

Teil

Imperative Programmierung

Unterrichtseinheit 1

Grundlagen der Programmierung

Dr. Dietrich Boles

- Terminologie
- Algorithmus
- Programmiersprachen
- Entwicklungsphasen
- Entwicklungswerkzeuge
- Computer
- Zusammenfassung

Programmierung:

Erstellung von Computerprogrammen

Softwareentwicklung:

Methoden zum Lösen von Problemen mit dem Computer

Programmiersprache:

Computerverständliche Notation zur Formulierung von Programmen

Programm:

In einer Programmiersprache formulierte Arbeitsanleitung
(→ Algorithmus) für einen Computer

Programmierer:

Entwickler von Programmen

Programmcode, Quellcode, Sourcecode:

(textuelle) Programmbeschreibung

ausführbares Programm:

(binäres) Programm in maschinenverständlicher Form

Programmaufruf:

Ausführung eines ausführbaren Programms

Arbeitsanleitungen:

- Kochrezepte
- Bastelanleitungen
- Partituren
- Spielregeln

Aufbau:

- Menge von Anweisungen

Charakteristika:

- Anweisungssequenzen
- bedingte Anweisungen
- Anweisungsschleifen
- Zutaten / Voraussetzungen
- "schwammige" Formulierungen

Definition:

Arbeitsanleitung zum Lösen eines Problems bzw. einer Aufgabe, die so präzise formuliert ist, dass sie von einem Computer ausgeführt werden kann

Formulierung von Algorithmen:

- Umgangssprachlich
- Programmiersprache
- Programmablaufpläne
- Struktogramme

Ausführung von Algorithmen:

- durch einen Prozessor (Mensch / Computer) → Prozess
- Computer: schnell, zuverlässig, hohe Speicherfähigkeit

Beispiel: Berechnung der Summe aller Zahlen von 1 bis n

Umgangssprachliche Formulierung:

Addiere für eine vorgegebene natürliche Zahl n die Zahlen von 1 bis n. Dies ist das Resultat.

Programmiersprache:

```
int n = readInt();  
int erg = 0;  
int aktZahl = 1;  
while (aktZahl <= n) {  
    erg = erg + aktZahl ;  
    aktZahl = aktZahl + 1;  
}  
printInt(erg);
```

theoretisch:

- Eindeutigkeit
- Parametrisierbarkeit
- Finitheit
- Ausführbarkeit
- Terminierung
- Determiniertheit
- Determinismus

praxisrelevant:

- korrekt
- vollständig
- effizient (Zeit)
- sparsam (Speicher)
- erweiterbar
- wieder verwendbar
- portabel
- verständlich

Anwender:

- Anfänger
- Experten

Abstrahierungsgrad:

- maschinennah
- problemorientiert

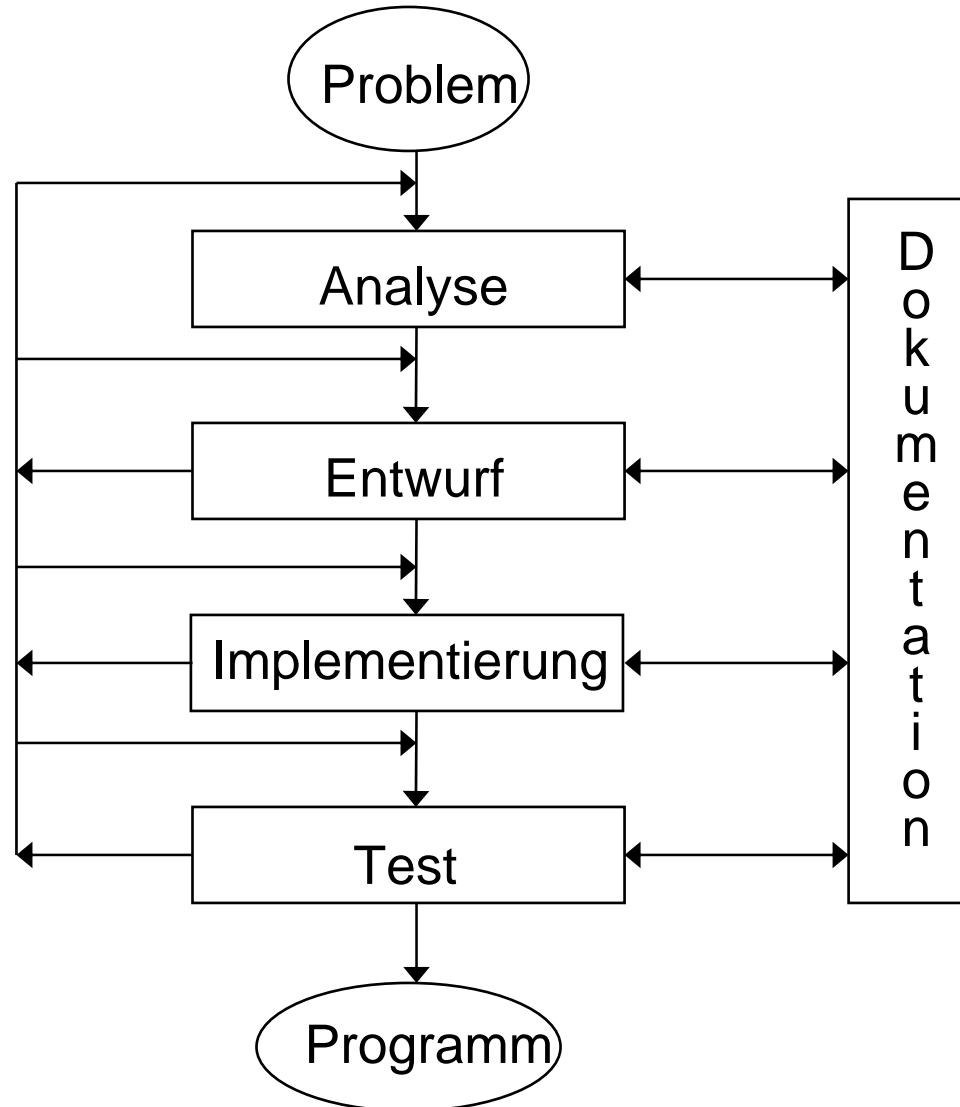
Komplexität:

