Programmieren 1

01 - Organisation und Einführung

Agenda

1) Organisation

2) Rechnerraum

3) Einführung

4) Übung

Übersicht Programmieren 1

Dozenten

- Prof. Dr. Melanie Baur (Informatik)
- Prof. Dr. Alexander Rausch (Wirtschaftsinformatik)

Termine

- Dienstags, 5. Block
- Mittwochs, 1. + 2. Block

Tutorium

- Samuel Maier
- Freitag, 4. Block

Nur IF

Prüfungen (SPO INF)

Modul		LV	SWS	CP	Prüfungs-	Leistungs-	Prüfungs-
Kürzel					vorleistung	nachweis	leistung
Grundstudium			**50**	**60**	**7**	**2**	**9**
1. Semester			**26**	**30**	**3**	**2**	**4**
MAT1	Mathematik 1	V	6	8	SC 2)		KL 90
DIM	Diskrete Mathematik	V	4	5	SC 2)		KL 90
EIF	Einführung in die Informatik	V		4	SC 2)		KL 60
GDI	Grundlagen der Informatik		2				
REP	Rechnerpraxis		2				
PRO1	Programmieren 1	V	6	7		PA 1)	
BWL	Betriebswirtschaftslehre	V		4			KL90
FSP1	Fremdsprachen 1		2	2		PA 1)	

7 CP entspricht ca. 7*30h=210h Arbeitsaufwand über das Semester verteilt
D.h. ca. 15h/Woche!

1) Der Leistungsnachweis ist Zulassungsvoraussetzung für die Prüfung im gleichnamigen Modul des 2. Semesters.

Nur WI

Prüfungen (SPO WINF)

Modul			Lehr-	SWS	CP	Prüfungs-	Leistungs-	Prüfungs-
Kürzel	ggf. Lehrveranstaltung		veranst.			vorleist.	nachweis	leistung
Grundstudium				**50**	**60**	**6**	**2**	**9**
1. Semester				**26**	**31**	**3**	**2**	**3**
MAT1		Mathematik 1	V	4	5		PA 1)	
DIM		Diskrete Mathematik	V	4	5	SC 2)		KL 90
GBWL		Grundlagen der BWL	V	4	5			KL 90
GWI		Grundlagen der Wirtschaftsinformatik	V	6	6	SC 2)		KL 90
	E-WI	Einführung in die Wirtschaftsinformatik						
	E-Inf	Einführung in die Informatik						
PRO1		Programmieren 1	V	6	8		PA 1)	
FSP	FSP1	Fremdsprachen 1	V	2	2	PA 2)		

1) Der Leistungsnachweis im Modul Mathematik 1 ist Zulassungsvoraussetzung für die Modulprüfung Mathematik 2, **der Leistungsnachweis im Modul Programmieren 1 ist Zulassungs-voraussetzung für die Modulprüfung Programmieren 2**.

Nur IF

Vorläufige Themenübersicht (Informatik)

Woche	Thema (Vorlesung, Mittwoch)	Buch
1	Variablen, Datentypen	2
2	Kontrollfluss	3
3	Strings	4
4	Klassen	5
5	1. Teilprüfung	
6	Vererbung	6
7	Abstrakte Klassen, Interfaces	7
8	Arrays, Collections, Maps	8
9	Exceptions, Dateien	9+11
10	2. Teilprüfung	
11	Programmierzeit	
12	Programmierzeit	
	Weihnachtsferien 26.12.-06.01.23	
13	Programmierzeit	
14	Präsentationen	

Jede Woche wird ca. 1 Kapitel aus dem Buch „Schrödinger programmiert Java“ besprochen.

Nur IF

Geplanter Rhythmus in den ersten 10 Wochen (Informatik)

Dienstag: Übung

- Besprechung der Übungen aus der Hausaufgabe
- Klärung aller Fragen
- Erklärung der Lösungen
- Bei Bedarf geführtes Programmieren durch schwierige Teile

Mittwoch: Vorlesung

- Bearbeitung einer Themeneinheit
- Aktive Mitarbeit durch die Studierenden
- Integrierte Beispiele und Übungen

Mittwoch: Übung

- Selbstständiges Programmieren einer auf die Vorlesung aufbauenden Übungen
- Erklärung der Lösungen
- Klärung aller noch offenen Fragen
- Weitere Übungen als Hausaufgabe und Vertiefen des Stoffs

Nur WI

Wochenplan (Wirtschaftsinformatik)

KW	Vorlesung	Übung	Thema	Kapitel	Seiten		Anzahl
					von	bis	
40	04.10.2022	05.10.2022	Einführung, Variablen, Datentypen	2	61	98	38
41	11.10.2022	12.10.2022	Kontrollfluss	3	99	134	36
42	18.10.2022	19.10.2022	Strings	4	135	168	34
43	25.10.2022	26.10.2022	Klassen	5	169	230	62
44	(Feiertag)	02.11.2022	1. Teilprüfung				
45	08.11.2022	09.11.2022	Vererbung	6	231	264	34
46	15.11.2022	16.11.2022	Abstrakte Klassen und Interfaces	7	265	302	38
47	22.11.2022	23.11.2022	Arrays, Collections, Maps	8	303	354	52
48	29.11.2022	30.11.2022	Exceptions	9	355	388	34
49	06.12.2022	07.12.2022	Dateien Teil 1, 2. Teilprüfung	11	415	432	18
50	13.12.2022	14.12.2022	Programmieraufgabe (1)				
51	20.12.2022	21.12.2022	Programmieraufgabe (2)				
52	27.12.2022	28.12.2022	Ferien				
1	03.01.2023	04.01.2023	Ferien				
2	10.01.2023	11.01.2023	Programmieraufgabe (3)				
3	17.01.2023	18.01.2023	Programmieraufgabe (4), Bewertung				

