

Programmierkurs

Vorlesung 00

M.Sc. Laslo Hunhold

Department Mathematik/Informatik
Abteilung Informatik
Universität zu Köln

4. November 2020



Organisatorisches

Allgemeines

Webseite

- ▶ <https://hunhold.cs.uni-koeln.de/programmierkurs/>
- ▶ Informationen, Vorlesungsaufzeichnungen und -folien, Übungen
- ▶ Regelmäßiger Aufruf wird erwartet

Kontakt

- ▶ hunhold@cs.uni-koeln.de
- ▶ Bei fachlichen Fragen: Vor Kontaktaufnahme den Bereich „Häufige Fragen/Probleme“ auf der Webseite lesen
- ▶ Persönliches Gespräch nach Absprache möglich

Skript

- ▶ Kein Skript, aber Folien
- ▶ Empfehlung: Notizen mit Vorlesungs- & Foliennummern (z.B. 00/3), Videoaufzeichnungen beliebig oft abrufbar
- ▶ Abschnitt „Errata“ auf der Webseite

Credit Points

6 CP

- ▶ Wirtschaftsinformatik
- ▶ Alle Vorlesungen und Übungen prüfungsrelevant

3 CP

- ▶ Wirtschaftsmathematik, Nebenfach Informatik, Studium Integrale, Schülerstudenten
- ▶ Vorlesungen und Übungen prüfungsrelevant nur bis einschließlich Vorlesung bzw. Übungsblatt 09 (veröffentlicht am 18. Januar 2021)
- ▶ Freiwillige Weitererteilnahme empfohlen

Vorlesung

- ▶ Veröffentlichung von nun an: **Montags um 08:00 Uhr**
- ▶ Fragen zur Vorlesung per E-Mail (siehe aber auch zuvor „Häufige Fragen/Probleme“)
- ▶ 12 Vorlesungen bis 1. Februar 2021
- ▶ Entfall am 28. Dezember und 4. Januar (Weihnachtsferien)

Klausur

Form

- ▶ Keine Zulassung nötig
- ▶ Keine eKlausur, ohne Hilfsmittel
- ▶ 90 Minuten (3CP und 6CP)
- ▶ Genaue Form wird noch bekanntgegeben (Online oder Präsenz)

Termine

- ▶ Klausur: Montag, 22. Februar 2021, 12:00-13:30 Uhr
- ▶ Nachklausur: Montag, 22. März 2021, 09:00-10:30 Uhr
- ▶ Räume werden, bei Präsenzklausur, rechtzeitig bekanntgegeben

Anmeldung

- ▶ Ab ca. 4 Wochen vorher
- ▶ Per KLIPS (Wirtschaftsinformatik, Wirtschaftsmathematik, Nebenfach Informatik)
- ▶ Per Formular (Studium Integrale, Schülerstudenten)
- ▶ Prüfungsamt kontaktieren bei Rückfragen zur Anmeldung
- ▶ Abmeldung bis 1 Woche vorher

Übungen

Übungsblätter

- ▶ Veröffentlichung: **Montags um 08:00 Uhr** (wie Vorlesung)
- ▶ Besprechung in der darauffolgenden Woche
- ▶ Übungsblatt 00: Wird heute veröffentlicht, Präsenzblatt für nächste Woche; Installation von OpenJDK 11/Eclipse vorab strengstens empfohlen

Abgaben

- ▶ Keine Abgaben; freiwillige Bearbeitung
- ▶ Freiwillige Einsendung von (auch fehlerhaften) Lösungen an den Übungsleiter für Rückmeldungen
- ▶ Einsendefrist: Eine Woche später, also **Montags um 08:00 Uhr** in der Besprechungswoche

Übungsgruppen

- ▶ Besprechung des abgegebenen Übungsblattes via Zoom
- ▶ Diskussion eingesendeter (auch fehlerhafter) Lösungen (freiwillig)
- ▶ Beginn: Nächste Woche
- ▶ Übungsleiter: Felix Behrmann, Marvin Pogoda, Simon Wolf

Übungsgruppenanmeldung

Allgemeines

- ▶ Formular auf Veranstaltungsw Webseite
- ▶ Heute (4. November 2020) geöffnet um 14:00 Uhr, schließt am Freitag (6. November 2020) um 12:00 Uhr
- ▶ Keine Matrikelnummer (Schüler, etc.)? Eine unbesetzte Zahl 0***** wählen

Übungsgruppentermine

Gruppe 0	Montags	12:00 Uhr	Simon Wolf
Gruppe 1	Dienstags	12:00 Uhr	Felix Behrmann
Gruppe 2	Dienstags	14:00 Uhr	Simon Wolf
Gruppe 3	Mittwochs	12:00 Uhr	Felix Behrmann
Gruppe 4	Donnerstags	12:00 Uhr	Marvin Pogoda
Gruppe 5	Freitags	12:00 Uhr	Marvin Pogoda