- problemspezifisch
- "general-purpose"

Paradigmen:

- imperativ
- funktional
- prädikativ
- regelbasiert
- objektorientiert

- Lexikalik:** gültige Zeichen und Wörter der Sprache
- Syntax:** korrekter Aufbau von Sätzen der Sprache
- Semantik:** Bedeutung von Sätzen der Sprachen
- Pragmatik:** Einsatzbereich einer Sprache



Analyse:

- Untersuchung des Problems und des Problemumfelds
- Präzisierung der Problemstellung

Entwurf:

- Entwicklung des Algorithmus
- Kreativer Prozess (Intelligenz, Erfahrung)

Implementierung:

- Übertragung des Entwurfs in eine Programmiersprache
- Eingabe in den Computer

Test:

- Überprüfung des Programms auf logische und Laufzeitfehler
- Debugging

Dokumentation:

- Exakte Problemstellung
- Beschreibung der generellen Lösungsidee
- Beschreibung des Lösungsverfahrens
- Programmcode
- Beschreibung der Testmengen
- Protokolle der Testläufe
- Aufgetretene Probleme
- Alternative Lösungsansätze

Weitere Tätigkeiten:

- Effizienzverbesserung
- Wartung
- Portierung auf andere Plattformen

Editore: Manipulation des Programmcodes

Compiler: Transformation eines Quellprogramms in ein Zielprogramm

Interpreter: Inkrementelle Abarbeitung des Quellcodes

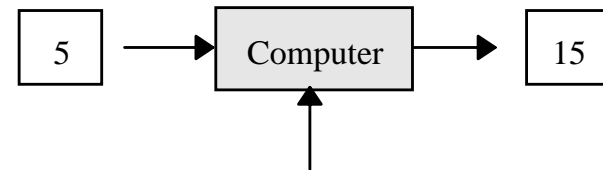
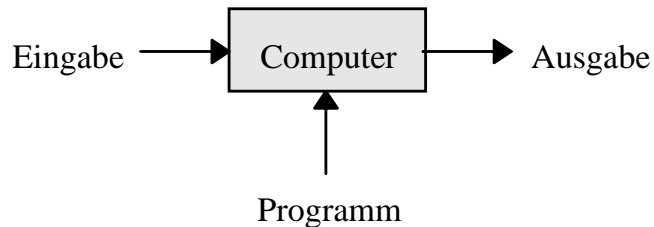
Debugger: Erkennung von Laufzeitfehlern

Dokumentationshilfen: Erstellung von Teilen der Dokumentation

Laufzeitsystem: Hilfsprogramme bei der Programmausführung

Programmbibliotheken: Sammlungen fertiggestellter Programme

- Gerät zur automatischen Verarbeitung von Daten
- Computer setzen sich zusammen aus
 - Hardware (physikalische Geräte; Zentraleinheit plus periphere Geräte)
 - Software (Programme)
- Arbeitsweise:

