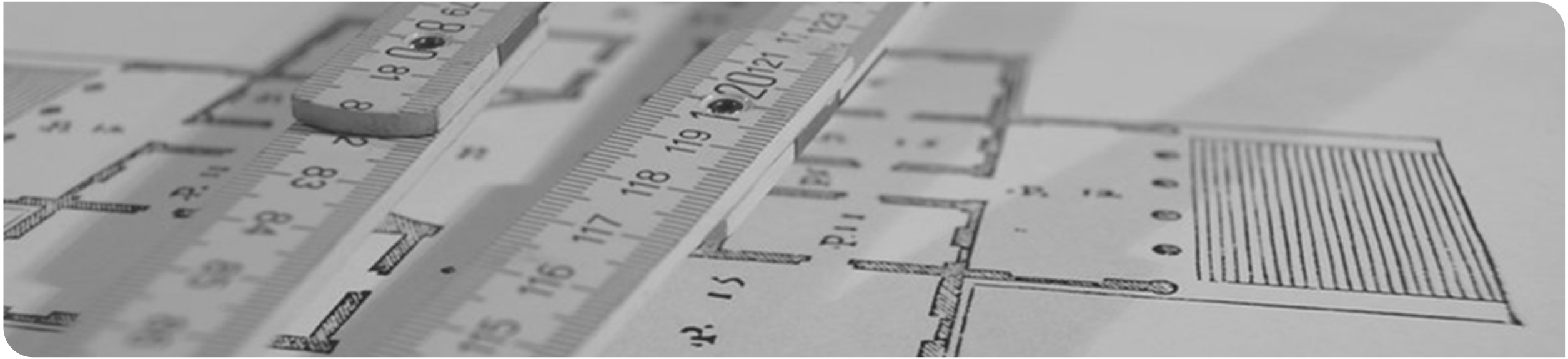
Karlsruher Institut für Technologie

Vorlesung Programmieren

Einführung

Prof. Dr.-Ing. Anne Koziolek



Karlsruher Institut für Technologie

Fragen?

Beitreten über

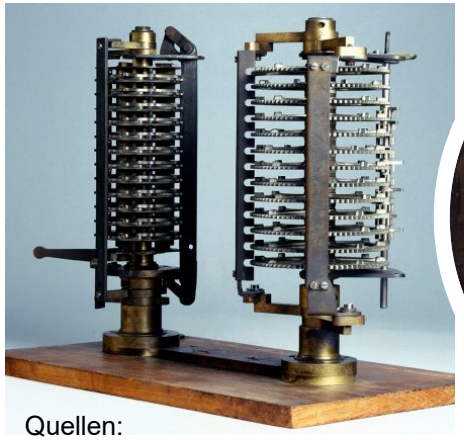
slido.com

#Prog

Was ist Programmieren?

Anfänge der Programmierung

- 1840er Jahre: **Ada Lovelace** beschreibt Lösungsverfahren für ein mathematisches Problem auf einem mechanischen Rechenapparat
  - Erstes „Programm“
  - Ada Lovelace gilt als erste Programmiererin



Quellen:
de.wikipedia.org/wiki/Analytical_Engine, de.wikipedia.org/wiki/Ada_Lovelace

- 1940er Jahre: **Konrad Zuse** entwickelt erste höhere Programmiersprache für Zuse Z3
  - „Plankalkül“
  - Abstrahiert von Maschinensprache

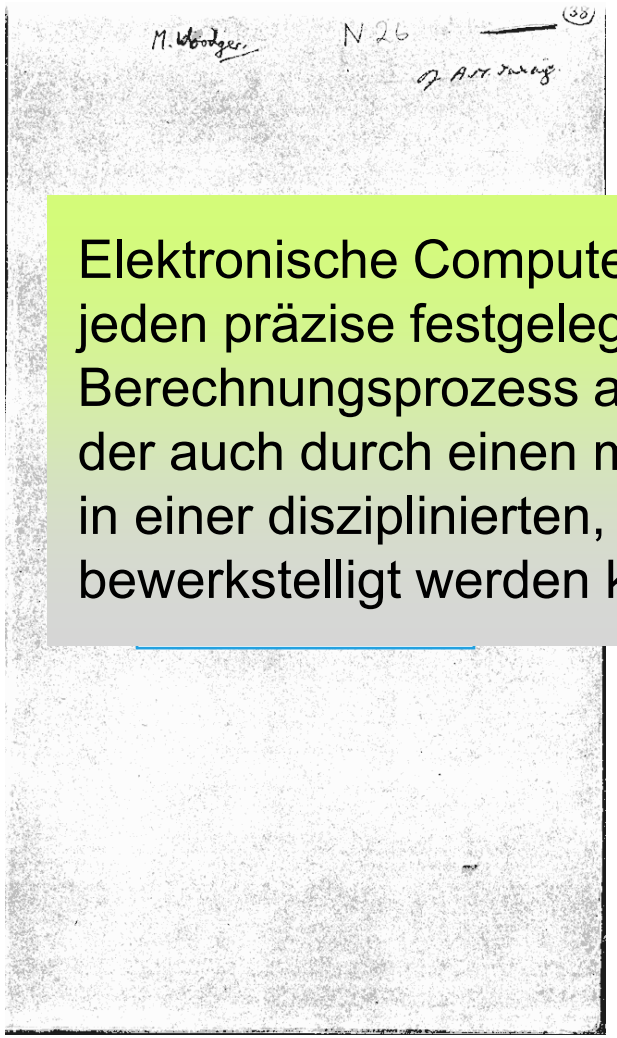
Rückblick: Ferranti Mark I (1951)