Angabe Präferenz (2 Angabemöglichkeiten)

1. Präferenzliste: Liste von Zahlen 0-5 (mindestens 3), absteigende Präferenz, kommasetrennt, keine Leerzeichen (Beispiel: 1,4,3,0)
2. „Anker“-Matrikelnummer (Beispiel: 7922552) für Arbeitsgruppen

Übungsgruppenzuteilung

- ▶ Zuteilung bis Freitag (6. November 2020) um 20:00 Uhr
- ▶ Bekanntmachung im Loginbereich des Übungsservers
- ▶ Gruppenwechsel nur mit Tauschpartner

Einführung

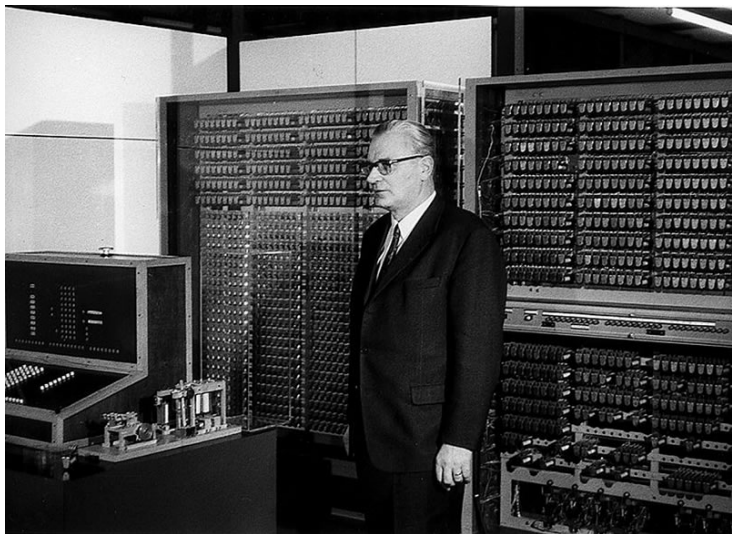


Abbildung: Konrad ZUSE (1910-1995) mit dem Z3 (1941) (Quelle: Stadtarchiv Hünfeld)

Was ist Informatik?

Informatik: „Informatik ist die Wissenschaft von der systematischen Verarbeitung und Speicherung von Informationen, besonders der automatischen Verarbeitung mit Hilfe von Computern.“, Gesellschaft für Informatik e.V.

- ▶ Theorie: Analyse von Algorithmen, Datenstrukturen, etc. (siehe Informatik I+II)
- ▶ Anwendung: Abstraktion, Programmierung, etc. (Programmierkurs)

Daten (Input) → Computer → Daten (Output)

Programmierung: „Einstellung“ des Computers auf die gewünschte Verarbeitung von (Eingabe-)Daten.

Programmierablauf

1. Problemerkfassung (Eingabedaten/Gewünschte Ausgabedaten)

	Jan/€	...	Dez/€
Internet	59.99	...	26.85
Kraftstoff	181.74	...	110.22

Tabelle: Finanzübersicht 2019

Gegeben: Tabelle „Finanzübersicht 2019“

Gesucht: Gesamtausgaben für Kraftstoff 2019

2. Entwurf eines Lösungsverfahrens („Algorithmus“)
Summiere in Zeile „Kraftstoff“ von Spalte „Jan“ bis „Dez“
3. Realisierung des Lösungsverfahrens („Implementierung“)

```
summe = 0.0;  
for (i in 'Jan'..'Dez') {  
    summe += tabelle['Kraftstoff'][i];  
}  
output(summe);
```
4. Testen

Übersetzung/Interpretierung

Wir möchten unseren Quellcode „...for (i in 'Jan'..'Dez')...“ auf dem Computer ausführen (= in Anweisungen umwandeln).

Verschiedene Vorgehensweisen (je nach Programmiersprache; unterstrichen ist das, was am Ende ausgeführt wird):

Direktübersetzung (z.B. C, C++, Rust):

- ▶ Quellcode $\xrightarrow{\text{Compiler}}$ Maschinencode = Anweisungen
- ▶ Am schnellsten, aber spezifisch für Betriebssystem/Architektur

Interpretierung (z.B. Python, JavaScript, PHP):

- ▶ Quellcode $\xrightarrow{\text{Interpreter}}$ Anweisungen
- ▶ Unspezifisch, aber Interpretierung langsam

Bytecodeübersetzung (z.B. Java, WebAssembly, C#):

- ▶ Quellcode $\xrightarrow{\text{Compiler}}$ Bytecode $\xrightarrow{\text{Virtuelle Maschine}}$ Anweisungen
- ▶ Unspezifisch und schneller als Interpretierung

Java-Programmierung

OpenJDK 11 (Java Development Kit), enthält u.a.

