



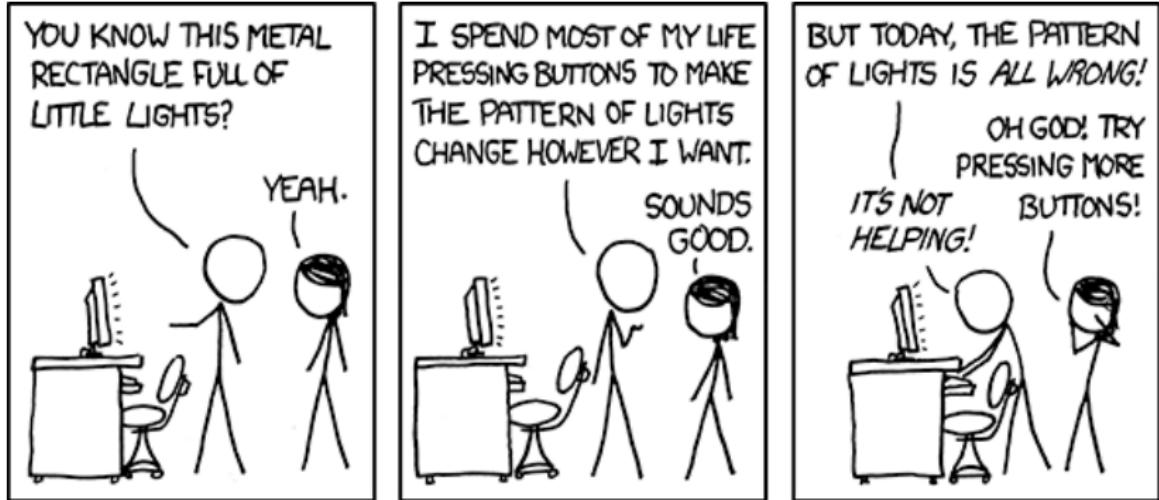
A photograph of a long, modern cable-stayed bridge stretching across a body of water under a clear blue sky. The bridge features multiple tall pylons and a dense network of cables supporting the roadway. A large, solid black circle is positioned in the upper right quadrant of the slide, containing the title text.

Hallo Java!, Eclipse, Variablen

Vorkurs Informatik



Getting started ...



©Randall Munroe, <http://xkcd.com/1513/>

Outline

- 1 Java als Programmiersprache
- 2 Eclipse
- 3 Variablen und Datentypen
- 4 Zusammenfassung

(nach Folien von L. Vettin, V. Weidler, W. Kessler)



Creative Commons Attribution-NonCommercial-ShareAlike 4.0 Licence

Outline

1

Java als Programmiersprache

2

Eclipse

3

Variablen und Datentypen

4

Zusammenfassung

Was ist Java?



Was ist Java?



- ▶ Eine der vier indonesischen Hauptinseln.

Was ist Java?



- ▶ Eine der vier indonesischen Hauptinseln.
- ▶ Eine Kaffeesorte.

Was ist Java?



- ▶ Eine der vier indonesischen Hauptinseln.
- ▶ Eine Kaffeesorte.
- ▶ Eine Technologie der Firma Oracle (früher: Sun), die sich aus der Programmiersprache Java, dem Entwicklungswerkzeug JDK und der Laufzeitumgebung JRE zusammensetzt.



© Robert Knapp www.modernartphotograph.com





Mensch:

Hey Computer,
hier hast du die
zwei Zahlen 5
und 7, addier sie
mir doch bitte!

Programmiersprachen



Mensch:

Hey Computer,
hier hast du die
zwei Zahlen 5
und 7, addier sie
mir doch bitte!

Computer:

0001 1101 0101
0101 0100 1010
1010 1101 0101
1000 0101 1010
0101 0100

Programmiersprachen



Mensch:

Hey Computer,
hier hast du die
zwei Zahlen 5
und 7, addier sie
mir doch bitte!

Computer:

0001 1101 0101
0101 0100 1010
1010 1101 0101
1000 0101 1010
0101 0100

Wie reden wir mit dem Computer so, dass er uns versteht?

Programmiersprachen



Mensch:

Hey Computer,
hier hast du die
zwei Zahlen 5
und 7, addier sie
mir doch bitte!

Computer:

0001 1101 0101
0101 0100 1010
1010 1101 0101
1000 0101 1010
0101 0100

Wie reden wir mit dem Computer so, dass er uns versteht?

⇒ Programmiersprachen

© <https://openclipart.org/detail/37135/personnageordinateur>

Programmiersprachen

- ▶ (Höhere) Programmiersprachen erlauben es uns, Programme in einer *menschenlesbaren* Form zu schreiben.
- ▶ Diese Programme müssen dann von einem speziellen Programm in eine *maschinenlesbare* Form gebracht werden.
- ▶ Dazu gibt es Compiler und Interpreter.

Warum Java?

- ▶ Umfangreiche Bibliotheken.
- ▶ Objektorientierung.
- ▶ Sicherheit.
- ▶ Effizienz.
- ▶ Weite Verbreitung.

