



FACHSCHAFT INFORMATIK
HS Karlsruhe

Programmiervorkurs

Tag 1

Waldemar Felde

Ablauf



- 10:00 Uhr Begrüßung u. Vorlesung
- 11:30 Uhr Mittagspause
 - 60 Minuten
- gegen 12:30 Uhr Übungen im LI 136 u. LI 137

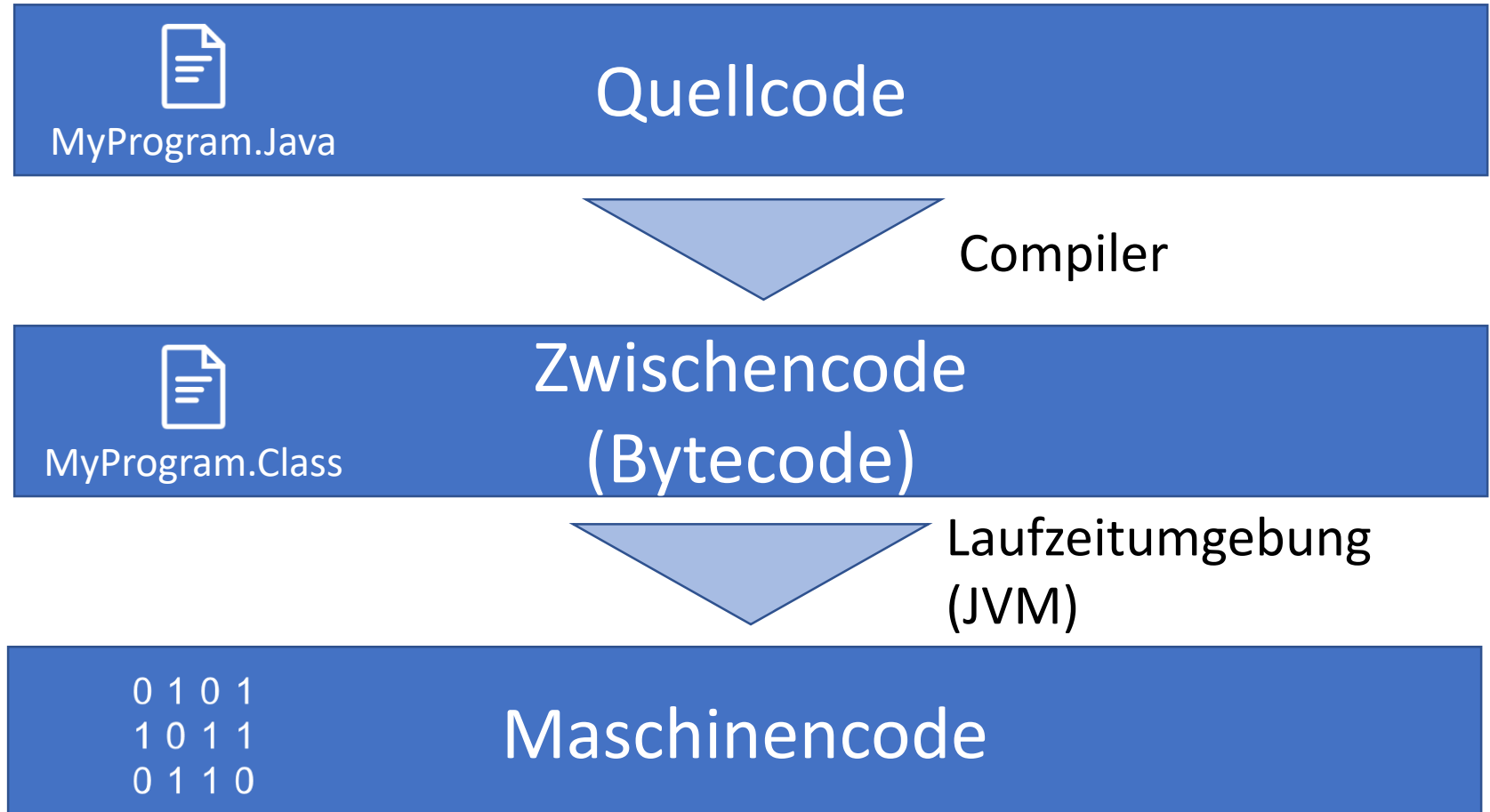
Inhaltsübersicht Vorkurs

- **Tag 1:** Zustände, Variablen, Datentypen, Konvertierungen, Arithmetik, Eclipse Live-Demo
- Tag 2: Kommentare, Boolesche Ausdrücke, If-Abfragen, Switch-Case
- Tag 3: Arrays, (Do-)While-Schleife, For-Schleifen, Weiterführung Debugging
- Tag 4: (statische) Methoden, Klassenvariablen

