

# Coma I

Einleitung



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



- Computer und Algorithmen

- Programmiersprachen

- Algorithmen versus Programmiersprachen

- Literaturhinweise



# 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?

• ???

• ???

• ???

• ???

• ???



# Was macht Computer so revolutionär?

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



# 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
  - Arbeiten müssen in Basisoperationen zerlegbar sein
  - muss dem Computer diese mitteilen können
- Eine solche Beschreibung der Aufgabe für den Computer nennt man **Algorithmus**



# 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

- **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



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



# Computer = 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 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 = **Anweisung** (Befehl, statement)

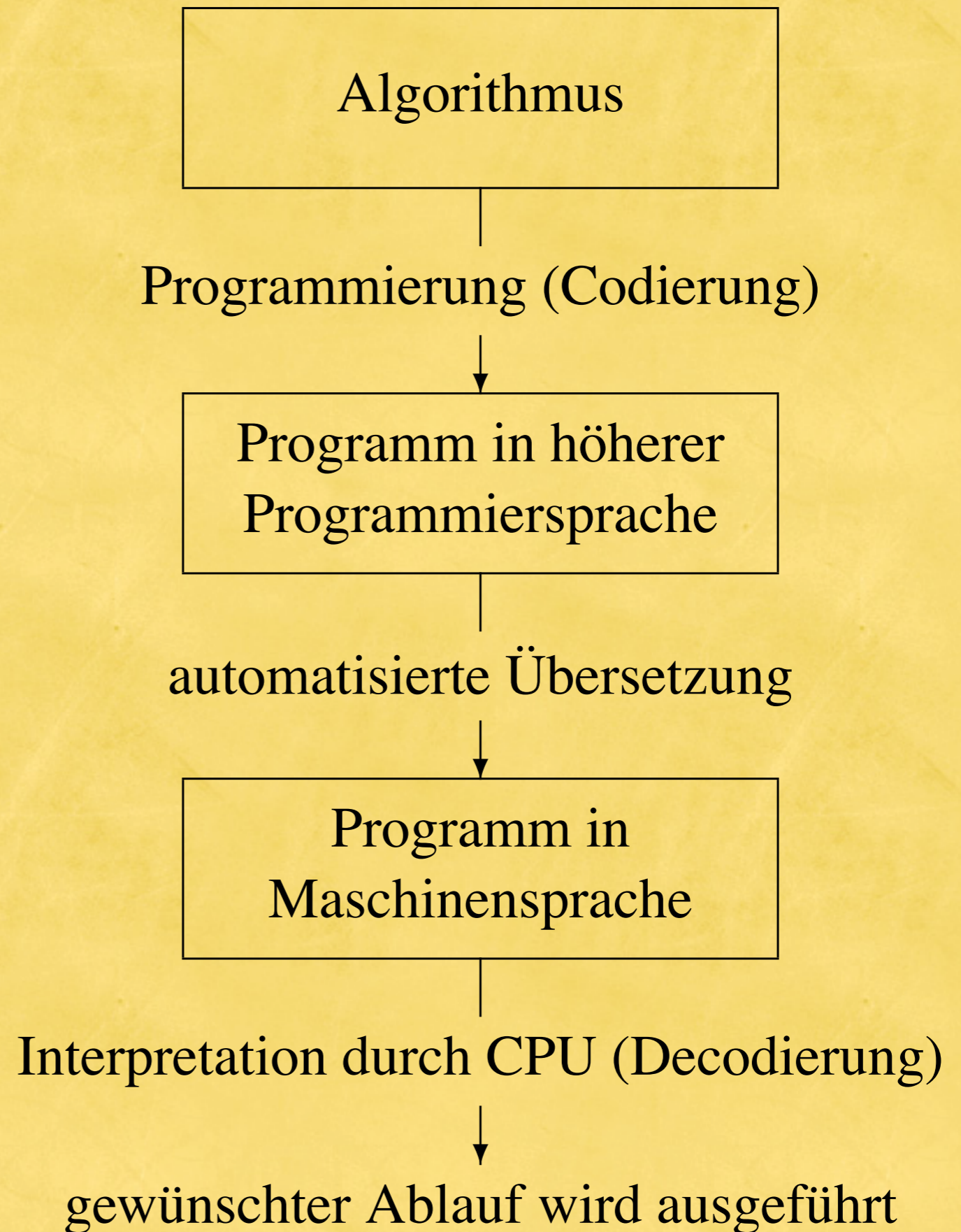


# Programmiersprachen

- 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



Hauptthema  
der  
Vorlesung





# Hierarchie bei Programmiersprachen

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







# 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

- 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, Macintosh ...) anpassen
- Der einmal kompilierte Code kann dann auf allen Architekturen ausgeführt werden
- Dies macht Java zur WWW-Sprache par excellence und ermöglicht auch die Programmierung von systemübergreifenden Applicationen




# Die Software-Hardware Hierarchie

Anwendungssoftware  
z.B. Textverarbeitung,  
Statistikpaket

Systemsoftware  
z.B. Betriebssystem, Editor,  
Compiler, Eclipse

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



Programmier-  
umgebung

In Coma:  
Unix + emacs  
später Eclipse



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



# Durchführung 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

- 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

- 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 Prozesse, 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 die 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

## • Java

- selber schauen, was dem eigenen Niveau entspricht

- Bruno R. Preiss. **Data Structures and Algorithms with Object-Oriented Design Patterns in Java.**

John Wiley & Sons, New York, 2000

## • Coma Skript

- <http://www.math.tu-berlin.de/coma/>