Mein erstes Java-Programm

- ▶ Jedes Javaprogramm hat diese Struktur:

```
public class Name {  
  
    public static void main (String[] args) {  
  
        // hier Anweisungen einfügen  
  
    }  
}
```

- ▶ Bis auf `Name` ist alles in jedem Programm gleich.
- ▶ Ein Programm besteht aus Anweisungen (Befehlen).
- ▶ Die Anweisungen stehen immer zwischen den geschweiften Klammern hinter `main`.

Mein erstes Java-Programm: Hallo Welt!

- ▶ Hinter jeder Anweisung steht ein Semikolon.

```
public class HalloWelt {  
  
    public static void main (String[] args) {  
  
        System.out.println("Hello World!");  
        System.out.println("My name is ...");  
  
    }  
}
```

- ▶ Die Auswertung eines Programms geschieht Zeile für Zeile.
- ▶ Ist die dritte Zeile (`main`) nicht vorhanden oder steht etwas außerhalb der Klammern, kommt es zu Fehlern!

Ausgabe von Texten

Ausgabe

Die Anweisung `System.out.println(...)` gibt etwas auf der Konsole ("system out") aus.

```
System.out.println("Hello World!");  
System.out.println("My favourite number: " + 42 + "!");
```

- ▶ Innerhalb der Klammern steht genau eine Sache.
- ▶ Anführungszeichen beginnen und beenden einen Text (String).
- ▶ Mehrere Strings können mit + aneinandergehängt werden.

Kommentare

Kommentare

Kommentare sind an Programmierer gerichtete Erläuterungen, die in den Quelltext geschrieben werden. Kommentare sind besonders gekennzeichnet und werden daher vom Compiler ignoriert.

```
System.out.println("How are you?"); // Question  
System.out.println("I'm fine!"); // Answer
```

- ▶ Ein Kommentar beginnt mit `//`.
- ▶ Der Kommentar geht über den Rest der Zeile.

Quiz: Java-Programm

Welche Fehler haben sich eingeschlichen?

```
public Class HalloWelt {  
  
    public static void hallo (String[] args) {  
        System.out.println("Hello World!");  
    }  
    System.out.println("How are you?");  
}
```

Quiz: Java-Programm

Welche Fehler haben sich eingeschlichen?

```
public Class HalloWelt {  
  
    public static void hallo (String[] args) {  
        System.out.println("Hello World!");  
    }  
    System.out.println("How are you?");  
}
```

- ▶ Statt `hallo` muss `main` stehen.
- ▶ Der zweite `print`-Befehl darf nicht außerhalb der geschweiften Klammern von `main` stehen.
- ▶ `Class` muss klein geschrieben werden - Java achtet auf Groß-/Kleinschreibung.

Quiz: Hallo Welt!

Was ist die Ausgabe des folgenden Programms?

```
System.out.println("go swimming ");
System.out.println("Today i will "); // in the evening
// System.out.println("and have a party ");
System.out.println("with " + 5 + " of my friends ");
```

- ▶ go swimming Today i will and have a party with 5 of my friends
- ▶ Today i will with 5 of my friends in the evening go swimming and have a party
- ▶ go swimming Today i will with 5 of my friends
- ▶ go swimming Today i will in the evening and have a party with 5 of my friends

Quiz: Hallo Welt!

Was ist die Ausgabe des folgenden Programms?

```
System.out.println("go swimming ");
System.out.println("Today i will "); // in the evening
// System.out.println("and have a party ");
System.out.println("with " + 5 + " of my friends ");
```

- ▶ go swimming Today i will and have a party with 5 of my friends
- ▶ Today i will with 5 of my friends in the evening go swimming and have a party
- ▶ go swimming Today i will with 5 of my friends (korrekt)
- ▶ go swimming Today i will in the evening and have a party with 5 of my friends

Outline

1 Java als Programmiersprache

2 Eclipse

3 Variablen und Datentypen

4 Zusammenfassung

Entwicklungsumgebung

Entwicklungsumgebung, IDE

Eine IDE (integrated development environment) ist eine Sammlung von Anwendungsprogrammen (Texteditor, Compiler, Debugger, ...), mit denen die Aufgaben der Softwareentwicklung komfortabel bearbeitet werden können.

Vorteile einer IDE:

- ▶ Kennzeichnung von Syntax-Fehlern
- ▶ automatische Code-Ergänzung
- ▶ Zusatzfunktionen wie Code-Formatierung und Debugging
- ▶ erweiterbar durch Plug-Ins
- ▶ ...

Eclipse

Eclipse ist eine sehr weit verbreitete IDE für Java.