Leseplan:
- Jede Woche ein Kapitel
- ca. 40 Seiten / Woche

Nur WI

Geplanter Rhythmus in den ersten 10 Wochen (Wirtschaftsinformatik)

Dienstag: Vorlesung

- Bei Bedarf: Nachklapp zum Vorlesungsthema der Vorwoche
- Aktuelles Thema (siehe Wochenplan)
- Zwischenfragen erwünscht

Mittwoch: Übung

- Bei Bedarf: Nachklapp zu ausgewählten Übungsaufgaben der Vorwoche
- Übungsaufgaben nach Schwierigkeitsgrad sortiert
  - Selbstständiges Programmieren
  - Geführtes Programmieren nur ausnahmsweise bei schwierigen Aufgabenteilen
  - Erläutern der Lösung für ausgewählte Aufgaben
  - Beantwortung offener Fragen

Donnerstag bis Montag

- Selbständiges Programmieren weiterer Aufgaben
- Nacharbeit und Vorbereitung – Buchlektüre (siehe Leseplan)

Umfassende Programmierübung (4 Wochen)

- Größere Programmierübung in 2er Teams
- Eigenständiges Programmieren über 4 Wochen
- Zwischenpräsentation „Sprint Review“
- Bewertete Endpräsentation

Leistungsnachweis Programmieren 1

- 2 Programmiertests
  - Voraussichtlich am 02.11.2022 und 07.12.2022
  - Voraussichtlich 40 Punkte im 1. Test und 60 Punkte im 2. Test
  - In beiden Tests zusammen müssen mind. 50% der Punkte (=50 Punkte) erreicht werden
- Umfassende Programmierübung
  - 50% der Punkte müssen mind. erreicht werden
- Rechtzeitige Anmeldung über Moodle notwendig
  - Später zusätzlich auch eine Anmeldung im LSF erforderlich!

Moodlekurs und Übungen

- https://moodle.hft-stuttgart.de/course/view.php?id=1168
- Zentrale Kommunikationsplattform
  - Vorlesungsunterlagen
  - Termine
  - Etc.
- Übungen und Lösungen werden hier bereit gestellt:
- https://speiser.hft-pages.io/programmieraufgaben/2022-ws-pro-1/

Literatur

- Hauptliteratur:
  Philip Ackermann, Schrödinger programmiert Java: Das etwas andere Fachbuch, Rheinwerk Verlag
  Zugang per Campusnetz, VPN, Shibboleth
  https://ebookcentral.proquest.com/lib/hft-stuttgart/detail.action?docID=6382971

- Ergänzend bei Bedarf:
- Jedes sonstige Java-Grundlagenbuch
- RRZN-Skripte von Prof. Dr. Peter Heusch, Infos hier:
  https://www.luis.uni-hannover.de/de/services/schulung-beratung-und-support/handbuecher/bezugsquellen/details/sources/hft-stuttgart/

Guter Rat

- Besuchen Sie alle Lehrveranstaltungen regelmäßig.
- Grundsätzlich ist der gesamte Lehrstoff von PRO1 und PRO2 für die Klausur PRO2 (!) relevant!
- Folien enthalten Stichworte. Bereiten Sie die Vorlesung mit der Literatur (Ihrer Wahl) regelmäßig nach.
- Bei Problemen fragen Sie zunächst Ihre Lerngruppe, dann die Tutoren oder Dozenten!
- Lesen Sie regelmäßig Ihre Mails und Moodle!

Fragen

Agenda

1) Organisation

2) Rechnerraum

3) Einführung

4) Übung

Einloggen

- User-ID
  - z.B. 21muma1bif (für Max Mustermann)
- Passwort

Emails lesen per Webmail

- Webmail unter mail.hft-stuttgart.de
  - User-ID
  - Passwort

- Auch zu finden unter https://www.hft-stuttgart.de/quicklinks

- Bitte lesen Sie täglich Ihre HFT-Mails!

Laufwerke

C

- Temporäres Laufwerk
- Daten werden beim Logout gelöscht! (gilt auch für Desktop)

O

- Öffentliches Laufwerk für Studenten
- Zum Austausch von Daten
- Jeder Student kann hier Daten löschen
- Kein Backup

P

- Privates Laufwerk
- 1GB Speicherplatz
- Backup

Agenda

1) Organisation

2) Rechnerraum

3) Einführung

4) Übung

Einführung

- Programmieren
- Java
- Elemente eines Programms
- Übersetzen und Ausführen eines Programms

