CoMa I

E g

- Computer und Algorithmen
- Programmiersprachen
- Algorithmen versus Programmiersprachen
- Literaturhinweise

- Computer und Algorithmen
- Programmiersprachen
- Algorithmen versus Programmiersprachen
- Literaturhinweise

Zeitalter der Computerrevolution

Zeitalter der Computerrevolution

- Auswirkungen auf Gesellschafts- und Sozialordnung wie die Industrielle Revolution

Zeitalter der Computerrevolution

- Auswirkungen auf Gesellschafts- und Sozialordnung wie die Industrielle Revolution
  - Industrielle Revolution
    → Steigerung der körperlichen Kräfte

Zeitalter der Computerrevolution

- Auswirkungen auf Gesellschafts- und Sozialordnung wie die Industrielle Revolution
  - Industrielle Revolution
    → Steigerung der körperlichen Kräfte
  - Computerrevolution
    → Verstärkung des menschlichen Gehirns

Zeitalter der Computerrevolution

- Auswirkungen auf Gesellschafts- und Sozialordnung wie die Industrielle Revolution
  - Industrielle Revolution
    → Steigerung der körperlichen Kräfte
  - Computerrevolution
    → Verstärkung des menschlichen Gehirns
- Informatik entsteht als neue Disziplin, behandelt alle Aspekte des Computereinsatzes und der Rechnerentwicklung

Was macht Computer so revolutionär?

- Geschwindigkeit
  - Selbst komplexe Algorithmen schnell ausgeführt
- Zuverlässigkeit
  - kaum technische Fehler, eher Programmierfehler
- Speicher, Datenbanken
  - riesige Informationsmengen, schneller Zugriff
- Kosten
  - niedrig im Vergleich zu menschlicher Arbeit
- Vernetzung
  - Internet, vernetzte Anwendungen (Flugbuchung)

Versuch einer Definition

Versuch einer Definition

- Computer ist eine Maschine

Versuch einer Definition

- Computer ist eine Maschine
  - die geistige Routinearbeiten durchführt

Versuch einer Definition

- Computer ist eine Maschine
  - die geistige Routinearbeiten durchführt
  - indem sie einfache Operationen (Basisoperationen) mit hoher Geschwindigkeit ausführt

Versuch einer Definition

- Computer ist eine Maschine
  - die geistige Routinearbeiten durchführt
  - indem sie einfache Operationen (Basisoperationen) mit hoher Geschwindigkeit ausführt
  - Beispiele:

Versuch einer Definition

- Computer ist eine Maschine
  - die geistige Routinearbeiten durchführt
  - indem sie einfache Operationen (Basisoperationen) mit hoher Geschwindigkeit ausführt
  - Beispiele:
    - Suchen eines Namens in einer Liste

Versuch einer Definition

- Computer ist eine Maschine
  - die geistige Routinearbeiten durchführt
  - indem sie einfache Operationen (Basisoperationen) mit hoher Geschwindigkeit ausführt
  - Beispiele:
    - Suchen eines Namens in einer Liste
    - Überweisungen von Gehältern in einer Firma

Versuch einer Definition

- Computer ist eine Maschine
  - die geistige Routinearbeiten durchführt
  - indem sie einfache Operationen (Basisoperationen) mit hoher Geschwindigkeit ausführt
  - Beispiele:
    - Suchen eines Namens in einer Liste
    - Überweisungen von Gehältern in einer Firma
- Konsequenzen dieser Definition

Versuch einer Definition

- Computer ist eine Maschine
  - die geistige Routinearbeiten durchführt
  - indem sie einfache Operationen (Basisoperationen) mit hoher Geschwindigkeit ausführt
  - Beispiele:
    - Suchen eines Namens in einer Liste
    - Überweisungen von Gehältern in einer Firma
- Konsequenzen dieser Definition
  - Arbeit muss in Basisoperationen zerlegbar sein

Versuch einer Definition

- Computer ist eine Maschine
  - die geistige Routinearbeiten durchführt
  - indem sie einfache Operationen (Basisoperationen) mit hoher Geschwindigkeit ausführt
  - Beispiele:
    - Suchen eines Namens in einer Liste
    - Überweisungen von Gehältern in einer Firma
- Konsequenzen dieser Definition
  - Arbeit muss in Basisoperationen zerlegbar sein
  - man muss dem Computer diese mitteilen können

Versuch einer Definition

- Computer ist eine Maschine
  - die geistige Routinearbeiten durchführt
  - indem sie einfache Operationen (Basisoperationen) mit hoher Geschwindigkeit ausführt
  - Beispiele:
    - Suchen eines Namens in einer Liste
    - Überweisungen von Gehältern in einer Firma
- Konsequenzen dieser Definition
  - Arbeit muss in Basisoperationen zerlegbar sein
  - man muss dem Computer diese mitteilen können
- Eine solche Beschreibung einer Aufgabe für den Computer mit Basisschritten nennt man Algorithmus