Alan Turing
(1912-1954)

Science Museum, Creative Commons BY-NC-ND

Ferranti Mark I Manual

Alan Turing
(1912-1954)



1. General remarks on electronic computers.

Electronic computers are intended to carry out any definite rule of thumb process which could have been done by a human operator working in a disciplined but unintelligent manner. The electronic computer should however obtain its results very much more quickly. The human computer with whom we are comparing it may be imagined as

conversely. If the analogy of the human computer is to be maintained these parts would correspond to his ears and voice, by means of which he communicates with his employer.

2. Scales of notation.

The information stored on paper by the human computer will mostly consist of sequences of digits drawn from 0, 1, ..., 9. There may also be other symbols such as decimal points, spaces etc. and there may be occasional remarks in English, Greek letters etc. There may in fact be anything from 10 to 100 different symbols used, and there is no particular need to decide in advance how many different symbols will be concerned. With an electronic computer however such a decision has to be made; the number of symbols chosen is ruled very

Elektronische Computer sind dazu gedacht,
jeden präzise festgelegten
Berechnungsprozess auszuführen,
der auch durch einen menschlichen Operator
in einer disziplinierten, aber stupiden Weise
bewerkstelligt werden könnte.

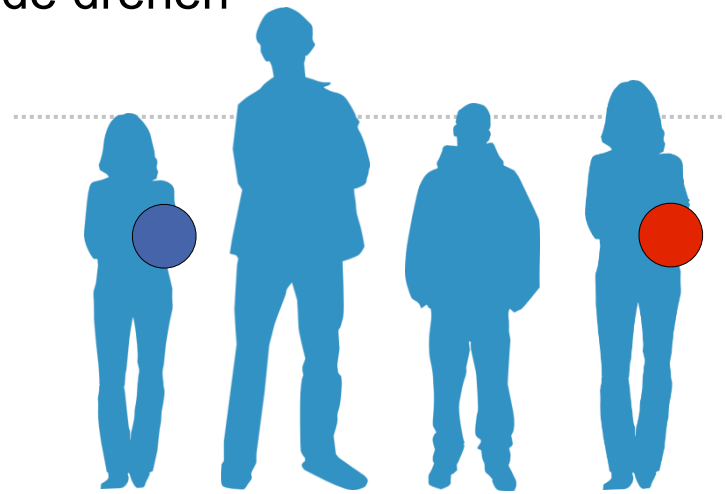
Programmierung

Wie kann nun ein Computer dazu gebracht werden, jeden präzise festgelegten Berechnungsprozess auszuführen?

Durch Programmierung!

Jetzt sind Sie dran!

- **Aufgabe:** Aufstellung der Personen der Größe nach
- **Verfügbare Instruktionen:**
  1.  nach links oder rechts weitergeben → mit einer Hand zeigen
  2.  nach links oder rechts weitergeben → mit zwei Händen zeigen
  3. Personen, die  oder  halten, wechseln den Platz → Hände drehen



War das nun „Programmieren“?

- Zum Teil **ja**:
  - Präzise festgelegter Prozess
  - Eindeutige Instruktionen
- Aber **nicht**:
  - „... in stupider Weise bewerkstelligt ...“
- Es fehlt:
  - Ein „allgemeiner Plan“, der angibt, welcher Schritt als nächstes zu tun ist
  - *Allgemein*: für beliebige Anzahl von Personen
  - Möglichst abstrakt, d.h. (z.B.) nicht nur auf Personen bezogen
- Dies wäre dann ein **Programm**!

Beispiel: Plan für das Sortieren von Personen

- Gegeben sei eine Anzahl n von Personen. Jede Person hat eine Größe.