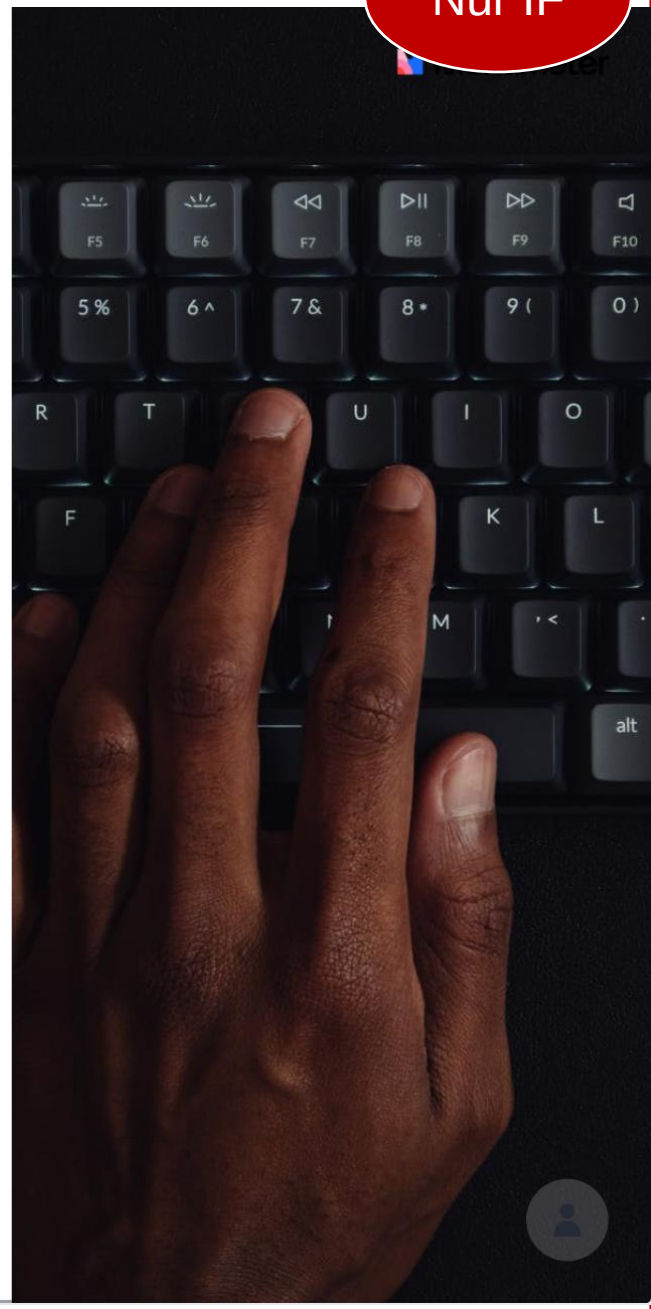
Nur IF

Was ist Ihr momentaner Berufswunsch?

Press **S** to show image

Nur IF

Welche Fähigkeiten brauchen Sie hierzu?



Press S to show image

Nur WI

Warum Wirtschaftsinformatik (und weder BWL noch Informatik)?

- Beispiele (bitte Präfix verwenden)
  - „BWL: Leider keinen Studienplatz erhalten ☹“
  - „IF: Cola und Chips machen dick“
  - „WI: Ich sehe $ und €“
  - *… Sie haben bestimmt gute und ernstzunehmende Ideen …*

- Gehen Sie auf www.menti.com
- Geben Sie den Code … an

Nur WI

Welche Art von Programmieraufgaben erwarten Sie im Berufsleben?

- Zum Nachdenken, bevor Sie Antworten geben
  - Worin könnten sich Programmieraufgaben zwischen Informatikern und Wirtschaftsinformatikern unterscheiden?
  - Was können Informatiker besser als Wirtschaftsinformatiker?
  - Was können Wirtschaftsinformatiker besser als Informatiker?
- Beispiele
  - …

- Gehen Sie auf www.menti.com
- Geben Sie den Code … an

Programmieren

Eine Programmiersprache zu lernen, ist unabdingbare Voraussetzung für eine Tätigkeit in der Informatik, z.B. als Entwickler, Projektleiter, Softwarearchitekt.

- Kennen Sie einen Trainer der Bundesliga, der nicht Profi-Fußballer war?
- Haben Sie Schwimmen oder Radfahren aus Büchern erlernt?
- Sind Sie zur praktischen Fahrprüfung erstmals Auto gefahren?
- Wie viele Jahre hat ein Musiker vor dem ersten Klavierkonzert geübt?

Programmieren

- Unter Programmieren versteht man das Erstellen von Anweisungen für eine Maschine.
- Im Rahmen der nächsten zwei Semester:
  - PRO 1: Erwerb der Fähigkeit, ein (mittelgroßes) Java-Programm zu schreiben
  - PRO 2: Erweiterung um die „Kommunikation mit der Außenwelt“ (z.B. GUIs, Datenbanken,...).

Programmieren

Ein Programm ist vergleichbar mit einem Kochrezept:

Ziel ist einen/mehrere fertige Pfannkuchen aus verschiedenen einzelnen Zutaten zu kochen.

Kochrezepte sind wie „**imperative**“ Programmierung:
von lat. imperare: anordnen, befehlen.

Pfannkuchen

Zutaten

0.5 l	Milch
4 Stk	Eier
300 g	Mehl
2 EL	Öl
1 TL (gestrichen)	Salz
2 EL	Zucker

Zubereitung

Die Milch mit dem Mehl gut verrühren und die Eier dazu geben. Salz, Zucker und einen EL ÖL dazugeben und nochmal kräftig verrühren bis man einen glatten Teig ohne Klümpchen hat. Eine beschichtete Pfanne erhitzen (mittlere Hitze). Etwas Küchenpapier mit Öl tränken und die heiße Pfanne damit ausreiben. Teig in der Pfanne gleichmäßig verteilen, so dass ein schöner runder Pfannkuchen entsteht. Wenn die Unterseite goldgelb ist den Pfannkuchen wenden.

Sie haben bereits:

- Mehl
- Öl
- Salz

Was brauchen Sie noch?