Algorithmus

Algorithmus

- Ein Algorithmus ist eine präzise, endliche Beschreibung eines allgemeinen Verfahrens unter Verwendung ausführbarer elementarer Verarbeitungsschritte zur Lösung einer gestellten Aufgabe

Algorithmus

- Ein Algorithmus ist eine präzise, endliche Beschreibung eines allgemeinen Verfahrens unter Verwendung ausführbarer elementarer Verarbeitungsschritte zur Lösung einer gestellten Aufgabe

- präzise: in einer festgelegten Sprache abgefasst

Algorithmus

- Ein Algorithmus ist eine präzise, endliche Beschreibung eines allgemeinen Verfahrens unter Verwendung ausführbarer elementarer Verarbeitungsschritte zur Lösung einer gestellten Aufgabe

- präzise: in einer festgelegten Sprache abgefasst
- endlich: nur endlich viel „Text" in der Sprache

Algorithmus

- Ein Algorithmus ist eine präzise, endliche Beschreibung eines allgemeinen Verfahrens unter Verwendung ausführbarer elementarer Verarbeitungsschritte zur Lösung einer gestellten Aufgabe

- präzise: in einer festgelegten Sprache abgefasst
- endlich: nur endlich viel „Text" in der Sprache
- ausführbar: diese muss verstanden werden und ausgeführt werden können

Algorithmus, Prozess, Prozessor

Algorithmus, Prozess, Prozessor

- Prozess: Konkrete Ausführung des Algorithmus

Algorithmus, Prozess, Prozessor

- Prozess: Konkrete Ausführung des Algorithmus
  - besitzt zu jedem Zeitpunkt einen Zustand, der den aktuellen Stand der Ausführung angibt

Algorithmus, Prozess, Prozessor

- Prozess: Konkrete Ausführung des Algorithmus
  - besitzt zu jedem Zeitpunkt einen Zustand, der den aktuellen Stand der Ausführung angibt
- Prozessor: Einheit, die den Prozess ausführt

Algorithmus, Prozess, Prozessor

- Prozess: Konkrete Ausführung des Algorithmus
  - besitzt zu jedem Zeitpunkt einen Zustand, der den aktuellen Stand der Ausführung angibt
- Prozessor: Einheit, die den Prozess ausführt
  - muss kein Computer sein

Beispiele

Beispiele

Beispiele

Prozess

Beispiele

Prozess

Algorithmus

Beispiele

Prozess	Algorithmus	typische Schritte

Beispiele

Prozess	Algorithmus	typische Schritte
Pullover stricken		

Beispiele

Prozess	Algorithmus	typische Schritte
Pullover stricken	Strickmuster	

Beispiele

Prozess	Algorithmus	typische Schritte
Pullover stricken	Strickmuster	2 links, 2 rechts

Beispiele

Prozess	Algorithmus	typische Schritte
Pullover stricken	Strickmuster	2 links, 2 rechts
Modellflugzeug bauen		

Beispiele

Prozess	Algorithmus	typische Schritte
Pullover stricken	Strickmuster	2 links, 2 rechts
Modellflugzeug bauen	Bauanleitung	

Beispiele

Prozess	Algorithmus	typische Schritte
Pullover stricken	Strickmuster	2 links, 2 rechts
Modellflugzeug bauen	Bauanleitung	leime Tragfläche an Rumpf

Beispiele

Prozess	Algorithmus	typische Schritte
Pullover stricken	Strickmuster	2 links, 2 rechts
Modellflugzeug bauen	Bauanleitung	leime Tragfläche an Rumpf
Beethovensonate spielen		

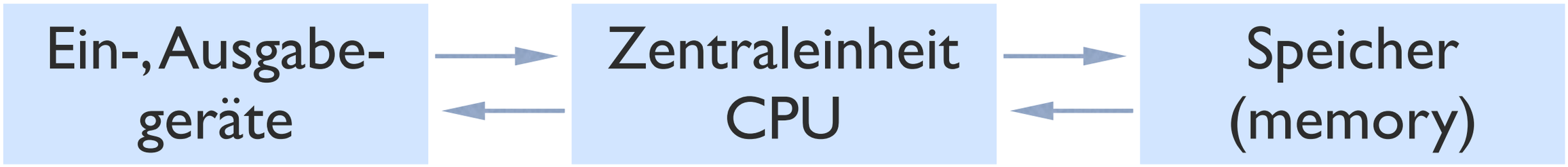
Beispiele

Prozess	Algorithmus	typische Schritte
Pullover stricken	Strickmuster	2 links, 2 rechts
Modellflugzeug bauen	Bauanleitung	leime Tragfläche an Rumpf
Beethovensonate spielen	Notenblatt	

Beispiele

Prozess	Algorithmus	typische Schritte
Pullover stricken	Strickmuster	2 links, 2 rechts
Modellflugzeug bauen	Bauanleitung	leime Tragfläche an Rumpf
Beethovensonate spielen	Notenblatt	einzelne Noten