Inhalt Tag 1

- Java
- Variablen
- Datentypen
- Variablen
- Ausgabe auf der Kommandozeile
- Arithmetik
- Konvertierung
- Eclipse Live-Demo

Java



Variablen



- Speicher für Werte, die sich ändern können
- Speicher im Computer ist endlich
- Werden mit beliebigen Namen definiert
 - Gewisse Namen dürfen nicht verwendet werden. Dazu später!

Variablen

- Primitive Datentypen
 - Ganzzahlen
 - Fließkommazahlen
 - Wahrheitswerte
 - Zeichen
- Referenzdatentypen
 - Zeichenkette

Simple Datentypen

Komplexe Datentypen

Ganzzahlen

Datentypen



- Entspricht den ganzen Zahlen \mathbb{Z}
 - Keine Kommazahlen
- byte (8 Bit / 1 Byte)
- short (16 Bit / 2 Byte)
- int (32 Bit / 4 Byte)
- long (64 Bit / 8 Byte)

Bytes bestehen aus Bits. 8 Bits ergeben 1 Byte.

Kommazahlen (Fließkommazahlen)



FACHSCHAFT
INFORMATIK
HS Karlsruhe

Datentypen

- Entspricht den reellen Zahlen \mathbb{R}
- float (32 Bit / 4 Byte)
- double (64 Bit / 8 Byte)

Wahrheitswerte

Datentypen



- Beschreibt ob etwas wahr oder falsch ist
- boolean (Größe undefiniert)

Zeichen

Datentypen



- Ein einzelnes Zeichen → keine Zeichenkette!
- Zahlen sind auch Zeichen
- char (16 Bit / 8 Byte)

Zeichenkette

Datentypen



- Mehrere Zeichen aneinandergereiht
- String (abhängig von der Zeichenkette)

Wertebereiche

Datentypen



Typ	Größe (Bit)	Wertebereich
boolean	undefiniert	true oder false
char	16 Bit	Unicode Zeichen
byte	8 Bit	-128 bis 127
short	16 Bit	-32768 bis 32767
int	32 Bit	-2.147.483.648 bis 2.147.483.647
long	64 Bit	-9.223.372E12 bis ~9.223.372E12
float	32 Bit	1,4E-45 bis 3,4E+38
double	64 Bit	4,9E-324 bis 1,7E+308

Verwendung

Datentypen

Deklaration

Variablen

- Bekanntmachen der Variablen

- Schema:

Datentyp variablenname;

- Beispiele:

int age;

char gender;

boolean isStudent;

Initialisierung

Variablen

- Variablen einen Wert zuweisen
- Variable muss vorher deklariert worden sein

- Schema:

Variablenname = Wert;

- Beispiele

```
age = 20;
```

```
gender = 'm';
```

```
isStudent = true;
```


Deklaration + Initialisierung

Variablen

- Wert direkt beim Deklarieren zuweisen und Initialisieren
- Nennt man auch Definition

- Schema:

Datentyp Variablenname = Wert;

- Beispiele:

double average = 5.2;

boolean hasFun = true;

Vorgaben

Variablennamen



- Erlaubte Zeichen
 - Buchstaben, Zahlen und _
- Erstes Zeichen darf keine Zahl sein

- Gesperrte Namen:
 - **true**
 - **false**
 - **new**
 - **class**
 - ...

Konventionen

Variablennamen

- sinnvolle und aussagekräftige Namen wählen
- keine Abkürzungen
- Substantive
- Keine Umlaute
 - Kein ä, ö, ü, ß, ...
 - Stattdessen ae, oe, ue, ss, ..
- Kein mischen von Sprachen, z.B. „Denglisch“
- lowerCamelCase-Schreibweise

camelCase bedeutet, dass der Bezeichner ohne Trennzeichen wie Leerzeile und Unterstrich angegeben werden und die folgenden Worte groß geschrieben werden. Die lower Variante beginnt das erste Wort klein geschrieben.