_____

_____

_____

Quelle: mat.madoo.net

Programmieren

Das Rezept wurde als eine Art Programm dargestellt:

```
liter milch       = 0,5;
gramm mehl        = 300;
stueck eier       = 4;
essloeffel oel    = 2;
teeloeffel Salz   = 1;
essloeffel Zucker = 2;

void Rühren();
void Braten();

Pfannkuchen_Machen {
  Teig = milch + mehl;
  Rühren(Teig);
  Teig = Teig + Salz + Zucker + Öl;
  Rühren(Teig);
  Pfannkuchen = Braten(Teig);
}
```

Variablen

Methoden

Hauptprogramm

Pfannkuchen

Zutaten

0.5 l	Milch
4 Stk	Eier
300 g	Mehl
2 EL	Öl
1 TL (gestrichen)	Salz
2 EL	Zucker

Zubereitung

Die Milch mit dem Mehl gut verrühren und die Eier dazu geben. Salz, Zucker und einen EL ÖL dazugeben und nochmal kräftig verrühren bis man einen glatten Teig ohne Klümpchen hat. Eine beschichtete Pfanne erhitzen (mittlere Hitze). Etwas Küchenpapier mit Öl tränken und die heiße Pfanne damit ausreiben. Teig in der Pfanne gleichmäßig verteilen, so dass ein schöner runder Pfannkuchen entsteht. Wenn die Unterseite goldgelb ist den Pfannkuchen wenden.

Sie haben bereits:

- Mehl
- Öl
- Salz

Was brauchen Sie noch?

_____

_____

_____

Quelle: mat.madoo.net

Programmieren

- Programmieren lernen ähnelt dem Lernen einer Fremdsprache
- Programmieren lernt man durch wiederholte Ausführung folgender Tätigkeiten:
  - Lesen und Verstehen anderer Programme
  - Schreiben eigener Programme
  - Testen und Fehlersuche in eigenen Programmen
- Die Beherrschung geeigneter Werkzeuge ist unabdingbarer Teil der Programmierausbildung.

Einführung

- Programmieren
- Java
- Elemente eines Programms
- Übersetzen und Ausführen eines Programms

Java

- Java ist eine weitverbreitete Programmiersprache.
- Java unterstützt viele Programmierparadigmen.
- Java wird ständig weiter entwickelt und um moderne Konzepte ergänzt
- Mit Java stellen wir uns auf „die Schultern eines Riesen“:
  - Hunderte Pakete
  - Tausende vordefinierte Klassen

Java

- Entwicklung in den frühen 90er Jahren unter dem Namen Oak als Programmiersprache für digitale Satellitenempfänger
- Erste Verbreitung als Technik zur Erstellung interaktiver Web-Inhalte (Applets)
- Heute sehr stark im Server-Bereich genutzt, aber auch für Desktop-Anwendungen
- Erfunden durch die Firma Sun Microsystems, seit einigen Jahren ein offener Standard
- Kommerzieller Vertrieb durch Oracle
- Die meisten Features sind kostenfrei nutzbar

Einführung

- Programmieren
- Java
- Elemente eines Programms
- Übersetzen und Ausführen eines Programms

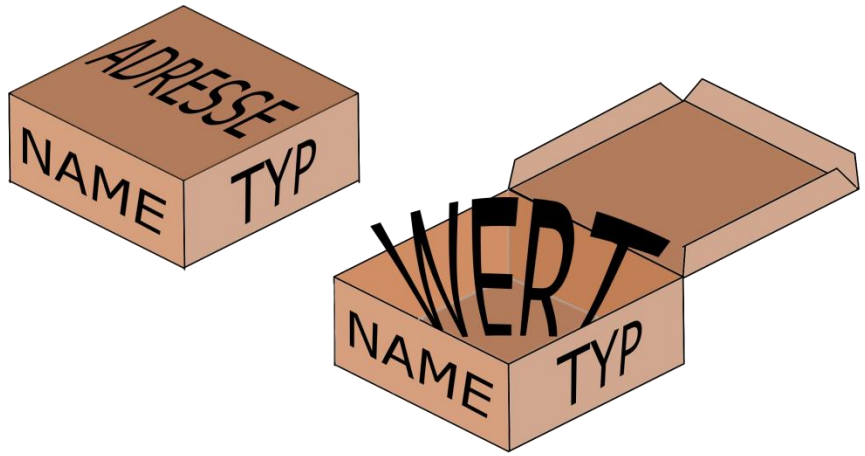
Bestandteile eines Programms

- Ein Computerprogramm besteht im wesentlichen aus verschiedenen Einzelkomponenten:
  - Variablen = Speicherplatz
  - Methoden = Werkzeug (weiteres folgt…)

- Es gibt eine strikte „Rechtschreibung“ (Syntax), die in jeder Programmiersprache unbedingt eingehalten werden muss!

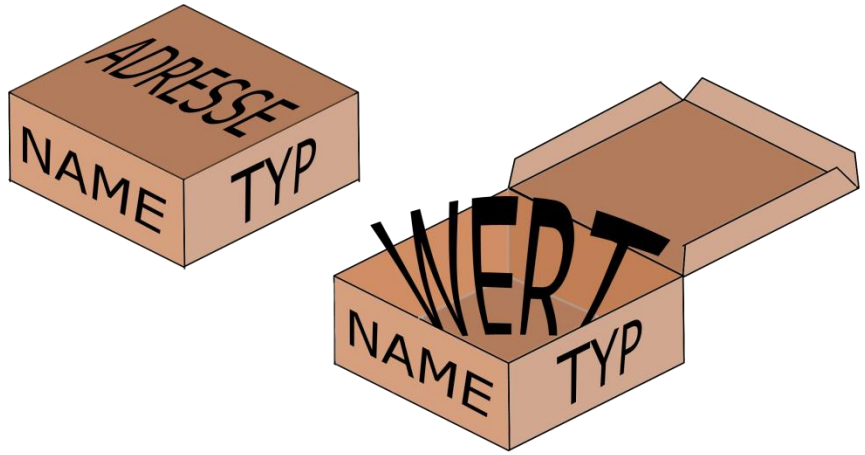
Bestandteile eines Programms

- Variablen sind eine Art Gefäß im Speicher. Werte können überprüft, verändert und vor allem gespeichert werden.
- Variablen haben einen bestimmten **TYP**,
- eine **ADRESSE** im Speicher, einem **NAMEN** und natürlich einem **WERT**.