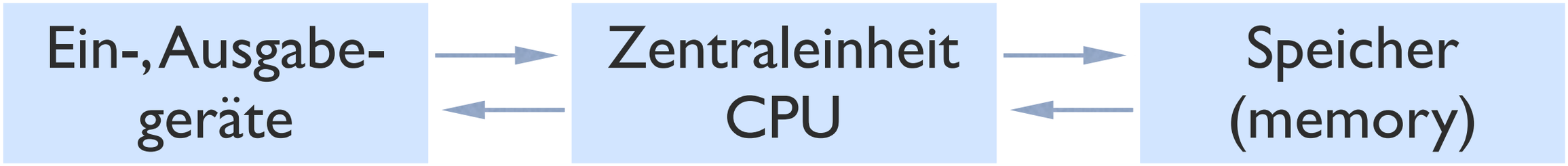
Computer ist spezieller Prozessor



Computer ist spezieller Prozessor

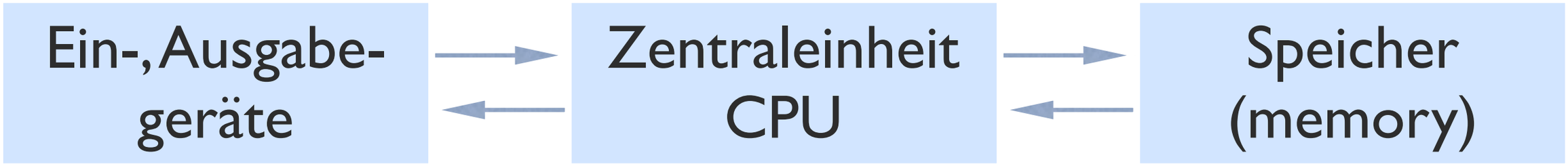
- CPU führt die Basisoperationen aus

Computer ist spezieller Prozessor



- CPU führt die Basisoperationen aus
- Speicher enthält
  - die auszuführenden Operationen des Algorithmus
  - die Information (Daten bzw. Objekte), auf der die Operationen wirken (von-Neumann Rechner)

Computer ist spezieller Prozessor



- CPU führt die Basisoperationen aus
- Speicher enthält
  - die auszuführenden Operationen des Algorithmus
  - die Information (Daten bzw. Objekte), auf der die Operationen wirken (von-Neumann Rechner)
- Ein-Ausgabegeräte
  - bringen Algorithmus und Daten in den Speicher
  - über sie teilt der Computer die Ergebnisse mit

- Computer und Algorithmen
- Programmiersprachen
- Algorithmen versus Programmiersprachen
- Literaturhinweise

Prozessor muss Algorithmus „verstehen”

Prozessor muss Algorithmus „verstehen”

- Prozessor muss verstehen

Prozessor muss Algorithmus „verstehen”

- Prozessor muss verstehen
  - was jeder Schritt bedeutet

Prozessor muss Algorithmus „verstehen”

- Prozessor muss verstehen
  - was jeder Schritt bedeutet
  - die jeweilige Operation ausführen können

Prozessor muss Algorithmus „verstehen”

- Prozessor muss verstehen
  - was jeder Schritt bedeutet
  - die jeweilige Operation ausführen können
- erreichbar durch schrittweise Verfeinerung bis auf das Niveau des Prozessors

Prozessor muss Algorithmus „verstehen”

- Prozessor muss verstehen
  - was jeder Schritt bedeutet
  - die jeweilige Operation ausführen können
- erreichbar durch schrittweise Verfeinerung bis auf das Niveau des Prozessors
  - „2 links-2 rechts” ist Teil der Verfeinerung der Anweisung „Zopfmuster”

Prozessor muss Algorithmus „verstehen”

- Prozessor muss verstehen
  - was jeder Schritt bedeutet
  - die jeweilige Operation ausführen können
- erreichbar durch schrittweise Verfeinerung bis auf das Niveau des Prozessors
  - „2 links-2 rechts” ist Teil der Verfeinerung der Anweisung „Zopfmuster”
- Computer als Prozessor

Prozessor muss Algorithmus „verstehen”

- Prozessor muss verstehen
  - was jeder Schritt bedeutet
  - die jeweilige Operation ausführen können
- erreichbar durch schrittweise Verfeinerung bis auf das Niveau des Prozessors
  - „2 links-2 rechts” ist Teil der Verfeinerung der Anweisung „Zopfmuster”
- Computer als Prozessor
  - Algorithmus in Programmiersprache ausdrücken

Prozessor muss Algorithmus „verstehen”

- Prozessor muss verstehen
  - was jeder Schritt bedeutet
  - die jeweilige Operation ausführen können
- erreichbar durch schrittweise Verfeinerung bis auf das Niveau des Prozessors
  - „2 links-2 rechts” ist Teil der Verfeinerung der Anweisung „Zopfmuster”
- Computer als Prozessor
  - Algorithmus in Programmiersprache ausdrücken
  - Schritt entspricht Anweisung (Befehl, statement)

Programmiersprachen

Programmiersprachen

- einfache Sprachen (Maschinensprachen)