Installation unter Linux:

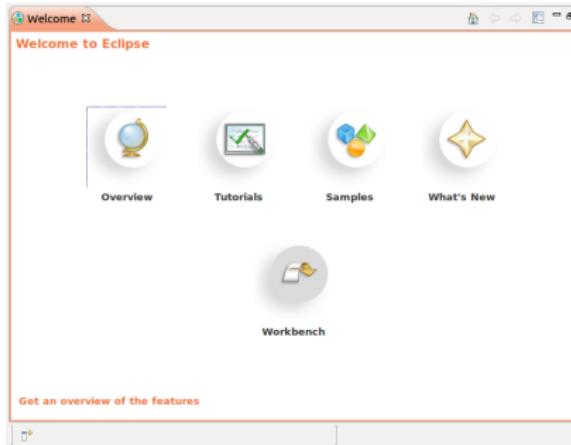
- ▶ Eclipse kann direkt aus den Paketquellen installiert werden
(z. B. in Ubuntu mit `sudo apt-get install eclipse`)

Installation unter Windows / Mac:

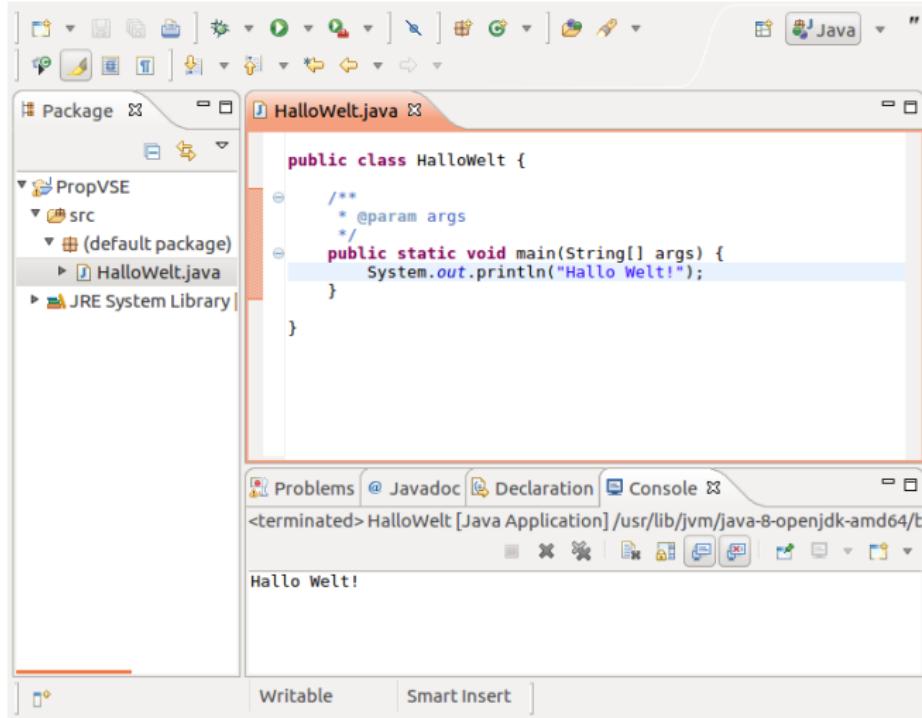
- ▶ Download von der Homepage: <http://www.eclipse.org>
- ▶ Auswahl: “Eclipse IDE for Java Developers”

Eclipse das erste Mal starten

- ▶ Der Workspace ist der Ordner, in dem alle erstellten Dateien gespeichert werden.
 - ▶ Kann unter File → Switch Workspace angepasst werden.
- ▶ Beim ersten Start kommt ein Welcome Screen. Diesen Tab einfach schließen (das 'x' neben "Welcome").



