

Praktikum für Programmier-Neulinge

Tag 1: Erste Schritte zum Programmieren

Friedrich-Alexander-Universität
Erlangen-Nürnberg

Informatik 9

Vorbemerkungen

- Algorithmen I
 - **Inhalt:** Datentypen, Datenstrukturen, Kontrollstrukturen, Paradigmen, Algorithmen etc. **am Beispiel Java!**
 - Übungen sind **kein Programmierkurs**
 - Tempo der Vorlesung und Übungen allerdings so, dass man Java semesterbegleitend lernen kann.
- Gewisse Hilfe wird geboten, darüber hinaus ist aber **Eigeninitiative** notwendig
- Das hier ist Starthilfe für alle, die **überhaupt keine** Erfahrung mit Programmieren haben.

Friedrich-Alexander-Universität Erlangen-Nürnberg
Praktikum für Programmieranfänger, WS 2006/07, Folie 1-2

R. Fraedrich

Inhalt

- Tag 1:
 - Datentypen und Variablen
 - Anweisungen
 - Schreiben, Übersetzen und Ausführen von Programmen
 - Kommentare
- Tag 2:
 - Verzweigung (if) und Schleifen (for, while)
- Tag 3:
 - Funktionen und Funktionsaufrufe
 - Zeichenketten (Strings)

Friedrich-Alexander-Universität Erlangen-Nürnberg
Praktikum für Programmieranfänger, WS 2006/07, Folie 1-3

R. Fraedrich

Zielsetzung

- Verstehen und Verfassen von einfachen Programmen wie z.B.:

- Ausrechnen der Potenz x^n :

```
int potenz(int x, int n) {  
    int ergebnis = 1;  
    for (int i = 0; i < n; i++) {  
        ergebnis *= x;  
    }  
    return ergebnis;  
}
```

- Auswerten eines Polynoms $a \cdot x^2 + b \cdot x + c$:

```
int quadPoly(int a, int b, int c, int x) {  
    return (a * potenz(x, 2) + b * x + c);  
}
```

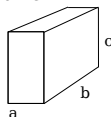
Friedrich-Alexander-Universität Erlangen-Nürnberg
Praktikum für Programmieranfänger, WS 2006/07, Folie 1-4

R. Fraedrich

Beispiel: Volumenberechnung eines Quaders

- Mathematische Lösung aus der Schule:

1. Benennung der Kanten:



2. Werte festlegen:

$$a = 3, \quad b = 6, \quad c = 5$$

3. Lösungsgleichung aufstellen:

$$V = a \cdot b \cdot c$$

4. Werte einsetzen und ausrechnen:

$$V = a \cdot b \cdot c = 3 \cdot 6 \cdot 5 = 90$$

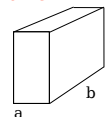
Friedrich-Alexander-Universität Erlangen-Nürnberg
Praktikum für Programmieranfänger, WS 2006/07, Folie 1-5

R. Fraedrich

Beispiel: Volumenberechnung eines Quaders

- Mathematische Lösung aus der Schule:

1. Benennung der Kanten:



2. Werte festlegen:

$$a = 3, \quad b = 6, \quad c = 5$$

3. Lösungsgleichung aufstellen:

$$V = a \cdot b \cdot c$$

4. Werte einsetzen und ausrechnen:

$$V = a \cdot b \cdot c = 3 \cdot 6 \cdot 5 = 90$$

Friedrich-Alexander-Universität Erlangen-Nürnberg
Praktikum für Programmieranfänger, WS 2006/07, Folie 1-6

R. Fraedrich

Variablen und Datentypen

■ Variablen

- Bedeutung
 - Platzhalter für einen Wert, der sich ändern kann
- bestehen aus
 - **Bezeichner**
 - aus Buchstaben, Zahlen und bestimmten Zeichen
 - Konvention: beginnen mit Kleinbuchstaben
 - Achtung! Java ist „case sensitive“
→ a und A sind unterschiedliche Variablen!
 - Beispiele:
 - » a, b1, c_2, temp, meineVariable
 - **Datentyp**

Variablen und Datentypen

■ Warum Datentypen?