Programmiersprachen

- einfache Sprachen (Maschinensprachen)
  - jede Anweisung kann direkt vom Computer interpretiert werden

Programmiersprachen

- einfache Sprachen (Maschinensprachen)
  - jede Anweisung kann direkt vom Computer interpretiert werden
  - lange Programme für komplexe Aufgaben

Programmiersprachen

- einfache Sprachen (Maschinensprachen)
  - jede Anweisung kann direkt vom Computer interpretiert werden
  - lange Programme für komplexe Aufgaben
  - Programmierung ist also langwierig, mühsam und dadurch fehleranfällig

Programmiersprachen

- einfache Sprachen (Maschinensprachen)
  - jede Anweisung kann direkt vom Computer interpretiert werden
  - lange Programme für komplexe Aufgaben
  - Programmierung ist also langwierig, mühsam und dadurch fehleranfällig
- höhere Programmiersprachen (Java, C++, Pascal, ...)

Programmiersprachen

- einfache Sprachen (Maschinensprachen)
  - jede Anweisung kann direkt vom Computer interpretiert werden
  - lange Programme für komplexe Aufgaben
  - Programmierung ist also langwierig, mühsam und dadurch fehleranfällig
- höhere Programmiersprachen (Java, C++, Pascal, ...)
  - Anweisung deckt ganze Algorithmusteile ab

Programmiersprachen

- einfache Sprachen (Maschinensprachen)
  - jede Anweisung kann direkt vom Computer interpretiert werden
  - lange Programme für komplexe Aufgaben
  - Programmierung ist also langwierig, mühsam und dadurch fehleranfällig
- höhere Programmiersprachen (Java, C++, Pascal, ...)
  - Anweisung deckt ganze Algorithmusteile ab
  - leichtere Erstellung von Programmen

