente des Arrays müssen zu ein und derselben Klasse gehören

⇒ Vektoren:

- Länge ist variabel und wird automatisch erweitert
- Elemente können zu verschiedenen Klassen gehören

Die Klasse Vector

Deklaration: `Vector myVector;`

Initialisierung: `int iniNum = 10; // Anzahl der Elemente bei Initial.`
`int extNum = 5; // Anzahl der Elemente, um die Vektor`
`// jeweils erweitert werden soll`
`myVector = new Vector(iniNum, extNum);`

Wertzuweisung: `myVector.add(new Integer(3)); // Neues Element anhängen`
`myVector.add(new Double(5.0)); // Wrapperklasse`

`String text = "Alpha";`
`int position = 1;`
`myVector.add(position, text); // An bestimmter Pos. einfügen`

Die Klasse Vector (2)

Elementzugriff:

```
int position = 2;  
  
Double d = (Double)myVector.get(position);  
  
double value = d.doubleValue();
```

Länge:

```
int numElements = myVector.size(); // Anzahl der Elemente  
  
int actualSize = myVector.capacity(); // aktuelle Laenge  
  
myVector.trimToSize(); // setzt die Laenge auf die Anzahl  
                        // der Elemente
```


Die Klasse Vector - Ein Beispiel

```
Vector v = new Vector();
```

```
v.add('123');
```

→



```
int k1 = v.capacity();
```

→

10

```
int k2 = v.size();
```

→

1

```
v.add(0, 'xyz');
```

→



```
v.add(1, 'ABV');
```

→



```
v.trimToSize();
```

→



2.5. Übung

Ändern Sie die Klasse `TestArray2.java` so ab, dass das Array aus `Double` Elementen anstelle von `double` besteht.

Definieren Sie zwei Variablen *long time1* und *time2*.

Fügen Sie die folgenden Anweisungen genau vor und nach der zweiten Schleife ein und geben Sie die Differenz (*time2* - *time1*) aus.

```
time1 = System.currentTimeMillis();
```

```
time2 = System.currentTimeMillis();
```

Schreiben Sie eine neue Klasse die anstelle des Arrays einen Vektor benutzt.

Vergleichen Sie die Zeiten, die von beiden Programmen benötigt werden.