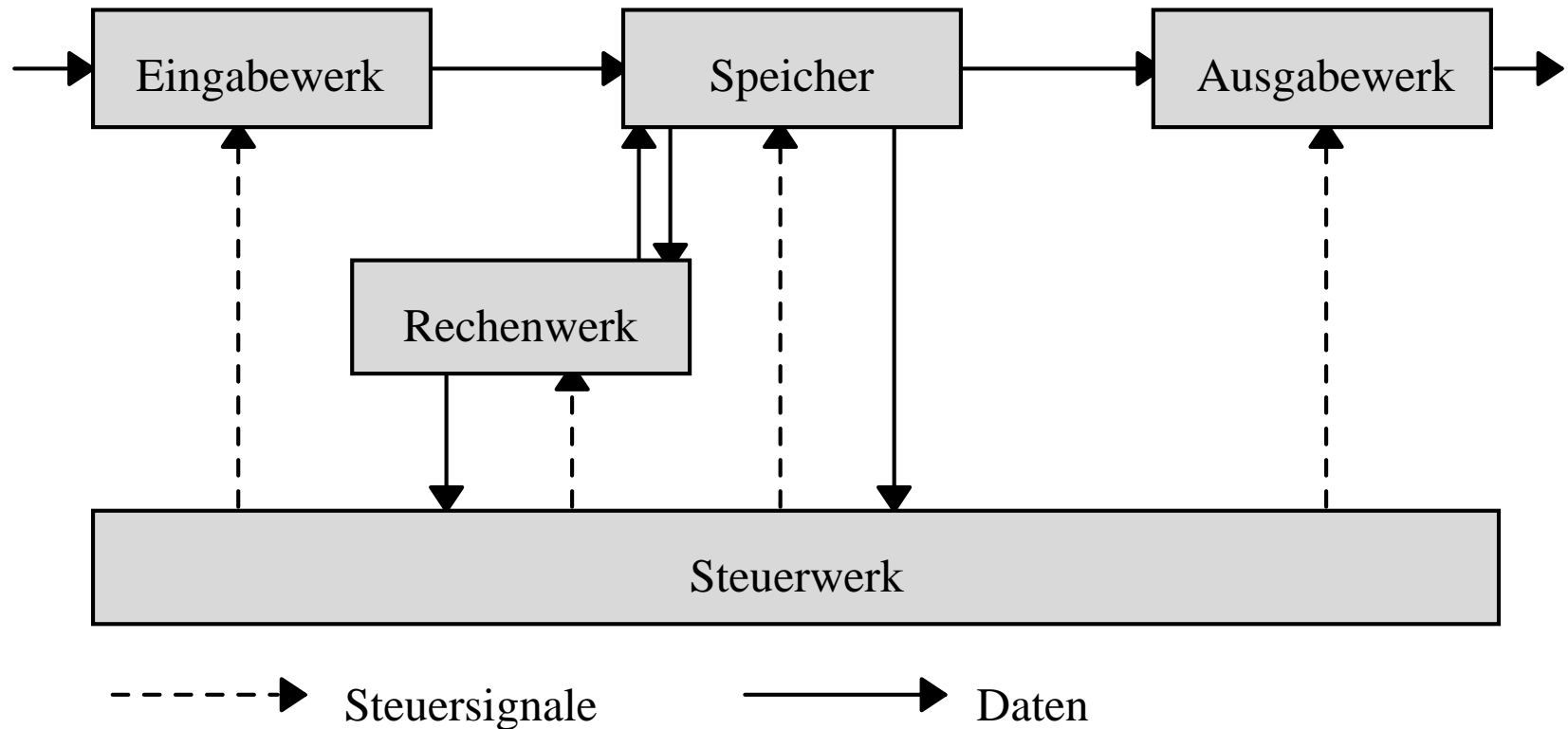


Lies eine Zahl ein.

Addiere alle natürlichen Zahlen
von 1 bis zur eingegebenen Zahl.