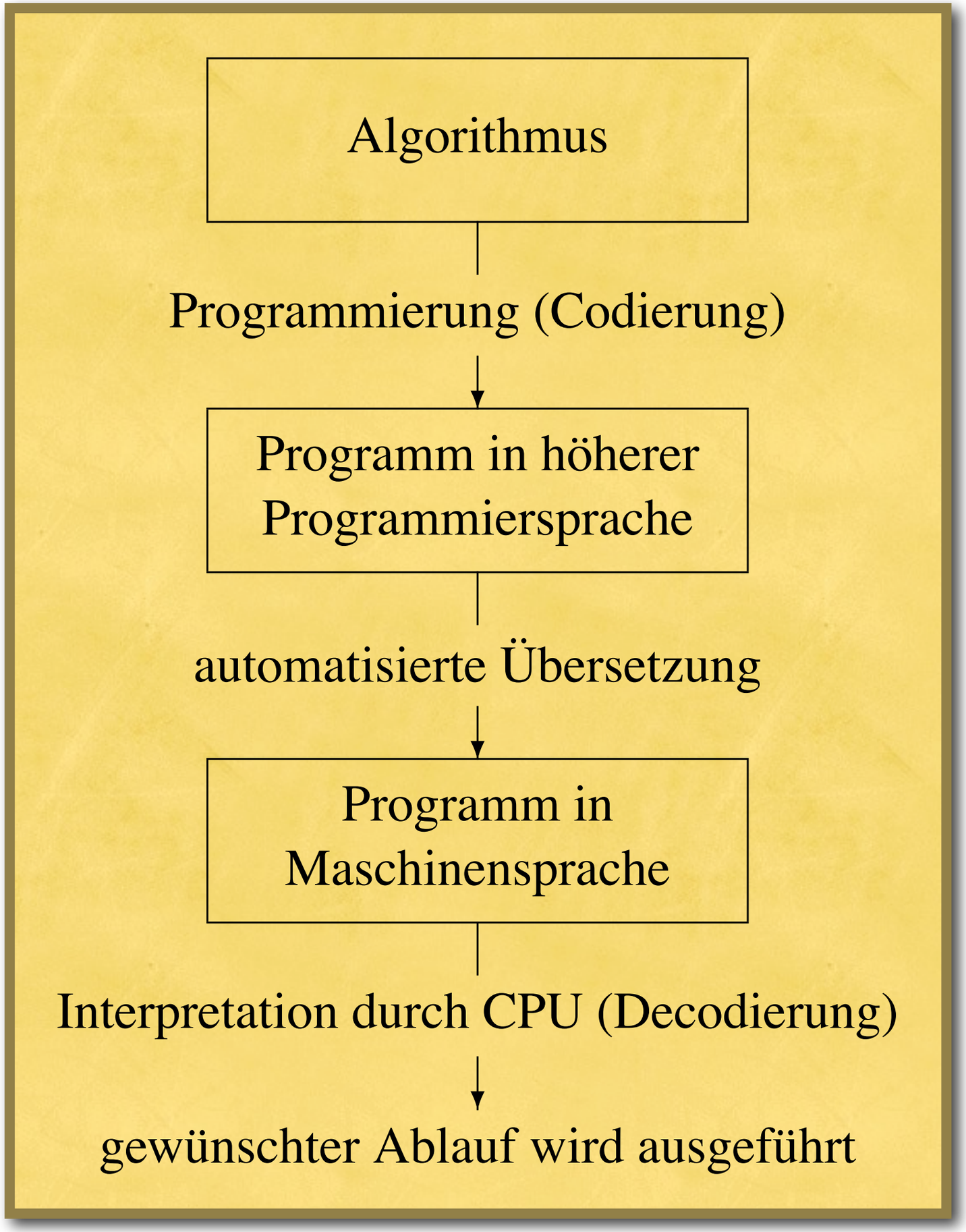
Programmiersprachen

- einfache Sprachen (Maschinensprachen)
  - jede Anweisung kann direkt vom Computer interpretiert werden
  - lange Programme für komplexe Aufgaben
  - Programmierung ist also langwierig, mühsam und dadurch fehleranfällig
- höhere Programmiersprachen (Java, C++, Pascal, ...)
  - Anweisung deckt ganze Algorithmusteile ab
  - leichtere Erstellung von Programmen
  - erfordert Verfeinerung auf Prozessor-Niveau

Programmiersprachen

- einfache Sprachen (Maschinensprachen)
  - jede Anweisung kann direkt vom Computer interpretiert werden
  - lange Programme für komplexe Aufgaben
  - Programmierung ist also langwierig, mühsam und dadurch fehleranfällig
- höhere Programmiersprachen (Java, C++, Pascal, ...)
  - Anweisung deckt ganze Algorithmusteile ab
  - leichtere Erstellung von Programmen
  - erfordert Verfeinerung auf Prozessor-Niveau

automatisiert


Algorithmus
Programmierung (Codierung)
Programm in höherer Programmiersprache
automatisierte Übersetzung
Programm in Maschinensprache
Interpretation durch CPU (Decodierung)
gewünschter Ablauf wird ausgeführt

Hauptthema der Vorlesung

Algorithmus

Programmierung (Codierung)

Programm in höherer Programmiersprache

automatisierte Übersetzung

Programm in Maschinensprache

Interpretation durch CPU (Decodierung)

gewünschter Ablauf wird ausgeführt

Hierarchie bei Programmiersprachen

Hierarchie bei Programmiersprachen

- oberes Ende
  - Java, C++, Pascal, ...

Hierarchie bei Programmiersprachen

- oberes Ende
  - Java, C++, Pascal, ...
- mittleres Niveau
  - C, Fortran
  - haben Konstrukte aus höheren Sprachen und für maschinennahe Programmierung

Hierarchie bei Programmiersprachen

- oberes Ende
  - Java, C++, Pascal, ...
- mittleres Niveau
  - C, Fortran
  - haben Konstrukte aus höheren Sprachen und für maschinennahe Programmierung
- Maschinensprachen
  - Assembler, Mikroprogrammierung

höhere Sprache

```
c = a + b;
```

höhere Sprache

```
c = a + b;
```

Assembler

```
MOVE   R1,a   (hole a aus Speicher und schreibe a in Register R1)
MOVE   R2,b   (hole b aus Speicher und schreibe b in Register R2)
ADD    R2,R1  (addiere Inhalt von R1 zum Inhalt von R2)
MOVE   c,R2   (schreibe Inhalt von R2 unter dem Namen
               c in den Speicher)
```

höhere Sprache

```
c = a + b;
```

Assembler

```
MOVE   R1,a   (hole a aus Speicher und schreibe a in Register R1)
MOVE   R2,b   (hole b aus Speicher und schreibe b in Register R2)
ADD    R2,R1  (addiere Inhalt von R1 zum Inhalt von R2)
MOVE   c,R2   (schreibe Inhalt von R2 unter dem Namen
               c in den Speicher)
```

Mikro-programm-mierung

```
0000  0111  0110  1011  1001  1111  1010  0010  1110  1010
0101  1110  1010  0101  0100  0101  0100  0010  1010  1010
0001  1010  1010  1010  1011  0101  0101  0101  0101  0101
1010  1010  1010  1010  1010  1010  1010  1010  1010  1010
```

Übersetzung in Maschinensprache

Übersetzung in Maschinensprache

- Interpretieren

Übersetzung in Maschinensprache

- Interpretieren
  - jede Anweisung einzeln übersetzt

Übersetzung in Maschinensprache

- Interpretieren
  - jede Anweisung einzeln übersetzt
  - vor der Übersetzung der nächsten Anweisung wird die vorige Anweisung ausgeführt

Übersetzung in Maschinensprache

- Interpretieren
  - jede Anweisung einzeln übersetzt
  - vor der Übersetzung der nächsten Anweisung wird die vorige Anweisung ausgeführt
  - bei jedem Lauf des Programms neu übersetzt

Übersetzung in Maschinensprache

- Interpretieren
  - jede Anweisung einzeln übersetzt
  - vor der Übersetzung der nächsten Anweisung wird die vorige Anweisung ausgeführt
  - bei jedem Lauf des Programms neu übersetzt
- Kompilieren