- 1. Gib die blaue Karte der ersten Person und die rote Karte der zweiten Person
- 2. Wenn die linke Person mit einer Karte größer als die rechte Person mit einer Karte, wechsle.
- 3. Gib die blaue Karte nach rechts weiter
- 4. Gib die rote Karte nach rechts weiter
- 5. Wenn noch keine Karte ganz rechts angekommen ist, gehe zu Schritt 2
- 6. Wenn die Personen noch nicht sortiert sind, gehe zu Schritt 1

Beispiel: Allgemeiner Plan für das Sortieren

- Gegeben sei eine Anzahl n von *Elementen*. Jedes *Element* hat eine zu sortierende *Eigenschaft*.
- 1. Gib die blaue Karte dem ersten *Element* und die rote Karte dem zweiten *Element*
- 2. Wenn die *Eigenschaft des linken Elements* mit Karte der beiden größer als die *Eigenschaft des rechten Elements* mit Karte, wechsle.
- 3. Gib die blaue Karte nach rechts weiter
- 4. Gib die rote Karte nach rechts weiter
- 5. Wenn noch keine Karte ganz rechts angekommen ist, gehe zu Schritt 2
- 6. Wenn die *Elemente* noch nicht sortiert sind, gehe zu Schritt 1

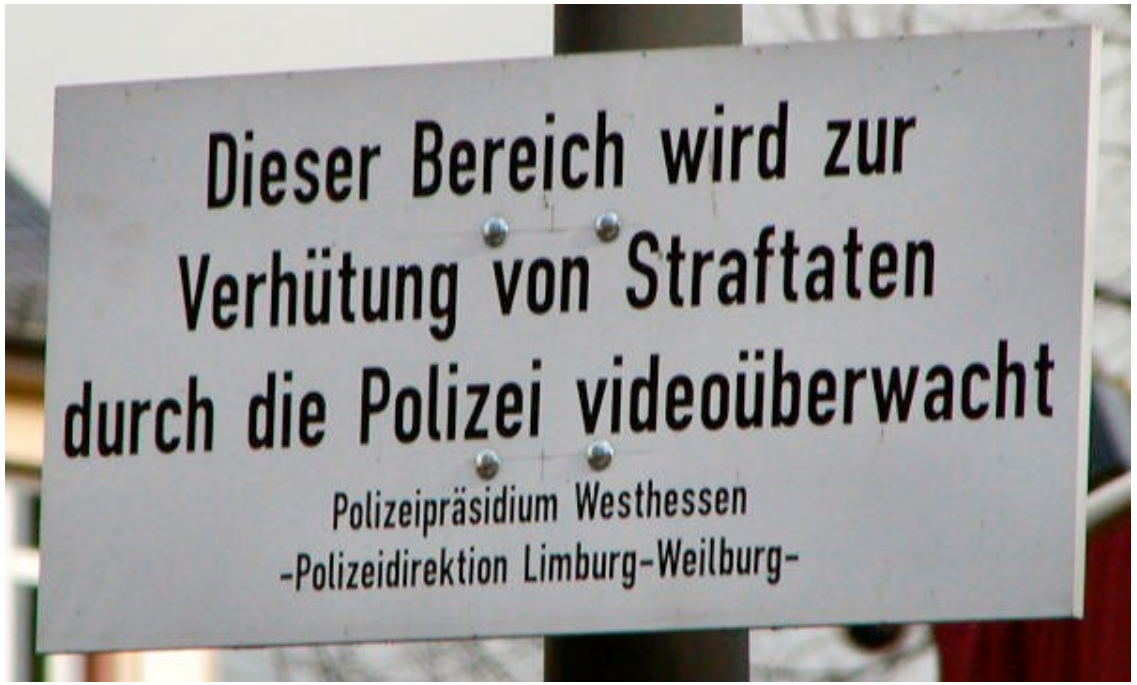
→ Dieses Vorgehen nennt man auch „Bubblesort“

Unsere Programmiersprache: Java

- **Java Entwurfsziele:**
  - Einfach, objekt-orientiert und „vertraut“ (familiar) (für Programmierer)
  - Robust und sicher
  - Architekturneutral und portabel
  - Performant
  - Nebenläufig und dynamisch

- **Warum Java als Programmiersprache?**
  - Warum nicht Maschinensprache?
  - Warum nicht Umgangssprache?

Warum nicht Umgangssprache?



mehrdeutig!