- Computer muss wissen
 - wie er Daten speichern soll (als Bits und Bytes)
 - wie er mit Daten umgehen (z.B. Rechnen) soll

■ Beispiel für Datentyp

- „**int**“
 - Ganzzahl (Integer)
 - Speicherbedarf: 4 Bytes
 - Wertebereich: [-2.147.483.648, 2.147.483.647]

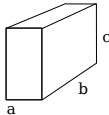
■ Einführung von Variablen (Deklaration)

- Datentyp gefolgt von Bezeichner
- Beispiel: **int a;**
- Variable kann ab ihrer Deklaration verwendet werden

Beispiel: Volumenberechnung eines Quaders

■ Mathematische Lösung aus der Schule:

1. Benennung der Kanten:



2. Werte festlegen:

$$a = 3, \quad b = 6, \quad c = 5$$

3. Lösungsgleichung aufstellen:

$$V = a * b * c$$

4. Werte einsetzen und ausrechnen:

$$V = a * b * c = 3 * 6 * 5 = 90$$

Anweisung

■ Syntax:

- Anweisungen werden in Java durch Semikolon getrennt
- **Anweisung1;**
Anweisung2;

■ Bedeutung:

- Einzelne Vorschrift die bei der Abarbeitung eines Programms auszuführen ist
- Anweisungen werden bei der Ausführung des Programms sequentiell ausgewertet

■ Beispiel:

- Deklaration einer Variablen: **int a;**

■ Konvention:

- maximal eine Anweisung pro Zeile
- Anweisungen unterschiedlicher Schachtelungstiefe (Klassen, Funktionen, Bedingungen, Schleifen) werden unterschiedlich eingerückt (siehe spätere Beispiele)

Zuweisung und Initialisierung

■ Variable einen Wert geben (Zuweisung)

- Schreibweise: **Variable = Wert;**
- Beispiel:
 - **a = 4;**
b = 2;
a = b;
- Bedeutung von „**a = b**“:
 - der Wert von b wird a zugewiesen (danach: a = b = 2)

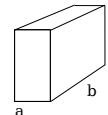
■ Zuweisung eines Startwerts (Initialisierung)

- Variable muss vor erster Verwendung Wert erhalten
- kann (muss aber nicht) bereits bei der Deklaration erfolgen
- Beispiele:
 - **int a = 4;**
int b = a; // korrekt
 - **int a;**
a = 4;
int b = a; // korrekt
 - **int a;**
int b = a; // falsch, da a bislang keinen Wert hat

Beispiel: Volumenberechnung eines Quaders

■ Mathematische Lösung aus der Schule:

1. Benennung der Kanten:



2. Werte festlegen:

$$a = 3, \quad b = 6, \quad c = 5$$

3. Lösungsgleichung aufstellen:

$$V = a * b * c$$

4. Werte einsetzen und ausrechnen:

$$V = a * b * c = 3 * 6 * 5 = 90$$

Multiplikation

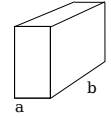
- Syntax
 - `zahl1 * zahl2`
- Bedeutung
 - wie in der Mathematik
- Beispiel:
 - `V = a * b * c;`
- Zusammen mit den Schritten davor:

```
...
int a = 3;
int b = 6;
int c = 5;
int V;

V = a * b * c;
...
```

Beispiel: Volumenberechnung eines Quaders

- Mathematische Lösung aus der Schule:
 1. Benennung der Kanten:



2. Werte festlegen:
 $a = 3, \quad b = 6, \quad c = 5$
3. Lösungsgleichung aufstellen:
 $V = a * b * c$
4. Werte einsetzen und ausrechnen:
 $V = a * b * c = 3 * 6 * 5 = 90$

Beispiel: Volumenberechnung eines Quaders

- Sequentieller Ablauf des Programms:

```
...
int a = 3;
int b = 6;
int c = 5;
int V;

V = a * b * c;
...
```

	a	b	c	V
int a = 3;	3	-	-	-
int b = 6;	3	6	-	-
int c = 5;	3	6	5	-
int V;	3	6	5	?
V = a * b * c;	3	6	5	90