Übersetzung in Maschinensprache

- Interpretieren
  - jede Anweisung einzeln übersetzt
  - vor der Übersetzung der nächsten Anweisung wird die vorige Anweisung ausgeführt
  - bei jedem Lauf des Programms neu übersetzt
- Kompilieren
  - Programm wird als Ganzes übersetzt durch ein spezielles Programm (Compiler)

Übersetzung in Maschinensprache

- Interpretieren
  - jede Anweisung einzeln übersetzt
  - vor der Übersetzung der nächsten Anweisung wird die vorige Anweisung ausgeführt
  - bei jedem Lauf des Programms neu übersetzt
- Kompilieren
  - Programm wird als Ganzes übersetzt durch ein spezielles Programm (Compiler)
  - die maschinenlesbare Form (object code) steht dann für jeden Aufruf zur Verfügung.

Java wird kompiliert *und* interpretiert

Java wird kompiliert *und* interpretiert

- Programmtext wird vom Java-Compiler in eine Zwischensprache, den Java Bytecode kompiliert

Java wird kompiliert *und* interpretiert

- Programmtext wird vom Java-Compiler in eine Zwischensprache, den Java Bytecode kompiliert
- Dieser wird dann durch die Java Virtual Machine (JVM) interpretiert und ausgeführt

Java wird kompiliert *und* interpretiert

- Programmtext wird vom Java-Compiler in eine Zwischensprache, den Java Bytecode kompiliert
- Dieser wird dann durch die Java Virtual Machine (JVM) interpretiert und ausgeführt
- Vorteil: nur JVM für verschiedene Betriebssysteme (Windows, Linux, Mac OS ...) anpassen

Java wird kompiliert *und* interpretiert

- Programmtext wird vom Java-Compiler in eine Zwischensprache, den Java Bytecode kompiliert
- Dieser wird dann durch die Java Virtual Machine (JVM) interpretiert und ausgeführt
- Vorteil: nur JVM für verschiedene Betriebssysteme (Windows, Linux, Mac OS ...) anpassen
- Der einmal kompilierte Bytecode kann dann auf allen Architekturen ausgeführt werden

Java wird kompiliert *und* interpretiert

- Programmtext wird vom Java-Compiler in eine Zwischensprache, den Java Bytecode kompiliert
- Dieser wird dann durch die Java Virtual Machine (JVM) interpretiert und ausgeführt
- Vorteil: nur JVM für verschiedene Betriebssysteme (Windows, Linux, Mac OS ...) anpassen
- Der einmal kompilierte Bytecode kann dann auf allen Architekturen ausgeführt werden
- Dies macht Java zur WWW-Sprache par excellence und ermöglicht auch die Programmierung von systemübergreifenden Applikationen