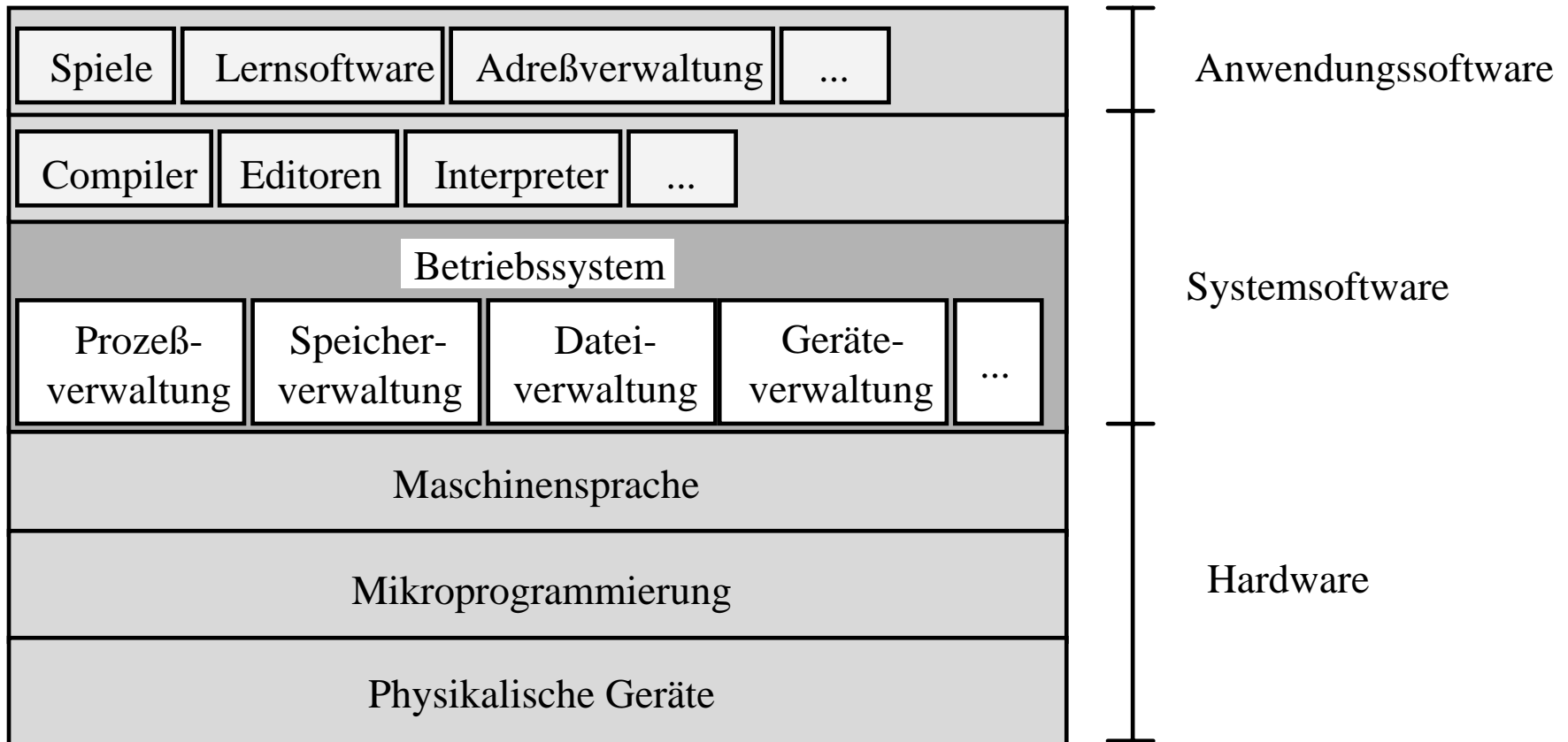
Gib dieses Ergebnis auf den Bildschirm aus.

Von-Neumann-Rechnerarchitektur:



- aufeinander folgende Befehle in aufeinander folgenden Zellen
- (sequentieller) Zugriff durch Steuerwerk (via Befehlsadresse)
- arithmetische Befehle (Addition, Multiplikation, ...)
- logische Befehle (Negation, Konjunktion, Vergleiche, ...)
- Sprungbefehle
- Transportbefehle
- Schiebeoperationen
- Ein-/Ausgabebefehle

- Speicher: Aufbewahrung von Programmen und Daten
- 1 Bit: kleinstes Speicherelement (2 Zustände: 0 und 1)
- 1 Byte: 8 Bit (2^8 Zustände) —————> Speicherzelle
- 1 Wort: 4 oder 8 Speicherzellen



Betriebssystem:

Menge aller Programme, die den Betrieb eines Computer bewältigt:

- Prozessverwaltung
- Speicherverwaltung
- Dateiverwaltung
- Geräteverwaltung
- ...

Festplatte: Billiger Hintergrundspeicher

Dateien: Logische Behälter zum Speichern von Daten

Verzeichnisse: Hilfsmittel für eine übersichtl. Strukturierung von Dateien

Window-System: Aufteilung des Bildschirm in unabhängige "Windows"

Window-Manager: Verwalter von "Windows"

- Computer: Hard-/Software zur automatischen Verarbeitung von Daten
- Programmieren: systematisches Entwickeln von Computerprogrammen zum Lösen gegebener Probleme
 - Durchlaufen mehrerer Phasen
 - Nutzung von Werkzeugen
- Algorithmus: Formulierung des Lösungsverfahrens
- Programmiersprache: Notation zur Formulierung von Algorithmen