Zwischenfazit

- bis hierhin
 - Datentypen z.B. `int`
 - Variablen
 - Einführung z.B. `int a;`
 - Initialisierung z.B. `int b = 1;`
 - Zuweisung z.B. `a = b;`
 - Operationen z.B. `b = a * a;`

Übersicht: Primitive Datentypen (Auszug)

- Ganzzahlen:

Typ	Größe	Wertebereich
byte	1 Byte	[-128, 127]
short	2 Byte	[-32.768, 32.767]
int	4 Byte	[-2.147.483.648, 2.147.483.647]
long	8 Byte	[-9.223.372.036.854.775.808, 9.223.372.036.854.775.807]

- Anmerkung: Tausender-Trennzeichen nur zur Veranschaulichung

- Fließkommazahlen:

Typ	Größe	Wertebereich
float	4 Byte	einfache Genauigkeit
double	8 Byte	doppelte Genauigkeit

- Anmerkung: Details siehe Vorlesungen

Operatoren

- Arithmetische Operatoren:

Operator	Bedeutung	Beispiel	Ergebnis
+	Addition	7 + 2	9
-	Subtraktion	9 - 4	5
*	Multiplikation	2 * 3	6
/	Division	6 / 2	3
%	Modulo (Rest)	9 % 5	4

- Auswertungsreihenfolge:

- wie in der Mathematik:
 1. *, / und %
 2. + und -
- Klammern:
 - auch wie in der Mathematik
 - „a * (b + c)“ entspricht „a * b + a * c“

Beispiel: Mehrwertsteuer-Berechnung

■ Gegeben: Nettopreise mehrerer Waren

```
double ware_1 = 8.50;
double ware_2 = 12.85;
double ware_3 = 4.90;
double ware_4 = 23.60;
```

■ Gesucht: Rechnungswert inklusive 19% MwSt

```
double rechnung;
```

■ Lösungen:

```
a) ...
double nettosumme = ware_1 + ware_2 + ware_3 + ware_4;
double mwst = nettosumme * 0.19;
rechnung = nettosumme + mwst;
...
b) ...
rechnung = (ware_1 + ware_2 + ware_3 + ware_4) * 1.19;
...
```

So weit, so gut...

■ Wie schreibt man ein Programm?

- Ein Programm besteht aus ein oder mehreren Textdateien
- Zum Programmieren kann man jeden Texteditor verwenden
- Wünschenswert:
 - Syntax-Highlighting
 - Automatische Formatierung
 - etc.
- Beispiele für Texteditoren mit solchen Funktionen
 - Kate, (X)Emacs, Joe, (g)Vim, ...
- Umfangreicher: Integrierte Entwicklungsumgebungen z.B. Eclipse

So weit, so gut...

■ Woraus besteht ein Programm?

- In Java (und anderen objektorientierten Programmiersprachen) besteht ein Programm aus Klassen
- Mehr zu Objektorientierung erfahrt ihr in der Vorlesung
- Jetzt nur so viel Information, dass ihr Programmieren könnt

Bestandteile von Java-Programmen

■ Beispiel Mehrwertsteuerberechnung (MwSt.java):

```
...
double ware_1 = 8.50;
double ware_2 = 12.85;
double ware_3 = 4.90;
double ware_4 = 23.60;

double nettosumme = ware_1 + ware_2 + ware_3 + ware_4;
double mwst = nettosumme * 0.19;
rechnung = nettosumme + mwst;
...
```

Bestandteile von Java-Programmen

■ Beispiel Mehrwertsteuerberechnung (MwSt.java):

```
public class MwSt {

    public static void main(String args[]) {

        double ware_1 = 8.50;
        double ware_2 = 12.85;
        double ware_3 = 4.90;
        double ware_4 = 23.60;

        double nettosumme = ware_1 + ware_2 + ware_3 + ware_4;
        double mwst = nettosumme * 0.19;
        rechnung = nettosumme + mwst;

    }

}
```

Bestandteile von Java-Programmen

■ Beispiel Mehrwertsteuerberechnung (MwSt.java):

```
public class MwSt {
    public static void main(String args[]) {
        double ware_1 = 8.50;
        double ware_2 = 12.85;
        double ware_3 = 4.90;
        double ware_4 = 23.60;

        double nettosumme = ware_1 + ware_2 + ware_3 + ware_4;
        double mwst = nettosumme * 0.19;
        rechnung = nettosumme + mwst;

    }

}
```

Definition einer Klasse mit dem Namen Test

Datei- und Klassenname müssen identisch sein!