Die Software-Hardware Hierarchie

Die Software-Hardware Hierarchie

Anwendungssoftware

z.B. Textverarbeitung,
Statistikpaket, Spiele

Die Software-Hardware Hierarchie

Anwendungssoftware

z.B. Textverarbeitung,
Statistikpaket, Spiele

Systemsoftware

z.B. Betriebssystem, Editor,
Compiler, Eclipse

Die Software-Hardware Hierarchie

Anwendungssoftware

z.B. Textverarbeitung, Statistikpaket, Spiele

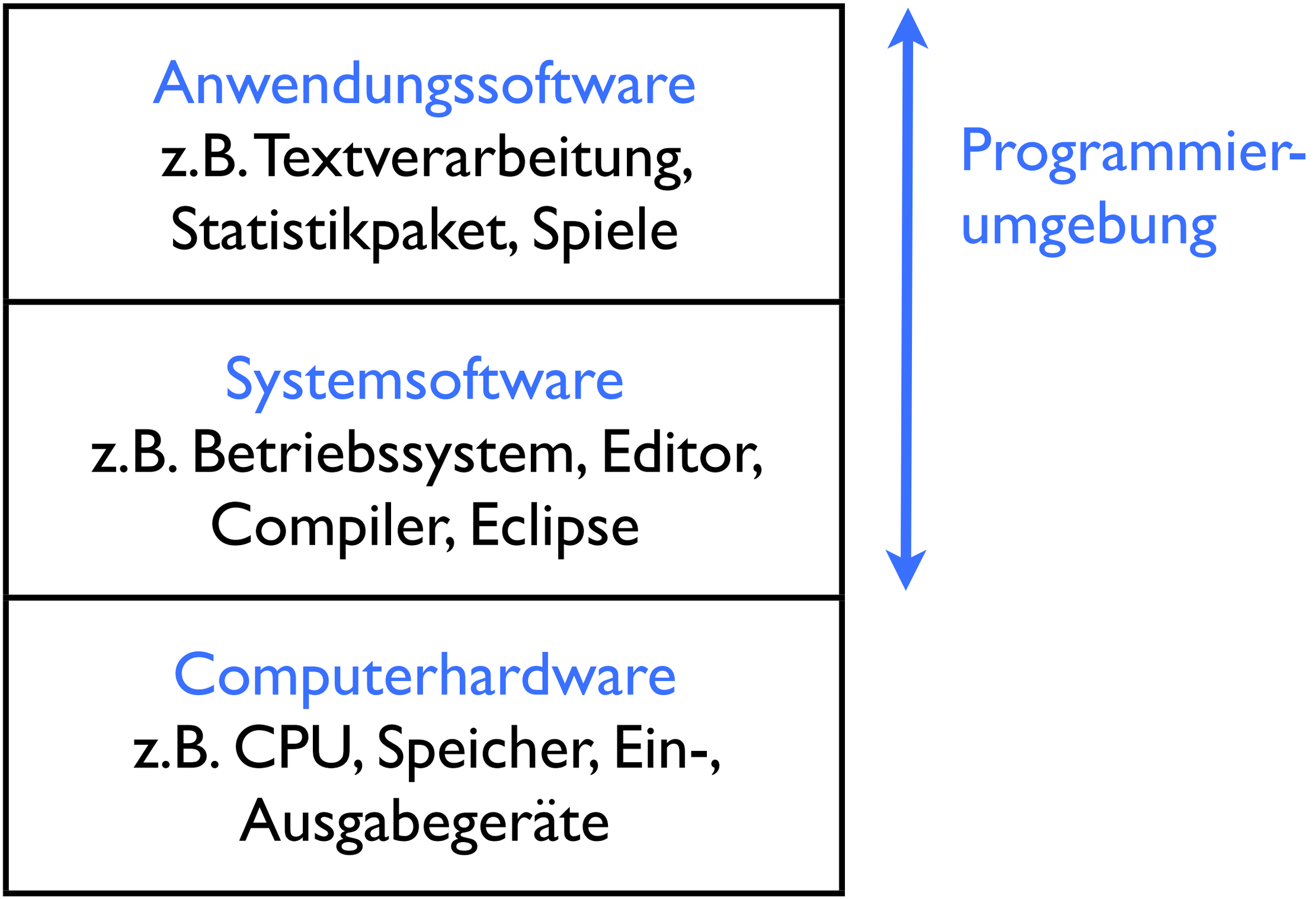
Systemsoftware

z.B. Betriebssystem, Editor, Compiler, Eclipse

Computerhardware

z.B. CPU, Speicher, Ein-, Ausgabegeräte

Die Software-Hardware Hierarchie



- Computer und Algorithmen
- Programmiersprachen
- Algorithmen versus Programmiersprachen
- Literaturhinweise

Durchführung von Prozess auf Computer

Durchführung von Prozess auf Computer

erfordert, dass:

Durchführung von Prozess auf Computer

erfordert, dass:

- ein Algorithmus entworfen wird, der beschreibt, wie der Prozess auszuführen ist,

Durchführung von Prozess auf Computer

erfordert, dass:

- ein Algorithmus entworfen wird, der beschreibt, wie der Prozess auszuführen ist,
- der Algorithmus als Programm in einer geeigneten Programmiersprache ausgedrückt wird,

Durchführung von Prozess auf Computer

erfordert, dass:

- ein Algorithmus entworfen wird, der beschreibt, wie der Prozess auszuführen ist,
- der Algorithmus als Programm in einer geeigneten Programmiersprache ausgedrückt wird,
- der Computer das Programm ausführt (nach automatischer Übersetzung).

Zur Rolle von Algorithmen

Zur Rolle von Algorithmen

- Die Rolle von Algorithmen ist grundlegend. Ohne Algorithmus gibt es kein Programm und ohne Programm gibt es nichts auszuführen

Zur Rolle von Algorithmen

- Die Rolle von Algorithmen ist grundlegend. Ohne Algorithmus gibt es kein Programm und ohne Programm gibt es nichts auszuführen

- Algorithmen sind unabhängig von einer konkreten Programmiersprache und einem konkreten Computertyp (unabhängig von Tagestechnologie), erfordern Modellierung und Mathematisierung der zugrunde liegenden Anwendung

Zur Rolle von Algorithmen

- Die Rolle von Algorithmen ist grundlegend. Ohne Algorithmus gibt es kein Programm und ohne Programm gibt es nichts auszuführen
- Algorithmen sind unabhängig von einer konkreten Programmiersprache und einem konkreten Computertyp (unabhängig von Tagestechnologie), erfordern Modellierung und Mathematisierung der zugrunde liegenden Anwendung
- Programmiersprachen und Computer sind nur Mittel um Algorithmen in Form von Prozessen auszuführen

Algorithmen in Mathematik und Informatik

Algorithmen in Mathematik und Informatik

- Entwurf (Design) von Algorithmen

Algorithmen in Mathematik und Informatik

- Entwurf (Design) von Algorithmen
  - schwierige Tätigkeit, erfordert viel Kreativität und Einsicht (es gibt keinen Algorithmus zum Entwurf von Algorithmen). Hauptgegenstand der CoMa!