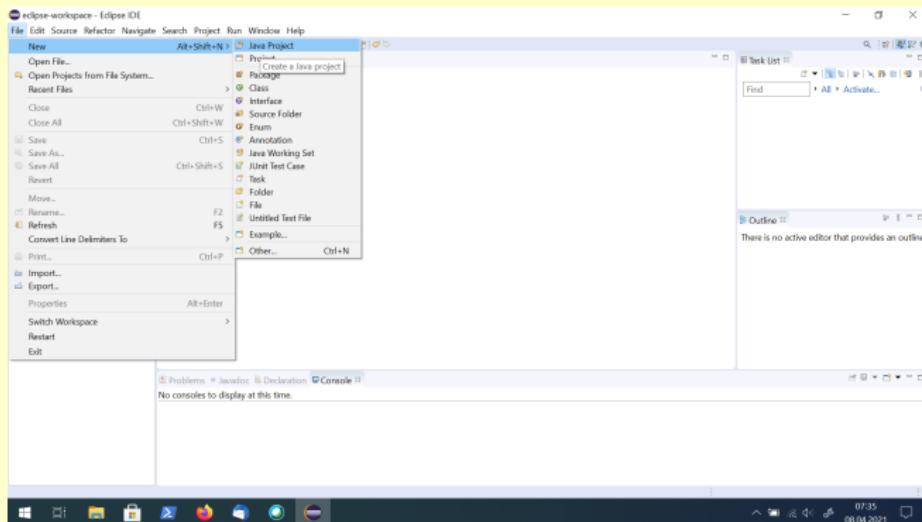
Oberfläche von Eclipse



Aufgabe: Projekt anlegen

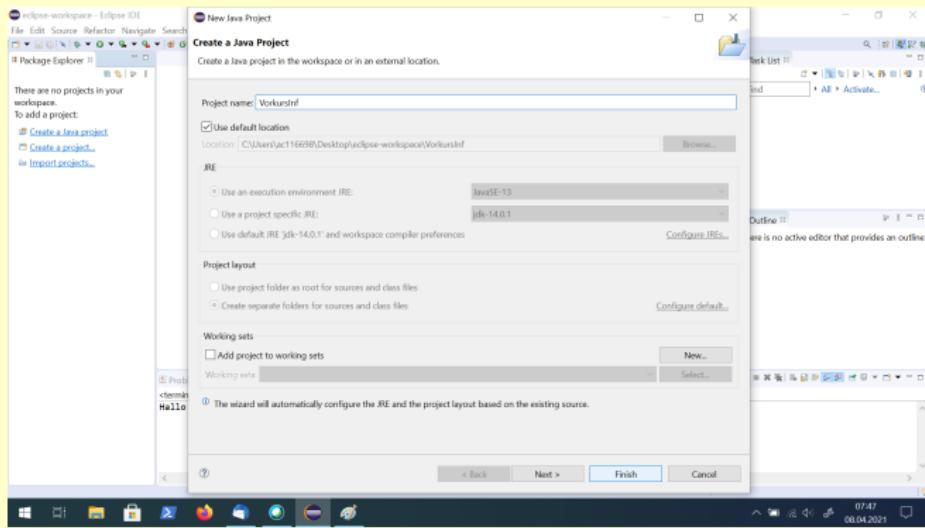
► Starten Sie Eclipse und legen Sie ein Projekt an:

File → New → Project → Java Project



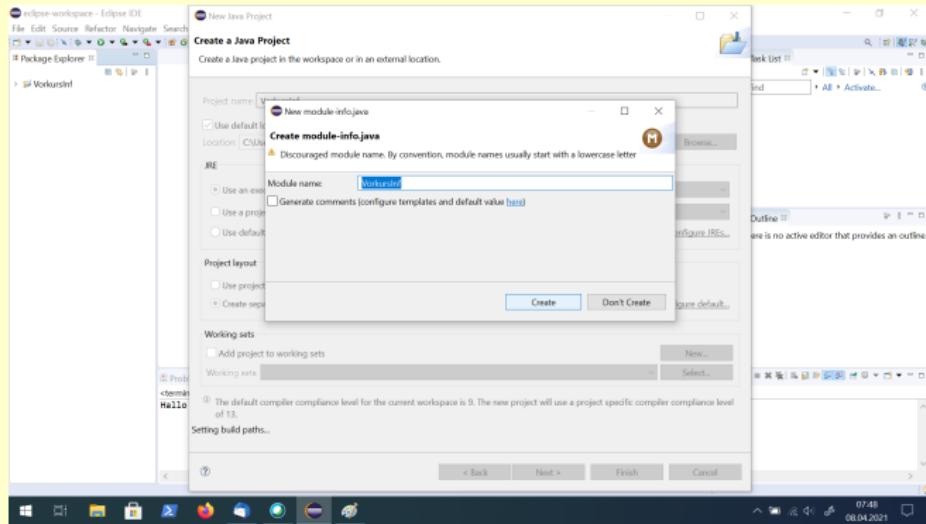
Aufgabe: Projekt anlegen

- Geben Sie dem Projekt den Namen `VorkursInf`.
Dann auf *Finish* klicken.