Bestandteile von Java-Programmen

- Beispiel Mehrwertsteuerberechnung (MwSt.java):

```
public class MwSt {  
  
    public static void main(String args[]) {  
  
        double ware_1 = 8.50;  
        double ware_2 = 12.85;  
        double ware_3 = 4.90;  
        double ware_4 = 23.60;  
  
        double nettosumme = ware_1 + ware_2 + ware_3 + ware_4;  
        double mwst = nettosumme * 0.19;  
        rechnung = nettosumme + mwst;  
  
    }  
  
}
```

Hier startet die
sequentielle
Auswertung des
Programms

Wie kann man das Programm jetzt ausführen?

- Programm ist für den Computer nur Text
- Ein Computer versteht aber nur eine bestimmte „Maschinensprache“
- Deswegen muss das Programm erst für den Computer übersetzt werden
- Warum dieser „Umweg“?
 - Maschinensprache ist wesentlich schwerer zu verstehen
 - Ein Programm einer Hochsprache kann für Computer mit verschiedenen Maschinensprachen verwendet werden

Wie kann man das Programm jetzt ausführen?

- Wie übersetzt man Java-Programme (z.B. MwSt.java)?
 - indem man den Java-Compiler (javac) mit der Programm-Datei ausführt
 - Ergebnis ist eine neue Datei: MwSt.class
- Wie führt man das Programm aus?
 - indem man den „Interpreter“ (java) mit dem Klassennamen aufruft
- Wie sieht das Ganze in der Realität aus?

```
faiui00a [~/PraProNeu]> javac MwSt.java  
faiui00a [~/PraProNeu]> java MwSt  
faiui00a [~/PraProNeu]>
```

- Was ist passiert?
 - Programm wurde erfolgreich ausgeführt

Ausgabe

- Syntax:
 - `System.out.print(Ausgabe);` // Ausgabe ohne Zeilenumbruch
 - `System.out.println(Ausgabe);` // Ausgabe mit Zeilenumbruch
- Bedeutung:
 - Ausgabe:
 - wird auf der Kommandozeile ausgegeben
 - kann aus Zeichenketten (in „...“) und aus primitiven Datentypen (z.B: Zahlen) bestehen
 - Konkatenation (Verknüpfung) von Ausgabeparametern mit „+“
- Beispiel:
 - Code:

```
int a = 7;  
System.out.print(„Variable a hat den Wert “);  
System.out.println(a);  
System.out.println(„Die Differenz a - 2 = “ + (a - 2) );
```
 - Ausgabe:

```
> java AusgabeProgramm  
Variable a hat den Wert 7  
Die Differenz a - 2 = 5
```

Ausgabe

- Beispiel Mehrwertsteuerberechnung (MwSt.java):

```
public class MwSt {  
  
    public static void main(String args[]) {  
  
        double ware_1 = 8.50;  
        double ware_2 = 12.85;  
        double ware_3 = 4.90;  
        double ware_4 = 23.60;  
  
        double nettosumme = ware_1 + ware_2 + ware_3 + ware_4;  
        double mwst = nettosumme * 0.19;  
        rechnung = nettosumme + mwst;  
  
        System.out.println(„Der Gesamtbetrag ist „ + rechnung);  
  
    }  
  
}
```

Compiler-Fehler

- Wie bei natürlichen Sprachen kann man Fehler machen
 - Rechtschreib-/Tipp-Fehler (ein Wort/Zeichen betreffend)
 - Grammatik-Fehler (die Struktur/Regeln betreffend)
- Compiler (Übersetzer) weist auf solche Fehler hin
 - Beispiel:

```
Exception in thread "main" java.lang.Error: Unresolved  
compilation problem:  
Syntax error, insert ";" to complete Statement  
at Test.main(Test.java:11)
```
- Was tun?
 - a) Übungsleiter rufen?
 - b) Warten bis Übungsleiter zufällig vorbeikommt?
 - c) Fehlermeldung lesen und versuchen selber zu beheben?