Algorithmen in Mathematik und Informatik

- Entwurf (Design) von Algorithmen
  - schwierige Tätigkeit, erfordert viel Kreativität und Einsicht (es gibt keinen Algorithmus zum Entwurf von Algorithmen). Hauptgegenstand der CoMa!
- Berechenbarkeit

Algorithmen in Mathematik und Informatik

- Entwurf (Design) von Algorithmen
  - schwierige Tätigkeit, erfordert viel Kreativität und Einsicht (es gibt keinen Algorithmus zum Entwurf von Algorithmen). Hauptgegenstand der CoMa!
- Berechenbarkeit
  - Gibt es Aufgaben, für die kein Algorithmus existiert?

Algorithmen in Mathematik und Informatik

- Entwurf (Design) von Algorithmen
  - schwierige Tätigkeit, erfordert viel Kreativität und Einsicht (es gibt keinen Algorithmus zum Entwurf von Algorithmen). Hauptgegenstand der CoMa!
- Berechenbarkeit
  - Gibt es Aufgaben, für die kein Algorithmus existiert?
- Komplexität von Algorithmen und Problemen

Algorithmen in Mathematik und Informatik

- Entwurf (Design) von Algorithmen
  - schwierige Tätigkeit, erfordert viel Kreativität und Einsicht (es gibt keinen Algorithmus zum Entwurf von Algorithmen). Hauptgegenstand der CoMa!
- Berechenbarkeit
  - Gibt es Aufgaben, für die kein Algorithmus existiert?
- Komplexität von Algorithmen und Problemen
  - Wie schnell/schwer sind Algorithmen/Probleme?

Algorithmen in Mathematik und Informatik

- Entwurf (Design) von Algorithmen
  - schwierige Tätigkeit, erfordert viel Kreativität und Einsicht (es gibt keinen Algorithmus zum Entwurf von Algorithmen). Hauptgegenstand der CoMa!
- Berechenbarkeit
  - Gibt es Aufgaben, für die kein Algorithmus existiert?
- Komplexität von Algorithmen und Problemen
  - Wie schnell/schwer sind Algorithmen/Probleme?
- Korrektheit von Algorithmen

Algorithmen in Mathematik und Informatik

- Entwurf (Design) von Algorithmen
  - schwierige Tätigkeit, erfordert viel Kreativität und Einsicht (es gibt keinen Algorithmus zum Entwurf von Algorithmen). Hauptgegenstand der CoMa!
- Berechenbarkeit
  - Gibt es Aufgaben, für die kein Algorithmus existiert?
- Komplexität von Algorithmen und Problemen
  - Wie schnell/schwer sind Algorithmen/Probleme?
- Korrektheit von Algorithmen
  - Methoden zum (automatischen) Korrektheitsbeweis

- Computer und Algorithmen
- Programmiersprachen
- Algorithmen versus Programmiersprachen
- Literaturhinweise

- Entwurf und Analyse von Algorithmen:

- Entwurf und Analyse von Algorithmen:
  - Thomas H. Cormen, Charles E. Leiserson, Ronald R. Rivest, and Clifford Stein. Introduction to Algorithms. The MIT Press, Cambridge, MA, second edition, 2001, (auf deutsch bei Oldenbourg, 2005)

- Entwurf und Analyse von Algorithmen:
  - Thomas H. Cormen, Charles E. Leiserson, Ronald R. Rivest, and Clifford Stein. Introduction to Algorithms. The MIT Press, Cambridge, MA, second edition, 2001, (auf deutsch bei Oldenbourg, 2005)
- Java:

- Entwurf und Analyse von Algorithmen:
  - Thomas H. Cormen, Charles E. Leiserson, Ronald R. Rivest, and Clifford Stein. Introduction to Algorithms. The MIT Press, Cambridge, MA, second edition, 2001, (auf deutsch bei Oldenbourg, 2005)
- Java:
  - selber schauen, was eigenem Niveau entspricht

- Entwurf und Analyse von Algorithmen:
  - Thomas H. Cormen, Charles E. Leiserson, Ronald R. Rivest, and Clifford Stein. Introduction to Algorithms. The MIT Press, Cambridge, MA, second edition, 2001, (auf deutsch bei Oldenbourg, 2005)
- Java:
  - selber schauen, was eigenem Niveau entspricht
  - Guido Krüger. Handbuch der Java-Programmierung, 6. Auflage. Addison-Wesley, 2009, 1300 Seiten http://www.javabuch.de/download.html

- Entwurf und Analyse von Algorithmen:
  - Thomas H. Cormen, Charles E. Leiserson, Ronald R. Rivest, and Clifford Stein. Introduction to Algorithms. The MIT Press, Cambridge, MA, second edition, 2001, (auf deutsch bei Oldenbourg, 2005)
- Java:
  - selber schauen, was eigenem Niveau entspricht
  - Guido Krüger. Handbuch der Java-Programmierung, 6. Auflage. Addison-Wesley, 2009, 1300 Seiten http://www.javabuch.de/download.html
- CoMa Skript: http://www.coga.tu-berlin.de/CoMa

Einen guten Start
und viel Erfolg!