Bestandteile eines Programms

- Variablentypen:
  - Ganzzahlen
  - Fließkommazahlen
  - Zeichenketten
  - Boolsche Werte
  - …



- Adressen sind in einem Computer die Orte, an denen etwas im Speicher abgelegt ist. Das „etwas“ hat einen Namen, so dass sich der Programmierer nicht mit Adressen beschäftigen muss. An einer Adresse gibt es „Platz“ für einen Wert. Je nach Typ ist ein bestimmter Bereich (Speichergröße) reserviert.

Bestandteile eines Programms

- Benutzung von Variablen
  - Deklaration:

```
int schuhgroesse;
```

  - Initialisierung:

```
schuhgroesse = 45;
```

  - Deklaration mit Initialisierung :

```
int schuhgroesse = 45;
```

Bestandteile eines Programms

Operatoren

- In der Regel will man mit Werten, die man in Variablen gespeichert hat, im Verlauf eines Programms Berechnungen durchführen.
- Hierfür gibt es in der Java-Programmierung die Operatoren.
- Die Operatoren werden innerhalb von Methoden verwendet.

Bestandteile eines Programms

Operatoren

Operator	Beispiel	Wirkung
+	a + b	Addiert a und b
–	a - b	Subtrahiert b von a
*	a * b	Multipliziert a und b
/	a / b	Dividiert a durch b
%	a % b	Liefert den Rest bei der ganzzahligen Division a / b

(Neben den oben genannten Operatoren gibt es noch weitere…)

Aufbau eines Programms

In Java besteht Programmcode aus kleineren Einzelteilen:

- Hauptklasse oder Startklasse: Oberste Struktureinheit in Java
- Hauptmethode oder main-Methode genannt: Hier läuft das eigentliche Programm ab, ohne die main-Methode läuft das Programm nicht.
- Weitere Methoden: Es können für Teilberechnungen weitere Methoden definiert werden, die dann von der main-Methode aufgerufen werden.

Aufbau eines Programms

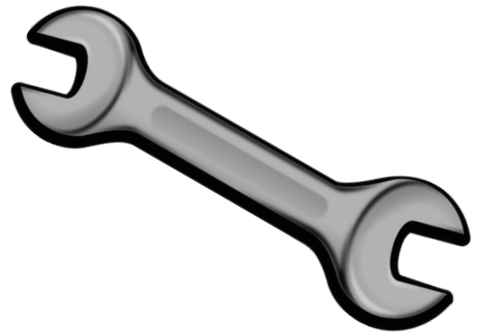
Beispiel

```
public class Uebung {
    public static void main (String args[]) {
        tue_was(4711);
    }
}
```

- Ein Java-Programm ist in (Klassen), Methoden und Blöcke aufgeteilt, die ineinander verschachtelt sind.
- Blöcke werden durch die **{}**-Klammern geschachtelt.
- Der äußere Block umfasst die Klasse **Uebung**, der innere Block bildet den Methodenrumpf von **main** mit dem Aufruf der Methode **tue_was**.

Aufbau eines Programms

- Methoden sind gespeicherte Anweisungen innerhalb eines Programms. Sie werden für einen bestimmten Zweck geschrieben und aufgerufen.
- Methoden kann man auch als Werkzeug betrachten. Man kann ein Werkzeug besitzen, aber es gar nicht einsetzen.
- Analog kann man in einem Programm eine Methode schreiben, sie jedoch im eigentlichen Hauptprogramm nicht benutzen.



Das erste Programm

```
/**
 * Mein erstes Java-Programm.
 * Übersetzen mit javac -d . HelloWorld.java
 * Ausführen mit java de.hft_stuttgart.hepe1bif.HelloWorld
 * @author Peter Heusch
 */
package de.hft_stuttgart.hepe1bif;
public class HelloWorld {
    public static void main(String [] args) {
      System.out.println("Hello World!");
  }
}
```

Achtung: Der Klassenname muss mit dem Dateinamen übereinstimmen!

Das erste Programm

- Jedes Programm beginnt mit einem Kommentar zur Einleitung
- Danach folgt die package-Anweisung: Sie dient zum „Einsortieren“ des Codes ins Verzeichnissystem.
- Zum Schluss kommt das eigentliche Programm mit den ausführbaren Anweisungen innerhalb von

