

1. AuD Tafelübung T-C3

Simon Ruderich

27. Oktober 2010

Über mich

- Simon Ruderich
- Informatikstudent im 2. Semester
- `he29heri@stud.informatik.uni-erlangen.de`
- `http://ruderich.org/simon/uni/aud-2010-2011`

Wer seid ihr?

Wer seid ihr?

Übungsbetrieb

- Tafelübung: Besprechung des aktuellen Stoffes
- Rechnerübung: Programmieren der Übungsaufgaben mit Hilfestellung
- Intensivübung: individuelle Hilfe bei Programmierproblemen; für Programmieranfänger **sehr** zu empfehlen

Aufgaben

- eine Übungsaufgabe pro Woche
- Aufgaben auf der AuD-Seite:
`http://www2.informatik.uni-erlangen.de/teaching/WS2010/AuD/uebungen/index.html`
- **Benutzername:** AUDDFAU **Passwort:** AUDDFAU
- Fragen zu den Übungen:
`aud@i2.informatik.uni-erlangen.de`

Abgabe

- Abgabeschluss Montag 10:00 Uhr
- Abgabe **ausschließlich** über EST:
`https://est.informatik.uni-erlangen.de/`
- spätere Abgaben werden **nicht** akzeptiert
- Server fallen leider auch mal aus!
- ca. 60% der Punkte für den Schein notwendig

Plagiarismus

- Aufgaben müssen alleine bearbeitet werden
- Gruppenarbeiten sind verboten
- Plagiate werden mit 0 Punkten für **alle** Beteiligten bewertet
- `http://www2.informatik.uni-erlangen.de/teaching/plagiarismus.html`

Home-Verzeichnis vor fremdem Zugriff schützen

Wichtig

```
chmod 0700 ~
```

- Erlaubt keinerlei fremden Zugriff.
- Wichtig um Plagiate zu verhindern!

Theorieaufgaben

- Fragen zum aktuellen Stoff
- Abgabe **ausschließlich** als PDF im A4-Format über EST
 - \LaTeX
 - OpenOffice: Datei → Export als PDF (im CIP-Pool installiert)
 - für Windows z. B. FreePDF (<http://freepdfxp.de/>)
 - Scanner im 2. Stock (Raum 02.151)
 - etc.
- bitte nicht als Foto

Programmieraufgaben

- Programmieraufgaben zum aktuellen Stoff
- Abgabe **ausschließlich** über EST
- Referenzsystem ist der CIP-Pool, Lösungen müssen dort laufen
- Zusatzaufgaben werden korrigiert, geben aber keine Punkte

Java I

- **keine** Umlaute oder andere Sonderzeichen, auch nicht in Kommentaren!
- Aufgaben **müssen** mit `javac -encoding ASCII` kompilieren
- EST prüft dies zusätzlich bei Abgabe
- Programm kompiliert nicht ⇒ **0 Punkte!**

Java II

- Schnittstellen der Klassenrumpfe dürfen nicht bearbeitet werden, sonst 0 Punkte!
- neue Methoden sind natürlich erlaubt
- eigene Klassen müssen `public` und in keinem `package` sein
- Fremdbibliotheken und fremder Quellcode sind nicht erlaubt

Wichtige Adressen

- **Vorlesung:** `http://www2.informatik.uni-erlangen.de/teaching/WS2010/AuD/`
- **Übungen:**
`http://www2.informatik.uni-erlangen.de/teaching/WS2010/AuD/uebungen/`
- **Forum:** `https://fsi.informatik.uni-erlangen.de/forum/`
- **IRC:** #fau12k10 im IRCnet

Hallo Welt

HalloWelt.java

```
public class HalloWelt {  
    public static void main(String [] args) {  
        System.out.println("Hallo Welt");  
    }  
}
```

Hallo Welt

Kompilieren und Ausführen

```
$ javac -encoding ASCII HalloWelt.java  
$ java HalloWelt  
Hallo Welt
```

- im Terminal ausführen, nicht in Eclipse
- \$ ist der sogenannte Prompt, nicht mit eingeben!

Java API

- dokumentiert alle verfügbaren Klassen und Methoden
- `http://download.oracle.com/javase/6/docs/api/`

Datentypen

- ganze Zahlen: `int`, `long`
- Fließkommazahlen: `float`, `double`
- Wahrheitswerte: `boolean`
- Nichts: `void` (nur als Rückgabewert)

Beispiele

```
int      a = 5;  
long    b = 42;  
float   c = 5.42 f;    // f!  
double  d = 5.42e12;  // = 5,42 · 1012  
double  e = 47.e-3;   // = 47,0 · 10-3 = 0,047  
boolean f = true;    // false
```

Fallunterscheidung I

if/else

```
if ( Ausdruck ) {  
    // falls Ausdruck wahr  
} else {  
    // anderenfalls  
}
```

Fallunterscheidung II

else if

```
if ( Ausdruck ) {  
    // falls Ausdruck wahr  
} else if ( Ausdruck 2 ) {  
    // falls Ausdruck 2 wahr  
} else if ( Ausdruck 3 ) {  
    // falls Ausdruck 3 wahr  
...  
} else {  
    // sonst  
}
```

Fallunterscheidung III

Beispiel

```
int m = 2;
int y = 2010;
if (m == 1 || m == 3 || m == 5 || m == 7 ||
    m == 8 || m == 10 || m == 12) {
    System.out.println("31 Tage");
} else if (m == 2) {
    if (isLeapYear(y)) {
        System.out.println("29 Tage");
    } else {
        System.out.println("28 Tage");
    }
} else {
    System.out.println("30 Tage");
}
```

for-Schleifen

for

```
for ( Initialisierung ; Bedingung ; Fortsetzung ) {  
    // zu Wiederholender Code  
}
```

Beispiel: $\sum_{n=1}^{10}$

```
int summe = 0;  
for (int i = 1; i <= 10; i++) {  
    summe += i; // summe = summe + i  
}  
// i nicht mehr verfuegbar  
System.out.println(summe);
```

while-Schleifen

while

```
while ( Bedingung ) {  
    // zu Wiederholender Code  
}
```

Beispiel: $\sum_{n=1}^{10}$

```
int summe = 0, i = 1;  
while ( i <= 10 ) {  
    summe += i; // summe = summe + i  
    i++; // i = i + 1  
}  
System.out.println(summe);
```

Fragen

Fragen?