Compiler-Fehler

Beispiel:

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
Syntax error, insert ";" to complete Statement
    at Test.main(Test.java:11)
```

Hmmm.... Was will mir der Compiler denn sagen?

1. Es besteht ein ungelöstes Compiler-Problem.
2. Füg ein Semikolon ein, sonst ist die Anweisung nicht komplett.
3. Und zwar in der Zeile 11 von der Datei Test.java.

Compiler-Fehler

Weitere Beispiele:

Schließende geschweifte Klammer der Klasse vergessen

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
Syntax error, insert "}" to complete ClassBody
```

Variable nicht initialisiert

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
The local variable i may not have been initialized
```

Verwendung von einer nicht definierten Variablen

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
b cannot be resolved
```

Klasse hat nicht den selben Namen wie die Datei

```
Exception in thread "main" java.lang.NoClassDefFoundError: Test
```

Kommentare

Erklärungen im Programm

- **wichtig!**
- insbesondere für andere, aber auch für sich selbst um
 - die Übersicht zu bewahren
 - Code schneller verstehen zu können
 - Fehler zu vermeiden oder schneller zu finden

Verwendung

- Informationen zur gesamten Quelldatei wie z.B. Autoren, Erstellungsdatum, Mail-Adresse für Rückfragen, etc.
- Gliederung vom Quelltext
- Erläuterung von schwer verständlichen Code-Zeilen
- Hinweis auf zu erledigende Arbeit
"TODO" und "FIXME" werden i.d.R. beim Syntax Highlighting hervorgehoben
- Auskommentieren
für Passagen die vorübergehend „gelöscht“ werden sollen, aber später wieder gebraucht werden

Kommentare

in Java:

- **Code** /* Kommentar der über mehrere Zeilen gehen kann. Praktisch um größere Passagen auszukommentieren. */ **Code**
- **Code** // Kommentar, der bis zum Zeilenende geht. // Soll er weiter gehen, muss er erneut als // Kommentar gekennzeichnet werden.
- **Code** /** Dokumentationskommentar zur automatischen Erstellung von einer Dokumentation (z.B. Javadoc) dient. */ **Code**

Hinweis: "/*...*/"-Kommentare kann man nicht schachteln!

```
Code
/* Äußerer Kommentaranfang          ← Hier fängt der Kommentar ganz normal an
Code
/* Innerer Kommentaranfang          ← "/*" ist Teil des Kommentars
Code
Innerer Kommentarende */           ← Mit dem ersten "*/" endet der Kommentar
Code
Äußerer Kommentarende */           ← Der Kommentar führt zum Syntax-Fehler
Code
```

Letzte Tipps zu den Übungen

- Arbeitet am besten in einem Unterverzeichnis
- Rumpf-Klassen sind online verfügbar:
<http://www9.informatik.uni-erlangen.de/teaching/praproneu/>
- Versucht Compiler-Fehler erstmal selber zu beheben
- Nutzt die Folien als kleines Nachschlagewerk
- Helft euch gegenseitig, wenn ihr nicht weiterkommt
- Stellt den Übungsleitern möglichst konkrete Fragen
- Wenn ihr schnell fertig seid, helft denen die mehr Probleme haben

Wie geht's weiter?

Heute: Rechnerübungen

- Raum: 01.155 (blaues Hochhaus – 1. Stock)
- Zeit:
 - 11:00 – 15:00
 - 15:00 – 18:00

Morgen: Tafelübung (und anschl. Rechnerübungen)

- Raum: H4 (im RRZE)
- Zeit:
 - 11:15 – 12:00
 - 14:15 – 15:00

Hinweis:

Morgen ab 18 Uhr: Lange Nacht der Wissenschaften