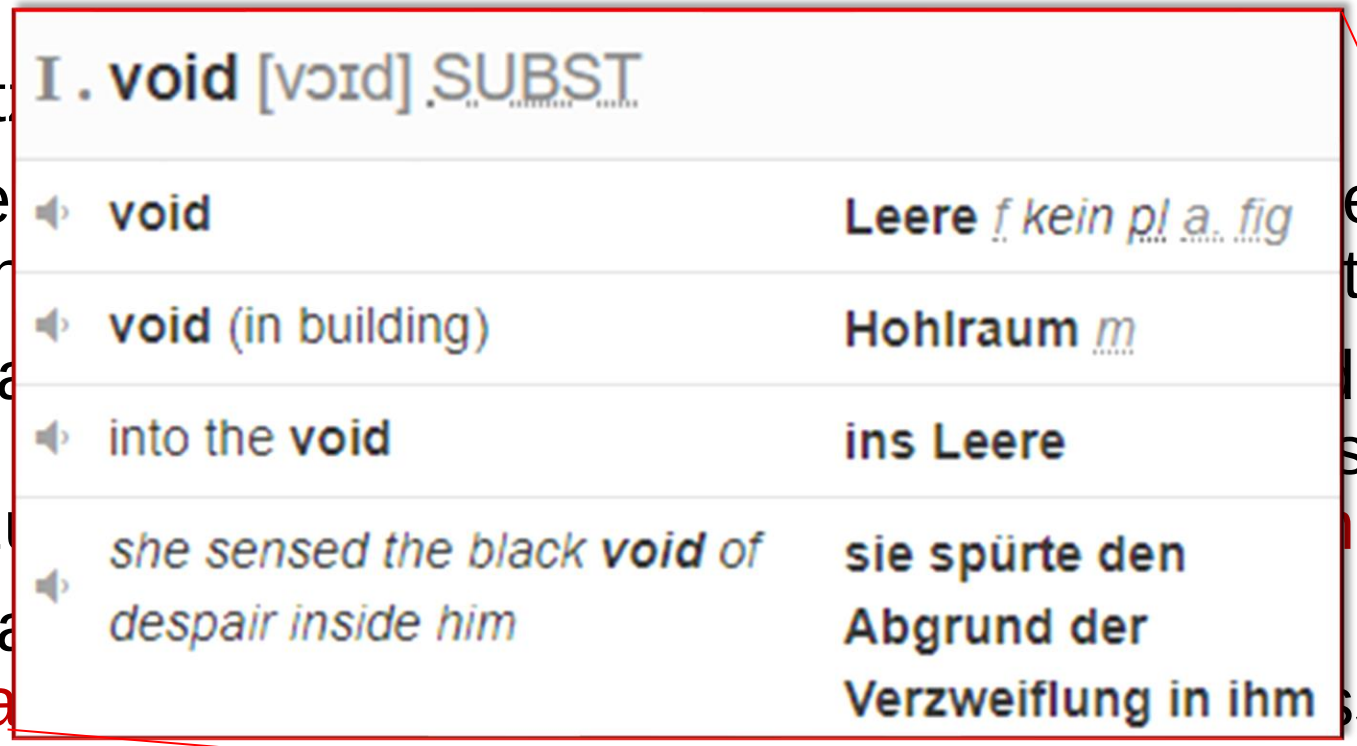
```
public static void main(String args[]) {}
```

Das erste Programm

Benutzung von Methoden

- Einer Methode können beim Aufruf Werte übergeben werden und die Methode kann einen Wert an die aufrufende Methode zurückgeben:
- Die aufzurufende Methode legt die Anzahl und den Typ der Variablen fest, die mit der aufrufenden Methode ausgetauscht werden. Die aufzurufende Methode stellt die formalen Parameter bereit.
- Die aufrufende Methode liefert die sogenannten aktuellen Parameter, die zu den formalen Parametern passen müssen.
- Wenn eine Methode keinen Wert liefert, wird dies mit **`void`** gekennzeichnet.

Das erste Programm



I . **void** [vɔɪd] SUBST	
void	**Leere** *f kein pl a. fig*
void (in building)	**Hohlraum** *m*
into the **void**	**ins Leere**
she sensed the black ***void*** *of despair inside him*	**sie spürte den Abgrund der Verzweiflung in ihm**

Benut

- Eine eben werden und die Meth thode zurückgeben:
- Die a len Typ der Variablen fest, scht werden. Die aufz eter bereit.
- Die a aktuellen Para ssen müssen.
- Wenn eine Methode keinen Wert liefert, wird dies mit **`void`** gekennzeichnet.

Aussehen von Programmen

Grundsätzlich sollen Programme ästhetisch aussehen:

- Auch schöne Programme können Fehler haben
- Aber bei hässlichen Programmen ist die Korrektur sehr viel schwieriger.

Formatieren ist einfach:

- IDE: Kontextmenü „Format“
- Leerzeilen bitte „nach Gefühl“ einfügen/löschen

Namensregeln

- Java-Klassen sind in einer hierarchischen Ordnung strukturiert
- Als Paketname dient oft die HFT-Mailadresse „rückwärts“
  - Der Bindestrich wird zu _
  - Die führenden Ziffern werden entfernt
  - Aus 42hepe1bif@hft-stuttgart.de wird dadurch de.hft_stuttgart.hepe1bif