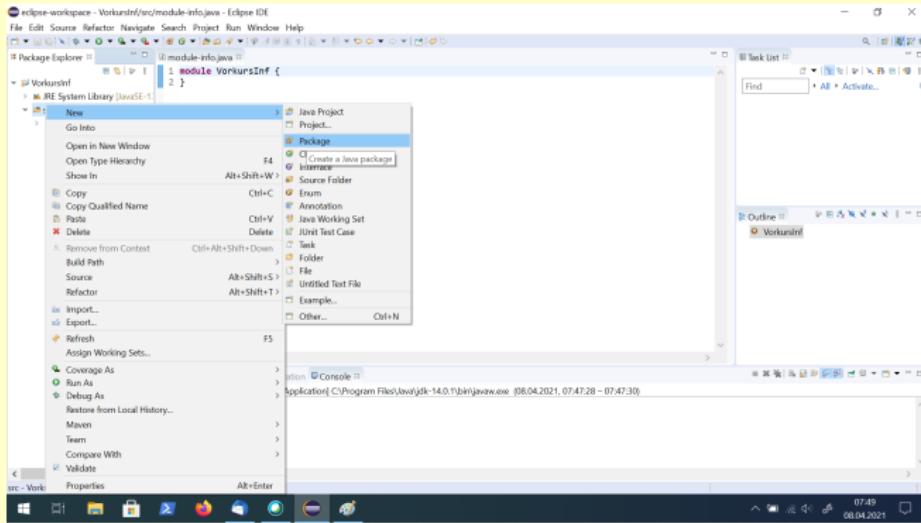
Aufgabe: Projekt anlegen

► Warnung ignorieren und auf *Create* klicken



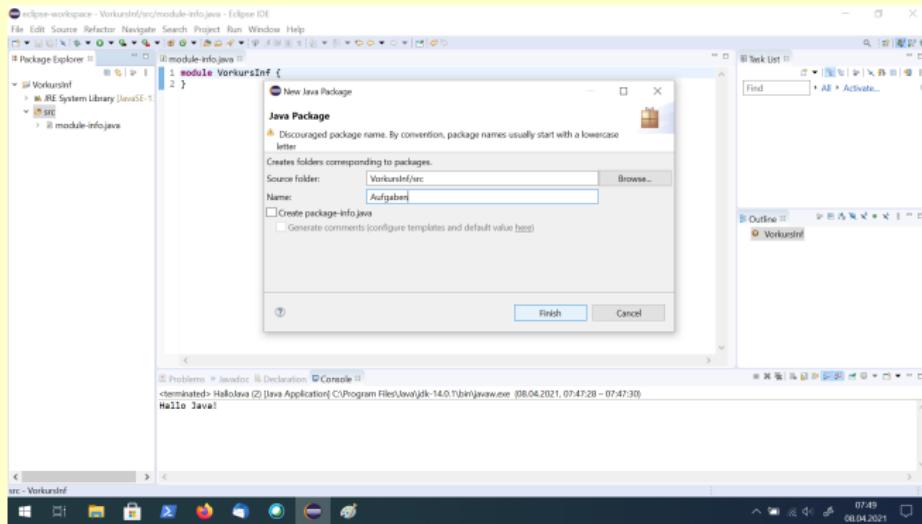
Aufgabe: Projekt anlegen

- ▶ Doppelklicken Sie auf `vorKursInf` und dann auf den Ordner `src`
 - ▶ Falls links noch nicht der Package Explorer geöffnet ist, öffnen Sie diesen (`Window → Show View → Package Explorer`)
 - ▶ Erstellen Sie ein neues Paket (`File → New → Package`).



Aufgabe: Projekt anlegen

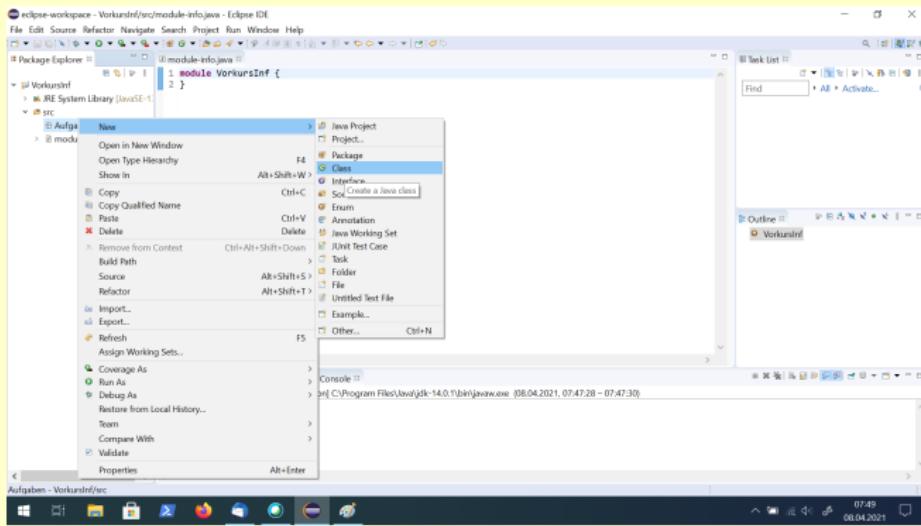
- Geben Sie als Name Aufgaben ein. Überprüfen Sie, dass der Eintrag bei *Source folder* korrekt ist.
- In diesem Paket bearbeiten wir gemeinsame Aufgaben.



- Für die Übungsblätter müssen Sie jeweils eigene Projekte anlegen.

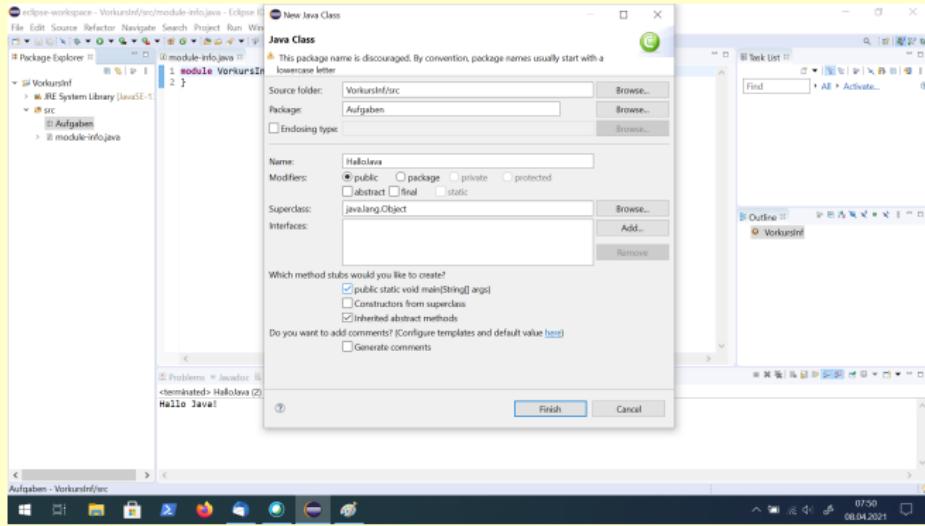
Aufgabe: Hallo Java!

- Klicken Sie auf das Paket Aufgaben im Package Explorer und erstellen Sie anschließend eine neue Klasse (File → New → Class) mit dem Namen HalloJava.



