

Hello World!

Javakurs 2013

Arne Kappen

www.freitagrunde.org

03.03.13



Inhalt

1. Organisatorisches
2. Was ist Java?
3. Programmieren in Java
 - Werkzeuge
 - Syntax
 - Fallunterscheidungen
 - Hello World!



Inhalt

1. Organisatorisches
2. Was ist Java?
3. Programmieren in Java
 - Werkzeuge
 - Syntax
 - Fallunterscheidungen
 - Hello World!



Ablauf

	Montag	Dienstag	Mittwoch	Donnerstag	Freitag
10:00	Hello World	Methoden	Übung	API	Vererbung
11:30	Übung	Übung		Übung	Übung
13:15	Pause	Pause	Pause	Pause	Pause
14:15	Schleifen	Übung	Übung	Kapselung	Übung
15:30	Übung			Übung	

Vorlesungen Montag/Dienstag: MA005, Donnerstag/Freitag MA042
Übungen im TEL 106/206

Inhalt

1. Organisatorisches
2. Was ist Java?
3. Programmieren in Java
 - Werkzeuge
 - Syntax
 - Fallunterscheidungen
 - Hello World!



Was ist Java

- 1995 von *SUN Microsystems* entwickelt
- Objekt-orientiert
- Ziel: Plattformunabhängigkeit
 - Deswegen: JVM - Java Virtual Machine
“zwischen” Programm und Betriebssystem

Vor- und Nachteile

- + Kein Nachdenken über Hardware
- + Schneller Einstieg
- + Umfangreiche Libraries
- + Komfortabel
- Wenig tiefgreifende Kontrolle
- Nur bedingt plattformunabhängig
- Relativ unperformant

Inhalt

1. Organisatorisches
2. Was ist Java?
3. Programmieren in Java
 - Werkzeuge
 - Syntax
 - Fallunterscheidungen
 - Hello World!



Syntax

- Programm beginnt mit

```
public class KlassenName {
```

- Darin beginnt die Ausführung bei der
“main-Methode”

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

Syntax

- main-Methode und Programm enden jeweils mit

```
}
```

- Ausdrücke enden mit

```
;
```

- Kommentare durch

```
// Kommentar
```

oder

```
/* Kommentar */
```

Wichtige Datentypen

byte, char	8 bit	0	'c'
short	16 bit	1234	
int, float	32 bit	12345	1.5f
long, double	64 bit	123456L	1.5
String		"Hello World!"	
boolean		true / false	

Variablen

- [Datentyp] name;

Deklaration

name = wert;

Initialisierung

```
int zahl;
```

```
zahl = 42;
```

```
String nachricht = "Hello World!";
```

Fallunterscheidungen

```
if( condition ){  
    // Code, der ausgeführt wird, wenn true  
} else {  
    // Code der ausgeführt wird, wenn false  
}
```

Fallunterscheidungen

- Logische Operatoren

==

Gleichheit

!=

Ungleichheit

||

Oder

&&

Und

!

Nicht

Fallunterscheidungen

- Sonderfall: String

`einString.equals(andererString)`

`!einString.equals(andererString)`

Gleich

Ungleich

4!

Java Code Conventions

- Klassennamen beginnen groß

```
public class HelloWorld { ...
```

- Variablennamen beginnen klein

```
int anzahlDerStudenten;
```

- Camel-Case

```
grossbuchstabeBeiJedemNeuenWort
```


Kompilieren

- Wir können nun korrekten *Quell-Code* schreiben
- Der *Compiler* übersetzt diesen in Anweisungen für die JVM
- Der erzeugte *Byte-Code* wird ausgeführt

“Hello World!”

Beispiel

“Hello World!”

Denkt an das Feedback :)