Einführung

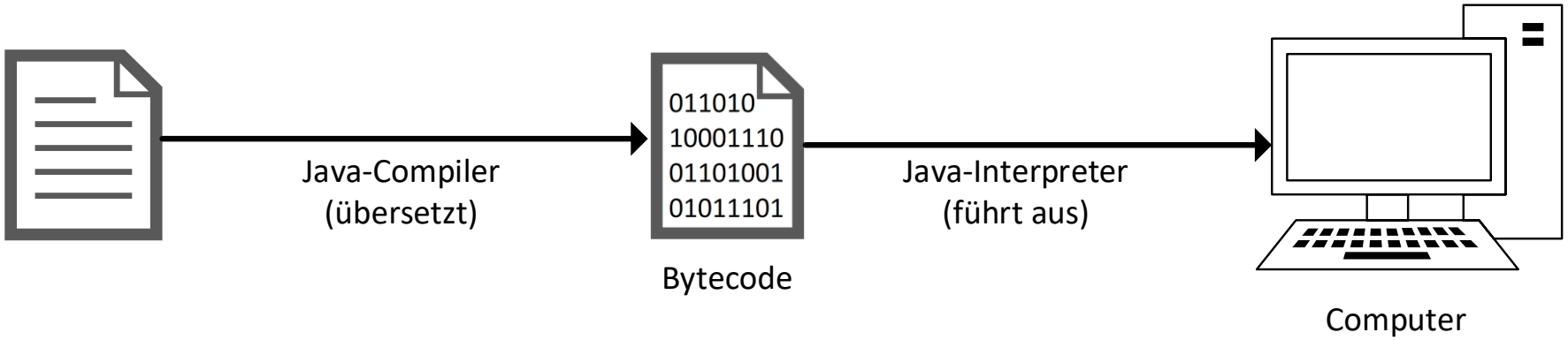
- Programmieren
- Java
- Elemente eines Programms
- Übersetzen und Ausführen eines Programms

Java-Programme ausführen

- Java-Programme können mit einem beliebigen Editor geschrieben werden. Die Dateiendung der Quell-Dateien lautet **`.java`**. Hier steht der Quellcode (engl. Source!)
- Der Java-Compiler **`javac`** übersetzt die Quelldatei in sogenannten Bytecode, dieser wird in der mit der Endung **`.class`** gespeichert.
- Die Java-Laufzeitumgebung (engl. Runtime Environment) **`java`** führt den Bytecode aus.
- Das erste Java-Programm gibt den Text „Hello World“ am Bildschirm aus.

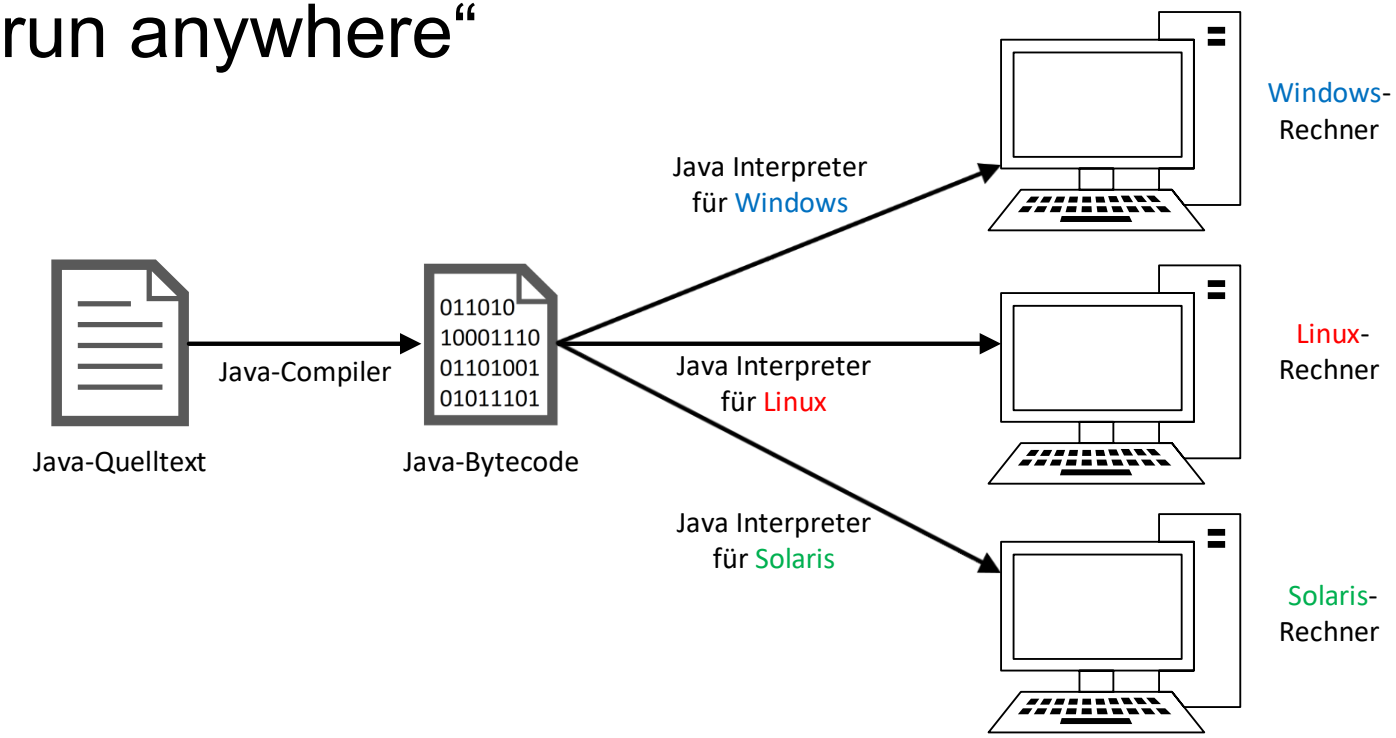
Java-Programme ausführen

- Übersetzen: In Java wird Programmcode über einen Compiler in den Bytecode übersetzt (kompiliert). Dieser kann dann von einem Interpreter ausgeführt werden.



Java-Programme ausführen

- Eine Besonderheit von Java ist, dass ein Bytecode von verschiedenen Systemen verwendet werden kann: „Write once, run anywhere“



Agenda

1) Organisation

2) Rechnerraum

3) Einführung

4) Übung

Arbeiten mit der Kommandozeile

Öffnen der Kommandozeile unter Windows mit cmd, Konsole oder Eingabeaufforderung

Befehle ausprobieren wie z.B.