Aufgabe: Hallo Java!

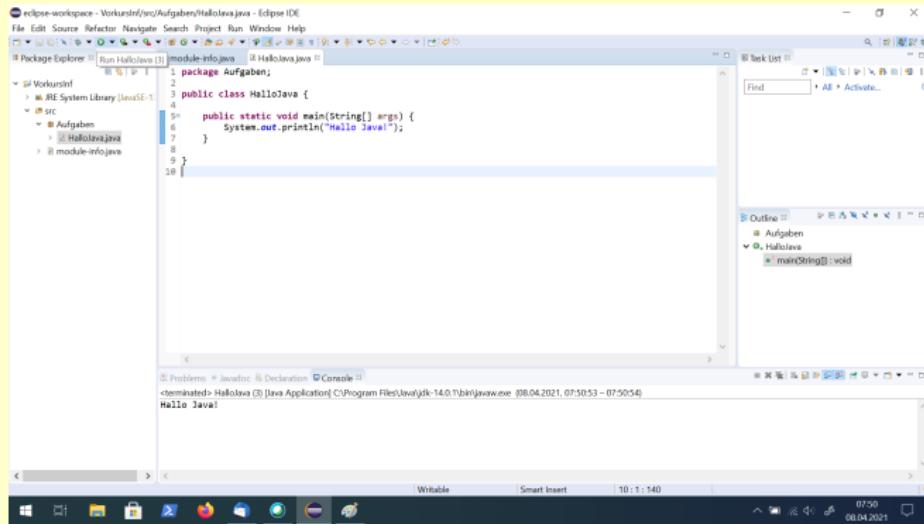
- Setzen Sie evtl. einen Haken bei
`public static void main(String[] args)`



Aufgabe: Hallo Java!

- Schreiben Sie Code ab und führen Sie das Programm aus (grüner Play-Button).

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



Outline

1 Java als Programmiersprache

2 Eclipse

3 Variablen und Datentypen

4 Zusammenfassung

Quiz: Variablen

- ▶ Was ist der Unterschied zwischen
 - ▶ ..., -3, -2, -1, 0, 1, 2, 3, ...
 - ▶ 3.4, 2.36, 7.01, ...
 - ▶ ja, nein
 - ▶ Hallo, Welt, ...
 - ▶ a, b, c, ?, :, ...

Quiz: Variablen

- ▶ Was ist der Unterschied zwischen
 - ▶ ..., -3, -2, -1, 0, 1, 2, 3, ...
 - ▶ 3.4, 2.36, 7.01, ...
 - ▶ ja, nein
 - ▶ Hallo, Welt, ...
 - ▶ a, b, c, ?, :, ...
- ▶ Es sind unterschiedliche Datentypen
 - ▶ Ganze Zahlen
 - ▶ Kommazahlen
 - ▶ Aussagen
 - ▶ Text
 - ▶ Einzelne(!) Buchstaben oder Zeichen

Variablen

Variable

Eine Variable ist ein Speicherplatz für Daten, die vom Programm gelesen und geschrieben werden können.

Variablen haben

- ▶ einen Datentyp,
- ▶ einen Namen,
- ▶ einen Wert.

Datentypen

Datentyp

Der Datentyp einer Variablen bestimmt,

- ▶ was man mit der Variable tun kann.
- ▶ wie viel Speicherplatz für die Variable reserviert wird.
- ▶ wie die Variable im Speicher dargestellt wird.

- ▶ Java kennt acht elementare Datentypen, u.a.
 - ▶ `int` für ganze Zahlen.
 - ▶ `double` für Kommazahlen.
 - ▶ `boolean` für Wahrheitswerte.
 - ▶ `char` für einzelne Zeichen.
- ▶ Daneben gibt es viele Objektdatentypen, z.B.
 - ▶ `String` für Texte.

Beispiel

- ▶ Ein einfaches Beispiel

```
int x = 4; // Datentyp, Name, Wert
int y;
y = x+4;
System.out.println(y); // gibt 8 aus
```

- ▶ Dem Programm muss mitgeteilt werden, dass es eine Variable `x` gibt und dass diese eine ganze Zahl speichern soll (`int`)
- ▶ Die Zuweisung eines Wertes kann gleich oder später erfolgen (wie bei `y`).
- ▶ Die Angabe des Datentyps (hier `int`) ist nur einmal nötig - wenn die Variable zum ersten Mal verwendet wird!

Speicherplatz für Variablen anlegen

Variablen-deklaration

Die Deklaration einer Variablen legt einen Speicherbereich von einem bestimmten Typ an, auf den über den Namen der Variablen zugegriffen werden kann.

```
// Deklariere die Variable 'number' vom Typ 'int'  
int number;  
// ... und die Variable 'firstName' vom Typ 'String'  
String firstName;
```

- ▶ Die Deklaration erfolgt genau einmal.
- ▶ Eine Variable muss deklariert werden, bevor sie verwendet werden kann.
- ▶ Bei der Deklaration werden Name und Datentyp festgelegt.
Diese können nicht mehr geändert werden.

