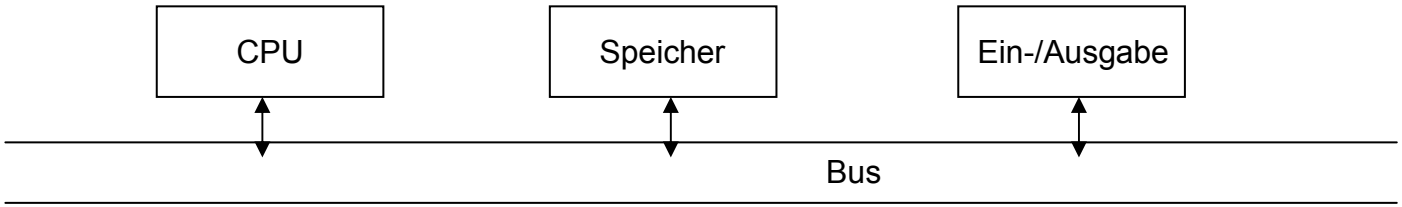
Imperative Sprachen

- Grundidee:
  - Programm lineare Folge von Befehlen
  - Abarbeitung in definierter Reihenfolge

- Grundlage: von-Neumann-Rechner
  - in Worte gegliederter Speicher
  - Wort: Programmbefehl oder Datum
  - Ablauf: Wort aus Speicher holen, als Befehl ausführen oder Manipulation daran vornehmen (Operationsprinzip SISD)
  - → Programm schreibt vor, wie Speicher sukzessive modifiziert wird



Imperative Sprachen

- Erweiterungen der einfachen von-Neumann-Maschine:
  - Typzuordnung (bisher typlose Worte)
  - aus mehreren Worten zusammengesetzte Datenelemente
  - aus primitiven Operationen zusammengesetzte komplexe Operationen

Imperative Sprachen

charakteristisch für imperative Sprachen:

- Variablen und Typen
- Blöcke
- bel. komplexe Datenstrukturen
- Kontrollstrukturen

- strukturiertes Programmieren

Imperative Sprachen

Variablen:

- speichern Daten
- Abstraktion einer Speicherzelle
- Wertänderung im Programmlauf erlaubt
  -- nur durch Grundidee möglich

Beispiel:

f(g(x),h(x))

g(x) könnte x-Wert ändern → Argumente von f müssen in fester Reihenfolge nacheinander ausgeführt werden, nicht parallel (in Gegensatz zu rein funktionalen Sprachen)