Was Sie in dieser Vorlesung lernen

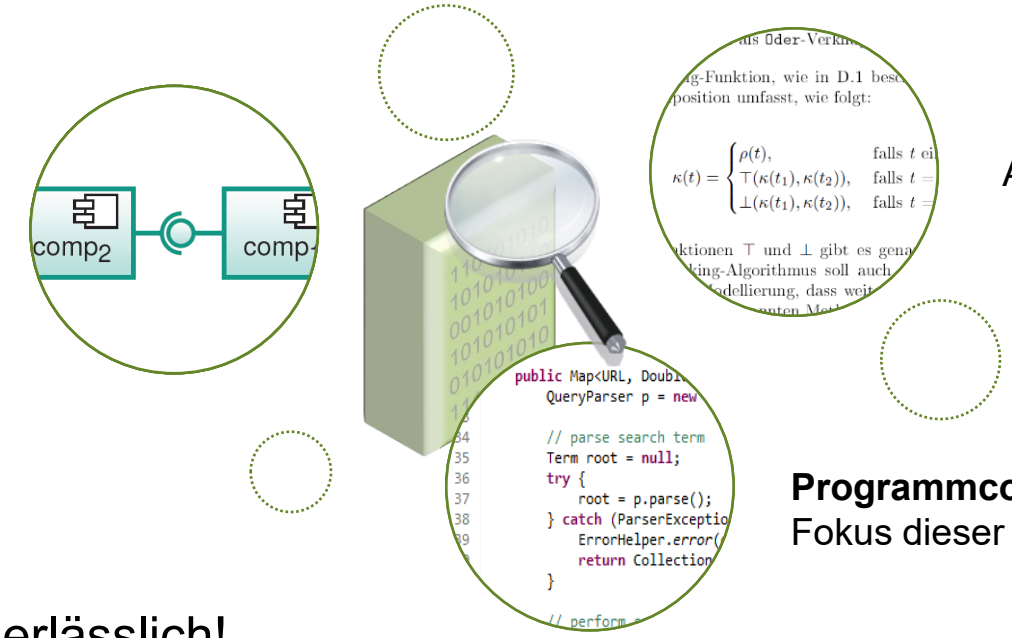
- Entwicklung von Verfahren (Algorithmen) zur Lösung einfacher Probleme
- Problem-Modellierung in einer Programmiersprache
- Fähigkeit zur Abstraktion

- Sauber zu programmieren!
  - Lesbare, verständliche, leicht wartbare Programme
  - Nachvollziehbare Modellierung
  - Gute Strukturierung, gute Dokumentation

Rolle der Programmierung

- Für die Softwaretechnik
  - Programmcode nur eine von vielen Sichten auf ein Softwaresystem
  - Weitere zentrale Sichten...

Software-Architektur



Anforderungen

Programmcode
Fokus dieser Vorlesung

- Für das weitere Studium
  - Grundlegende Programmierfähigkeiten unerlässlich!
  - Fördert strukturiertes und algorithmisches Denken

Kann man heute nicht alles mit KI generieren?

- Im Kleinen ja:
  - Kleine Programme mit bekannten Algorithmen
  - Auch mittelgroße Programme, aber hier macht die KI schon Fehler!
- Im Großen nicht:
  - Um Fehler zu finden, muss man Code verstehen können
    - Unsere Hypothese: Um Code verstehen zu können, muss man Code schreiben können
  - Um große Systeme bauen und weiterentwickeln zu können, muss man an den richtigen Stellen den richtigen Programmcode einfügen können
  - Um Anforderungen einer Domäne in sinnvolle Lösungen zu übersetzen, muss man die Programmierung beherrschen
  - Im Studium ist Programmierung wichtige Grundlage für das Verständnis fundamentaler Prinzipien
- Daher: Lernziel hier ist, selber Programme zu schreiben
  - Mehr dazu am Mittwoch
  - Siehe auch https://www.informatik.kit.edu/faq-wiki/doku.php?id=generative_ki

Was kann ich mit guten Programmierkenntnissen anfangen?

- Mitarbeit in Open-Source-Projekten
- Mitarbeit an interessanten Projekten in einem Unternehmen
- Selbstständig machen
- Entwicklung von Apps für Smartphones
- Gesellschaftliche Probleme lösen (Digitalisierung)
  - Neue Mobilitätskonzepte
  - Energieversorgung
  - Fabrik der Zukunft
  - Neue medizinische Geräte
  - …

Wohin geht die Reise?

- Alan Turing über den Ferranti Mark I Computer:

> *This is only a foretaste of what is to come (...).*
>
> *(...) I do not see why it should not enter any of the fields normally covered by the human intellect and eventually compete on equal terms.*

Was Sie diese Woche tun müssen:

- Bei **YouSubscribe** zum Tutorium anmelden **ab 22.10.2024 18:00 Uhr bis 24.10.2024, 18:00 Uhr**
  https://www.informatik.kit.edu/tutorieneinteilung/

- Während des gesamten Semesters: Prüfen Sie regelmäßig Ihren **KIT-E-Mail-Account uxxxx@student.kit.edu**

- **Organisatorische Fragen?**
  1. FAQ auf Vorlesungshomepage prüfen
  2. Diskussionsforum prüfen, ob Frage schon gestellt
  3. Im Diskussionsforum fragen (Beitragsbenennungskonvention einhalten!)
  4. Tutorin oder Tutor fragen
  5. E-Mail an programmieren-vorlesung@cs.kit.edu