Werte speichern

Zuweisung

Eine Zuweisung weist einer Variablen einen Wert zu. Die Variable muss vorher deklariert worden sein.

```
// Weise der Variablen 'number' den Wert 42 zu
number = 42;
number = 6 * 7 + 40 - (3 + 1) * 10;
```

- ▶ Für eine Zuweisung wird das Symbol = verwendet.
- ▶ Links vom = steht der Variablenname, rechts der Wert.
- ▶ Die Zuweisung liest sich von rechts nach links:
 - ▶ Auswertung/Berechnung der rechten Seite.
 - ▶ Speicherung des Ergebnisses in der Variablen der linken Seite.

Initialisierung

Initialisierung

Die erstmalige Zuweisung eines Wertes an eine Variable wird Initialisierung genannt.

```
// Deklariere die Variable 'number' vom Typ 'int'  
// Weise der Variablen 'number' den Wert 42 zu  
int number = 42;
```

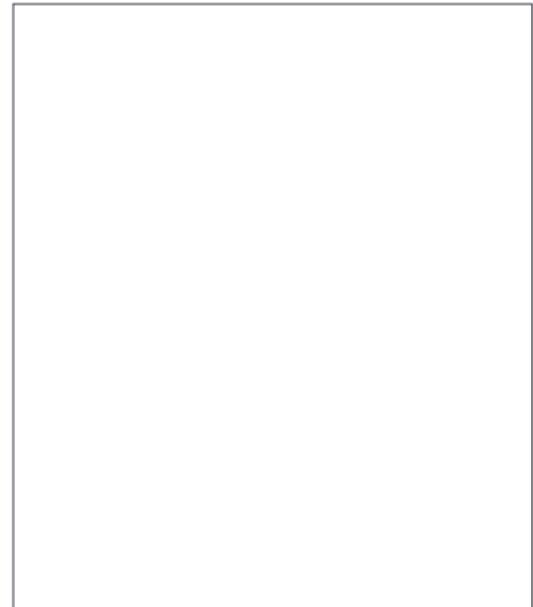
- ▶ Deklaration und Initialisierung können zusammen in einer Anweisung geschrieben werden.
- ▶ Es ist guter Stil, immer beides auf einmal zu machen!
- ▶ Es passieren trotzdem zwei getrennte Dinge, nacheinander.

Deklaration vs. Zuweisung

- Deklaration: Legt die Variable im Speicher an.

```
int number;  
String name;  
double decimalNumber;
```

Speicher:

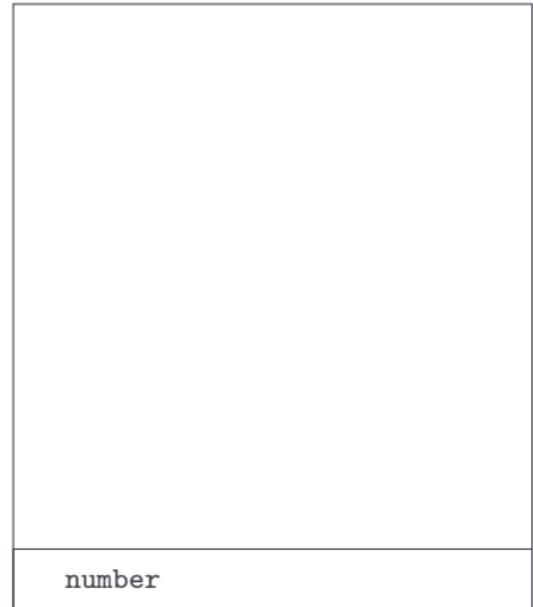


Deklaration vs. Zuweisung

- Deklaration: Legt die Variable im Speicher an.

```
int number;  
String name;  
double decimalNumber;
```

Speicher:

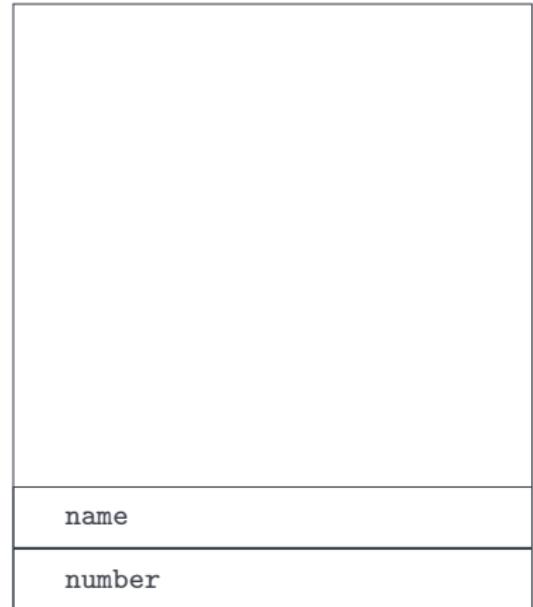


Deklaration vs. Zuweisung

- Deklaration: Legt die Variable im Speicher an.

```
int number;  
String name;  
double decimalNumber;
```

Speicher:

