01 Einführung

Begriffe

Begriff	Definition
Programmiersprache	Formal konstruierte Sprache, entworfen um Befehle an Maschinen (speziell Computer) zu übermitteln. Gemeinsame Sprache zwischen Mensch und Maschine. (Eine Programmiersprache muss **eindeutig** und **Turing-vollständig** sein) $\rightarrow$ alle Sprachen sind gleichmächtig, haben aber eigene Schwächen und Stärken untereinander
Syntax	Gibt das Muster (die formale Struktur) vor, nach dem Programme einer Sprache aufgebaut sind (Zusammenfügungsregeln).
Semantik	Definiert die Bedeutung von Programmen (Anweisungen, Operatoren, usw.) (Interpreta- tionsregeln).
Programmierparadigma	Fundamentaler Programmierstil. Eine bestimmte Art die Struktur und Ele- mente von Programmen aufzubauen. Man kann mit allen Programmiersprachen die gleichen Probleme lösen (sind alle Turing-Vollständig), aber nicht gleich elegant.
Algorithmus	Schrittweises Verfahren zur Lösung von einem Problem oder einer Klasse von Problemen.

Wie werden Programmierparadigmen eingeordnet?

- **imperative** Programmierparadigmen
- **deklarative** Programmierparadigmen

Sind diese zwei Paradigmen eindeutig voneinander auseinanderzuhalten?

- Nicht zwingend

- Deklarative Sprachen haben meist auch imperative Eigenschaften
- Eher komplementär als sich gegenseitig ausschliessend

Was sind imperative Programmiersprachen?

- Sprachen, die beschreiben **wie** ein Problem gelöst werden kann
- Quellcode gibt an, was der Computer in welcher Reihenfolge tun soll
- Zur Steuerung der Befehlsausführung stehen Kontrollstrukturen (Sequenz, Schleife, Verzweigung) zur Verfügung
- Eng angelehnt an Ausführung von Maschinen-Code (Assembler) auf Computern $\rightarrow$ Von-Neumann-Architektur

Was sind deklarative Programmiersprachen?

- Sprachen, die beschreiben **was** das Problem ist (ohne Kontrollfluss)
- Zu lösende Problem wird beschrieben
- Lösungsweg wird dann automatisch ermittelt
- Programm enthält genügend Information, sodass das Problem gelöst werden kann

Welche Unterkategorien gibt es von imperativen Programmiersprachen?

- strukturiert (Pascal)
- prozedural (C)
- objektorientiert (Java)

```c
// In C
int a = 25;                 // Variablendeklaration
int b = 15;                 // Variablendeklaration

start:                      // Sprungmarke
if (a > b) a = a - b;       // Auswahl
else b = b - a;             // Alternative
if (a != b) goto start;     // Bedingung (goto!!)

printf("result = %i", a);   // Konsolenausgabe
```

Welche Unterkategorien gibt es von deklarativen Programmiersprachen?

- funktional (Scheme)
- logisch (Prolog)

Strukturierte Programmierung (imperativ)

- Beschränkung auf drei Kontrollstrukturen
- Sequenzen, Auswahl (Verzweigung), Wiederholung (Schleife)
- *C, Fortran, Pascal*

```
int a = 25;                 // Variablendeklaration
int b = 15;                 // Variablendeklaration

while (a != b) {            // Schleife
    if (a > b) a = a - b;   // Auswahl
    else b = b - a;         // Alternative
}

printf("result = %i", a);   // Konsolenausgabe
```

Prozedurale Programmierung (imperativ)

- Unterteilung von Programmen in Teilprogramme (Prozedur, Funktion)
- Code-Duplikationen verhindern
- Oft Gegenstück zu OOP
- *C, Fortran, Pascal*

```
int main() {                        // Hauptfunktion
    int result = gcd(25, 15);       // Funktionsaufruf
    printf("result = %i", result);  // Konsolenausgabe
    return 0;                       // Funktionsrückgabe
}

int gcd(int a, int b) {             // Funktionskopf
    while (a != b) {                // Schleife
        if(a > b) a = a - b;        // Auswahl
        else b = b -a;              // Alternative
    }
    return a;                       // Funktionsrückgabe
}
```

Objektorientierte Programmierung (imperativ)

- Laufende Programme bestehen aus einzelnen Objekte, welche miteinander interagieren
- Objekte sind typischerweise Instanzen von Klassen
- Klassen definieren Zustand (Variablen) und Verhalten (Methoden)
- *Smalltalk, Objective C, C++, Java, C#*

Logische Programmierung (deklarativ)

- Programm besteht aus Fakten und Regeln
- Aus welchen auf Anfrage automatisch versucht wird, eine Lösungsaussage herzuleiten
- Basiert auf mathematischer Logik
- *Prolog*

Funktionale Programmierung (deklarativ)

- Programme bestehen ausschliesslich aus Definitionen von Funktionen mit Parametern und Rückga- bewerten
- Rückgabewert hängt ausschliesslich von den Parametern ab
- keinen Zustand und keine veränderbaren Daten
- Basiert auf Lambda-Kalkül
- *Clojure, Erlang, Haskell, Lisp, ML, Scheme*