

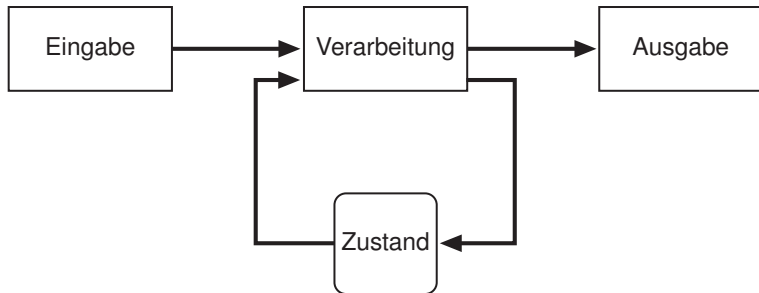
Einführung in die Objektorientierte
Programmierung
Vorlesung 3: Einführung in die Programmierung

Sebastian Küpper

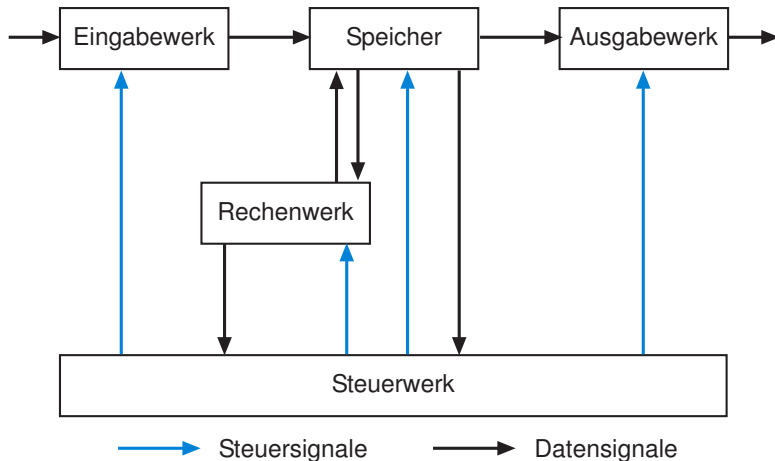
Wozu Programmiersprachen?

- Binäre Codierung von grundlegenden Operationen für Menschen schwer lesbar
- Natürliche Sprache für Computer unverständlich und zu ambig
- Darum: Erst lesbare Befehle um Grundoperationen zu formulieren (Assembler)
- Dann: Zusammenfassen oft genutzter Funktionalität aus nicht atomaren Grundoperationen, insbes. Funktionsaufruf (imperative Sprachen)
- Schließlich: Modellorientierte Strukturbefehle, die auf bestimmte Denkweisen optimiert sind (Objektorientierung, Funktionale Programmierung, Logikprogrammierung)

Klassisches Programmiermodell: EVA – Eingabe Verarbeitung Ausgabe



Von Neumann-Rechnerarchitektur



Zustand: Speicher

Verarbeitung: Rechenwerk, Steuerwerk

Eingabe: Eingabewerk

Ausgabe: Ausgabewerk

Fragestellung der OOP

Reine sequentielle Befehlsfolgen sind ein unzulängliches Modell um komplexe Software mit Aspekten wie

- Sensoren
- Nutzern
- Kommunikation über das Netzwerk
- multimedialer Ausgabe

zu realisieren.

Fragestellung der OOP

Reine sequentielle Befehlsfolgen sind ein unzulängliches Modell um komplexe Software mit Aspekten wie

- Sensoren
- Nutzern
- Kommunikation über das Netzwerk
- multimedialer Ausgabe

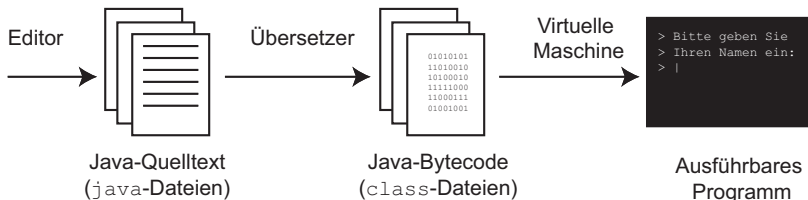
zu realisieren.

- Welche Einheiten werden gebraucht?
- Welche Rolle spielen sie im Verbund mit anderen?
- Welche Funktionen bieten sie an, welche benötigen sie?
- Wie muss ihre innere Arbeitsweise gestaltet werden?
- Wie arbeiten sie zusammen?

Wieso Java?

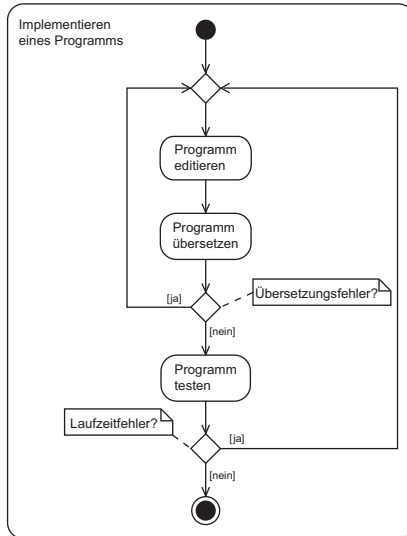
- Hohe Verbreitung: Neben C/C++ und Python die wichtigste Programmiersprache in der Wirtschaft
- Viele Konzepte der Programmierung sichtbar umgesetzt: Typisierung, OOP inkl. Geheimnisprinzip, Modularisierung
- Plattformunabhängig
- Breite Anwendung: Sowohl Endanwendersoftware als auch Webservices
- Syntaktische Ähnlichkeit zu C/C++ und C#, aber mit weniger Stolpersteinen als C/C++
- Kostenfrei
- Umfangreiche Bibliothek von Klassen, incl. facettenreicher API

Java-Übersetzungsprozess



VM garantiert Plattform-Unabhängigkeit. Implementationen verfügbar z. B. für Windows, Mac OS und Linux

Ablauf der Implementation



Besonderheiten

- (öffentliche) Klassen müssen in Dateien stehen, die gleich heißen; z. B. die Klasse `schildkroete` in der Datei `schildkroete.java`
- Es muss eine `main`-Methode in einer Klasse geben, die der Einstiegspunkt für das Programm ist.

Fragen zur Vorlesungseinheit

- 1 Erläutern Sie die Schanierfunktion moderner Programmiersprachen zwischen Maschinencode und natürlicher Sprache.
- 2 Wie ist nach dem von Neumann-Rechnermodell ein Computer aufgebaut?
- 3 Welchem Zweck dient die Virtuelle Maschine?