- whoami
- dir (directory, was befindet sich in meinem Verzeichnis)
- cd … (change directory), z.B. cd Desktop
- mkdir … (make directory), z.B. mkdir Test
- java –version
- exit
- …

Übungsblatt

- Programmieren ohne IDE
- Programmieren mit Eclipse
- Fleißaufgaben

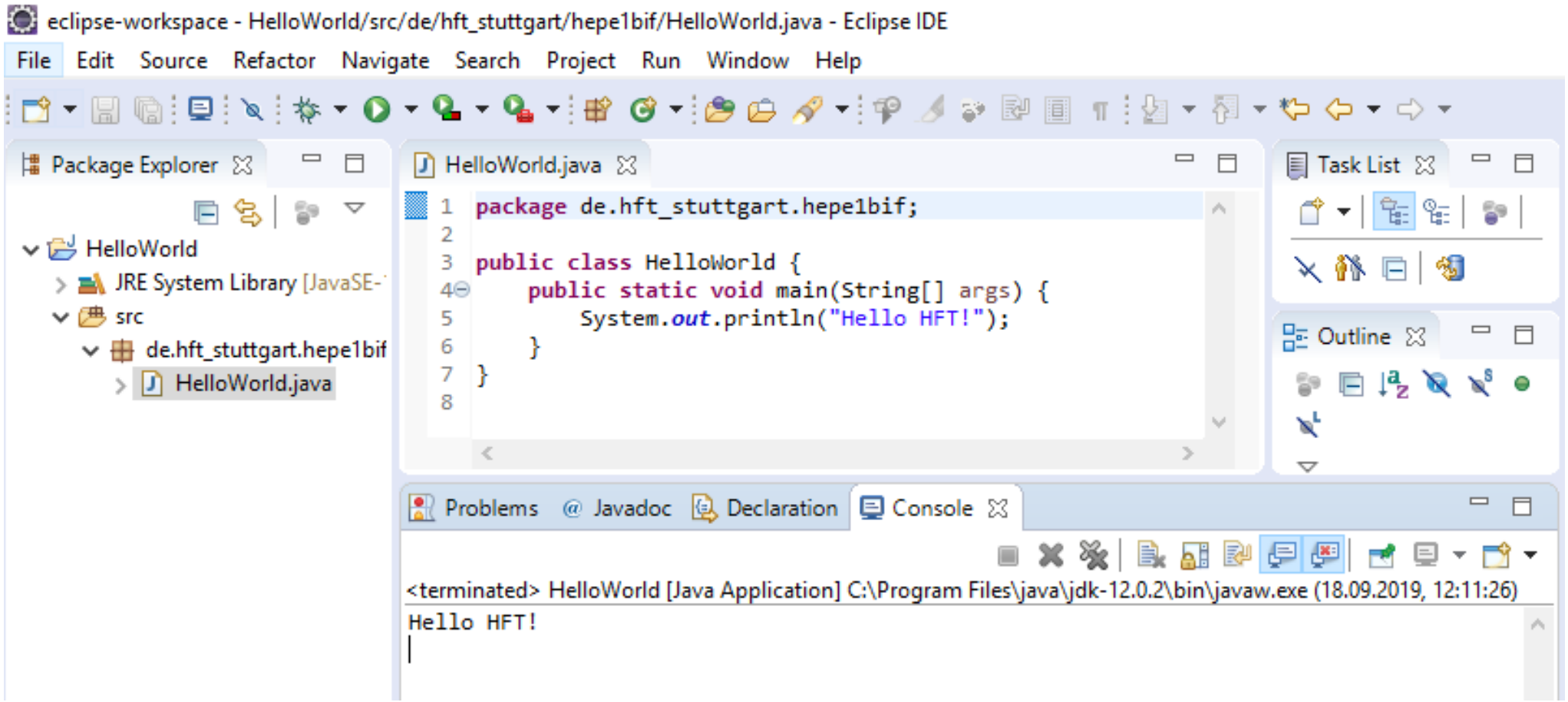
Programmieren ohne IDE (ohne Paket)

- Erstellen Sie mit Hilfe eines einfachen Texteditors ein Programm HalloStudent.java, welches Ihren Namen auf Konsole ausgibt.
- Übersetzen Sie das Programm mit
  javac HalloStudent.java
- führen Sie es mit
  java HalloStudent
  aus.
- Suchen Sie den erzeugten Binärcode

Übungsblatt

- Programmieren ohne IDE
- Programmieren mit Eclipse
- Fleißaufgaben

Kommandozeile und IDE benutzen



Aufgabe mit Eclipse

- Wir machen nun die gleiche Aufgabe mit Eclipse!

- Schreiben Sie ein Programm, das Ihre Schuhgröße ausgibt. Initialisieren Sie hier die Schuhgröße als Variable.

- Probieren Sie die Operatoren aus und lassen Sie sich die Ergebnisse ausgeben.

Hausaufgaben

- Lernpartner suchen
- Eclipse und Java zu Hause installieren
- Wiederholen der Folien und Aufgaben
  - Offene Fragen notieren
- Mit Vorkenntnissen: Weitere Aufgaben bearbeiten: https://speiser.hft-pages.io/programmieraufgaben/2022-ws-pro-1/uebung-00/index.html
  - Zahlensumme
  - Münzen auf Schachbrett
  - Pizza Rezept

Fragen