Beispiele

Variablen



- Ganzzahlen

```
byte number = -128;
```

```
short newNumber = 32000;
```

```
int counter = 1337;
```

```
long bigNumber = 92233721;
```

- Fließkommazahlen

```
float balance = 42.01f;
```

```
double number = 1337;
```

- Wahrheitswerte

```
boolean isStudent = true;
```

```
boolean isEnabled = false;
```

- Zeichen

```
char gender = 'm';
```

```
char model = 'b';
```

- Zeichenkette

```
String name = "Max";
```

```
String textNumber = "42";
```

Ausgabe

- Wird benötigt, um Inhalte auf der Kommandozeile auszugeben
- Schema

```
System.out.println("Ausgabe");
```

Ausgabe mit einem Zeilenumbruch am Ende

```
System.out.print("Ausgabe");
```

Anhängen der Ausgabe am Ende

Beispiele

Ausgabe

```
int x = 1337;  
String name = "Max";  
  
System.out.println("x ist " + x + ".");  
System.out.println("Hallo, " + " Student.");  
System.out.println("Lieber " + name + ", Dankeschön!");  
System.out.println(x);
```

Ausgabe:

x ist 1337.

Hallo, Student.

Lieber Max, Dankeschön!

1337

Arithmetik

- Addition +
- Subtraktion -
- Multiplikation *
- Division /
- Inkrement ++
- Dekrement --
- Modulo %

- Ergebnis kann Variablen zugewiesen werden

- Beispiel

```
int result = 5 + 2;
```

```
double division = 3.5 / (result - 1);
```

Modulo (Restwertberechnung)

- Das Ergebnis des Modulo ist der Rest der Division
- Rechnen wie in der Grundschule (ohne Kommazahlen)

- Beispiele

$$26 / 5 = 5 \text{ [Rest 1]} \quad \Rightarrow \quad 26 \% 5 = 1$$

$$30 / 2 = 15 \text{ [Rest 0]} \quad \Rightarrow \quad 30 \% 2 = 0$$

Wozu?

- Zum Beispiel, um zu Prüfen ob eine Zahl gerade oder ungerade ist.
 - $3 \% 2 = 1 \rightarrow$ Rest 1, also ungerade
 - $30 \% 2 = 0 \rightarrow$ Rest 0, also gerade

Inkrement & Dekrement

- Erhöht bzw. verringert den Wert einer Variablen um jeweils 1
- Beispiel

```
int counter = 0;
```

```
counter++;
```

```
counter--;
```

Integer-Division

- Dividieren mit Integer-Variablen
- Beispiel

```
int x = 7;
```

```
int y = 2;
```

```
double z = x / y;
```

Welchen Wert beinhaltet die Variable z?

$z = 3.5$

$z = 3.0$

Integer-Division

- Dividieren mit Integer-Variablen
- Beispiel

```
int x = 7;
```

```
int y = 2;
```

```
double z = x / y;
```

Welchen Wert beinhaltet die Variable z?

z = 3.5

z = 3.0

Integer-Division

- Beispiel von gerade eben

```
int x = 7;
```

```
int y = 2;
```

```
double z = x / y;
```

- x und y wurden als Integer deklariert und initialisiert
- x / y wird zuerst ausgerechnet und dann in double gecastet
- Nachkommastellen werden einfach ignoriert!
- Kein Auf- oder Abrunden!

Dazu später mehr!

Explizite Konvertierungen

- Variablenwerte können umgewandelt werden (explizites „casten“)
- Beispiele

```
int x = 42;
```

```
short y = (short)x;
```

```
double pi = 3.14;
```

```
int b = (int)pi;
```

Implizite Konvertierungen

- Einige Typen können in andere ohne Probleme umgewandelt werden

- **byte** → **short** → **int** → **long** → **float** → **double**

- Beispiel:

```
int x = 42;
```

ist äquivalent zu

```
float y = x;
```



```
float y = (float)x;
```

Zurück zum Divisionsproblem

- Bei Rechnungen wird in den bestmöglichen Typ gecastet
- **byte** → **short** → **int** → **long** → **float** → **double**

Am besten Eingangsvariablen nach der Zielvariablen definieren

```
double x = 7.0;
```

```
double y = 2.0;
```

```
double z = x / y;
```

Eclipse Demo

Im Labor