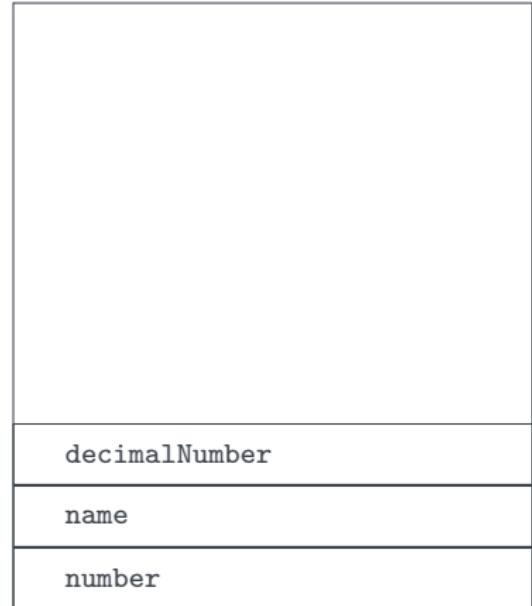


Deklaration vs. Zuweisung

- Deklaration: Legt die Variable im Speicher an.

```
int number;  
String name;  
double decimalNumber;
```

Speicher:



Deklaration vs. Zuweisung

- Deklaration: Legt die Variable im Speicher an.

```
int number;  
String name;  
double decimalNumber;
```

- Zuweisung: Weist der Variablen einen Wert zu.

```
number = 5;  
decimalNumber = 3.1415;  
number = 17;  
name = "Max Mustermann";  
number = 7839;
```

Speicher:

decimalNumber	
name	
number	5

Deklaration vs. Zuweisung

- Deklaration: Legt die Variable im Speicher an.

```
int number;  
String name;  
double decimalNumber;
```

- Zuweisung: Weist der Variablen einen Wert zu.

```
number = 5;  
decimalNumber = 3.1415;  
number = 17;  
name = "Max Mustermann";  
number = 7839;
```

Speicher:

decimalNumber	3.1415
name	
number	5

Deklaration vs. Zuweisung

- Deklaration: Legt die Variable im Speicher an.

```
int number;  
String name;  
double decimalNumber;
```

- Zuweisung: Weist der Variablen einen Wert zu.

```
number = 5;  
decimalNumber = 3.1415;  
number = 17;  
name = "Max Mustermann";  
number = 7839;
```

Speicher:

decimalNumber	3.1415
name	
number	17

Deklaration vs. Zuweisung

- Deklaration: Legt die Variable im Speicher an.

```
int number;  
String name;  
double decimalNumber;
```

- Zuweisung: Weist der Variablen einen Wert zu.

```
number = 5;  
decimalNumber = 3.1415;  
number = 17;  
name = "Max Mustermann";  
number = 7839;
```

Speicher:

decimalNumber	3.1415	
name	Max Mustermann	
number	17	

Deklaration vs. Zuweisung

- Deklaration: Legt die Variable im Speicher an.

```
int number;  
String name;  
double decimalNumber;
```

- Zuweisung: Weist der Variablen einen Wert zu.

```
number = 5;  
decimalNumber = 3.1415;  
number = 17;  
name = "Max Mustermann";  
number = 7839;
```

Speicher:

decimalNumber	3.1415	
name	Max Mustermann	
number	7839	

Besonderheiten

Was passiert?

```
int x = 4;  
int y = 5;  
y = 6;  
x = x*2;  
x += 1;  
y = x;  
y = y + 2;
```

Besonderheiten

Was passiert?

```
int x = 4;  
int y = 5;  
y = 6;  
x = x*2;  
x += 1;  
y = x;  
y = y + 2;
```

```
int x = 4;  
int y = 5;  
y = 6; // y wird überschrieben  
x = x*2; // x wird überschrieben (neuer Wert = alter Wert*2)  
x += 1; // x wird um 1 erhöht (alternativ: x = x+1)  
y = x; // y bekommt selben Wert wie x  
y = y + 2; // y wird um 2 erhöht, x nicht!
```

Fehlersuche mit dem Java Compiler

Fehlermeldungen beginnen mit Dateiname und Zeilennummer:

HalloWelt.java:4: error: ...

- ▶ x cannot be resolved to a variable.
→ Deklaration für x fehlt oder x ist falsch geschrieben.
- ▶ The local variable x may not have been initialized.
→ Erste Wertzuweisung an x (Initialisierung) fehlt.
- ▶ Variable x is already defined in method main().
oder Duplicate local variable x.
→ doppelte Deklaration für x.
- ▶ Syntax error, insert ";" to complete Statement.
oder Error: ';' expected.
→ vergessenes Semikolon am Ende.

Outline

1

Java als Programmiersprache

2

Eclipse

3

Variablen und Datentypen

4

Zusammenfassung

Zusammenfassung: Erste Schritte in Java

- ▶ Eine **Variable** ist ein Speicherplatz für Daten und hat einen Namen, einen Datentyp und einen Wert.
- ▶ Der **Datentyp** bestimmt die Darstellung, Verwendung und Speicherplatzgröße einer Variablen.
- ▶ Eine **Zuweisung** weist einer Variablen einen Wert zu.



Vielen Dank!



Frank Schweiner

E-Mail frank.schweiner@mint-kolleg.de
Telefon +49 (0) 711 685-84326
Fax —

Universität Stuttgart
MINT-Kolleg Baden-Württemberg
Azenbergstr. 12
70174 Stuttgart