- ▶ *Compiler* (javac), wandelt .java-Dateien (Quellcode) in .class-Dateien (Bytecode) um
- ▶ *Java Runtime Environment (JRE)* (java), führt Bytecode aus (Virtuelle Maschine), gibt es auch eigenständig
- ▶ *Java Archiver* (jar), zum Zusammenpacken von .class-Dateien

Eclipse Java IDE (Integrated Development Environment) (optional)

- ▶ Entwicklungsumgebung für Java
- ▶ Editor und graphische Benutzeroberfläche für das JDK

„Hallo Welt“ in Java

```
1 public class HalloWelt {
2     public static void main (String[] args) {
3         System.out.println("Hallo Welt!");
4     }
5 }
```

"HalloWelt.java"

- ▶ **public**: Zugriffsmodifikator (später)
- ▶ **class HalloWelt**: Klassendeklaration „HalloWelt“ (Dateiname=Klassenname.java, Klassenname wird groß geschrieben)
- ▶ **public, static**: (später)
- ▶ **void**: Kein Rückgabewert
- ▶ **main(String args[])**: main-Methode (Einstiegspunkt), muß genau einmal im Programm vorkommen
- ▶ **System.out.println()**: Anweisung gibt Argument in der Konsole aus
- ▶ Anweisungen enden immer mit einem Semikolon

Struktur eines Java-Programms

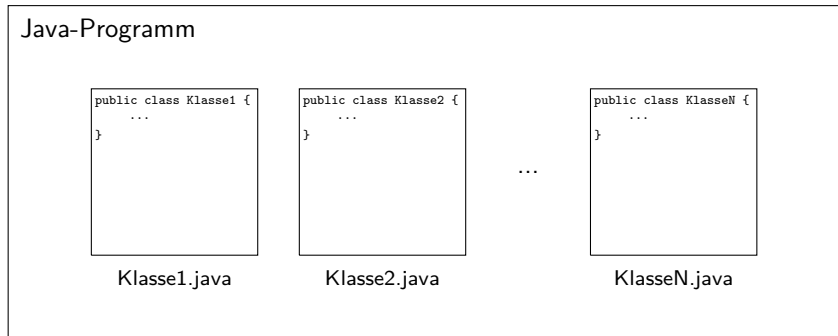


Abbildung: Jedes Java-Programm besteht aus einer oder mehreren Klassen.

Zusammenfassung

- ▶ Begriff der Informatik
- ▶ Programmierablauf
- ▶ Programmübersetzung (Compiler, Interpreter, Quellcode, Maschinencode, Bytecode)
- ▶ Java-„Hallo Welt“-Programm und -Programmstruktur

Literatur

Für Anfänger

- ▶ Hans-Peter Habelitz: *Programmieren lernen mit Java: Der leichte Einstieg für Programmieranfänger*. Rheinwerk Computing, 6. aktualisierte Auflage, 2020 (geringer Preis)
- ▶ Kai Günster: *Einführung in Java*. Rheinwerk Computing, 3. aktualisierte Auflage, 2020 (geringer Preis)
- ▶ Christian Ullenboom: *Java ist auch eine Insel*. Rheinwerk Computing, 15. aktualisierte und überarbeitete Auflage, 2020

Für Fortgeschrittene

- ▶ Michael Inden: *Der Weg zum Java-Profi*. dpunkt.verlag, 5. überarbeitete und aktualisierte Auflage, 2020 (großer Tiefgang)
- ▶ Kathy Sierra , Bert Bates: *Java von Kopf bis Fuß*. O'Reilly Verlag, 1. Auflage, 2006 (Klassisch)

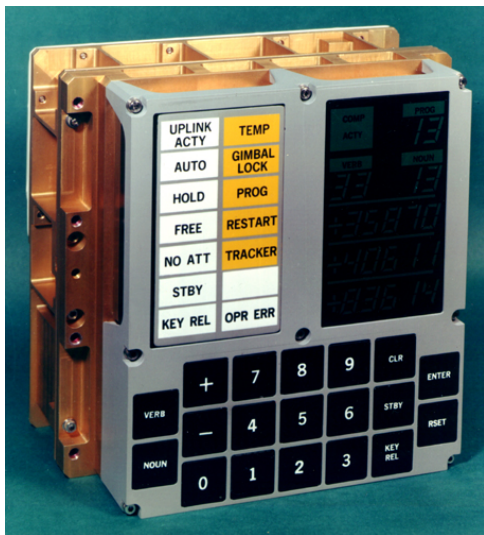


Abbildung: Apollo Guidance Computer (1966) (Quelle: NASA Photo Archive)