

# Programmierkurs Java

## UE 1 - Grundlagen

Dr.-Ing. Dietrich Boles

**Programmieren ist wie küssen:  
Man kann darüber reden,  
man kann es beschreiben,  
aber man weiß erst, was es bedeutet,  
wenn man es getan hat.**

**(Andree Beaulieu-Green)**

- Terminologie
- Programmiersprachen
- Entwicklungsphasen
- Entwicklungswerkzeuge
- Compiler
- Computer
- Zusammenfassung

## **Programmierung:**

Erstellung von Computerprogrammen

## **Softwareentwicklung:**

Methoden zum Lösen von Problemen mit dem Computer

## **Algorithmus:**

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

## **Programmiersprache:**

Computerverständliche Notation zur Formulierung von Programmen

## **Programm:**

In einer Programmiersprache formulierte Algorithmus

## **Programmierer:**

Entwickler von Programmen

## **Programmcode, Quellcode, Sourcecode:**

(textuelle) Programmbeschreibung

## **ausführbares Programm:**

(binäres) Programm in maschinenverständlicher Form

## **Programmaufruf:**

Ausführung eines ausführbaren Programms

## Anwender:

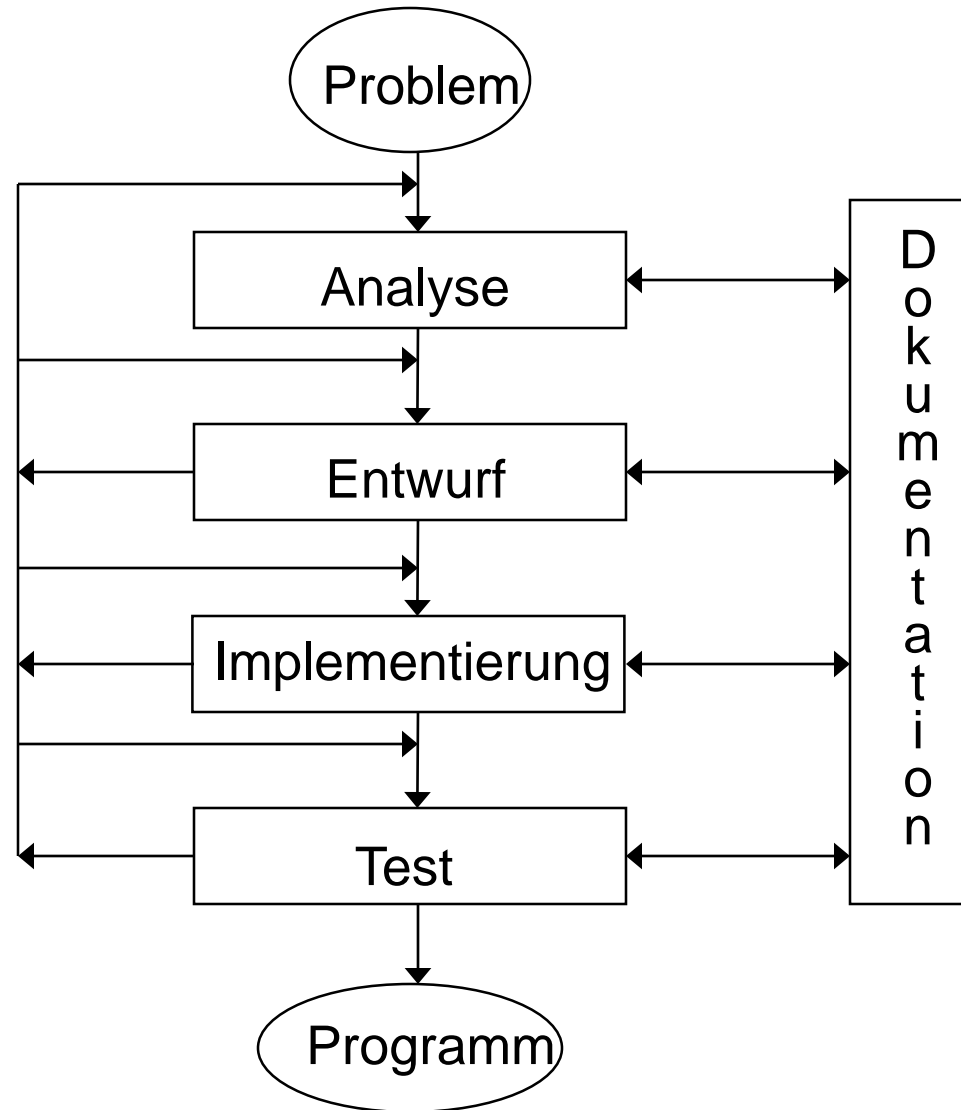
- Anfänger
- Experten

## Abstrahierungsgrad:

- maschinennah
- problemorientiert

## Paradigmen:

- imperativ
- funktional
- prädikativ
- regelbasiert
- objektorientiert



**Editore:** Manipulation des Programmcodes

**Compiler:** Transformation eines Quell- in ein Zielprogramm

**Interpreter:** Inkrementelle Abarbeitung des Quellcodes

**Debugger:** Erkennung von Laufzeitfehlern

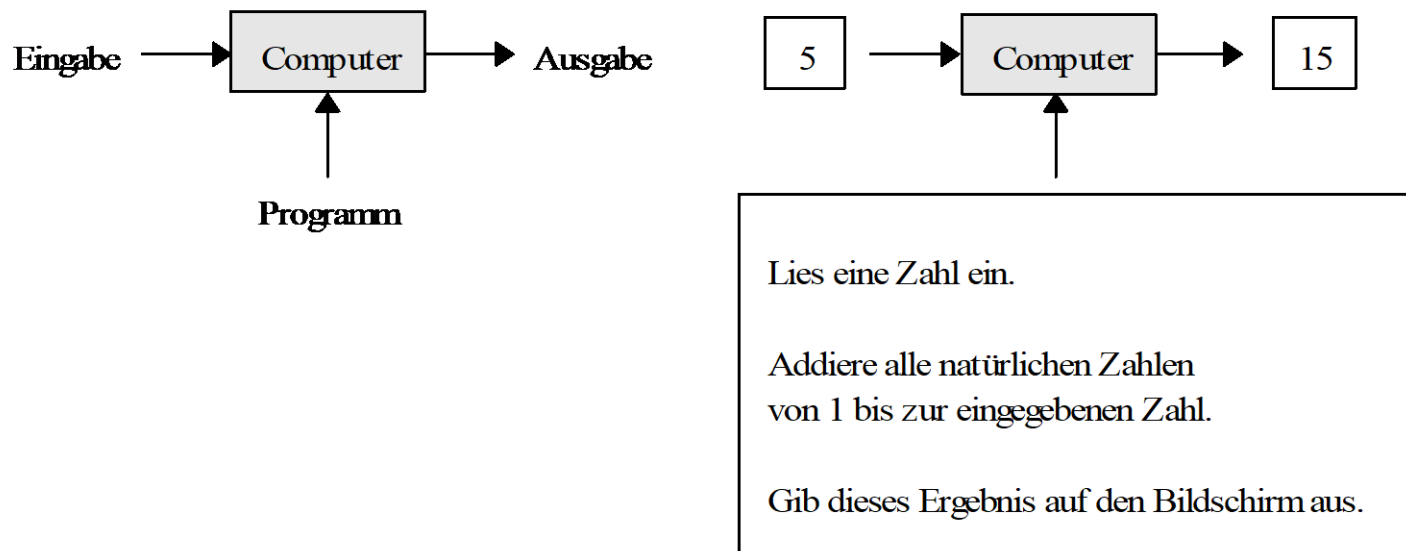
**Dokumentationshilfen:** Erstellung von Teilen der Dokumentation

**Laufzeitsystem:** Hilfsprogramme bei der Programmausführung

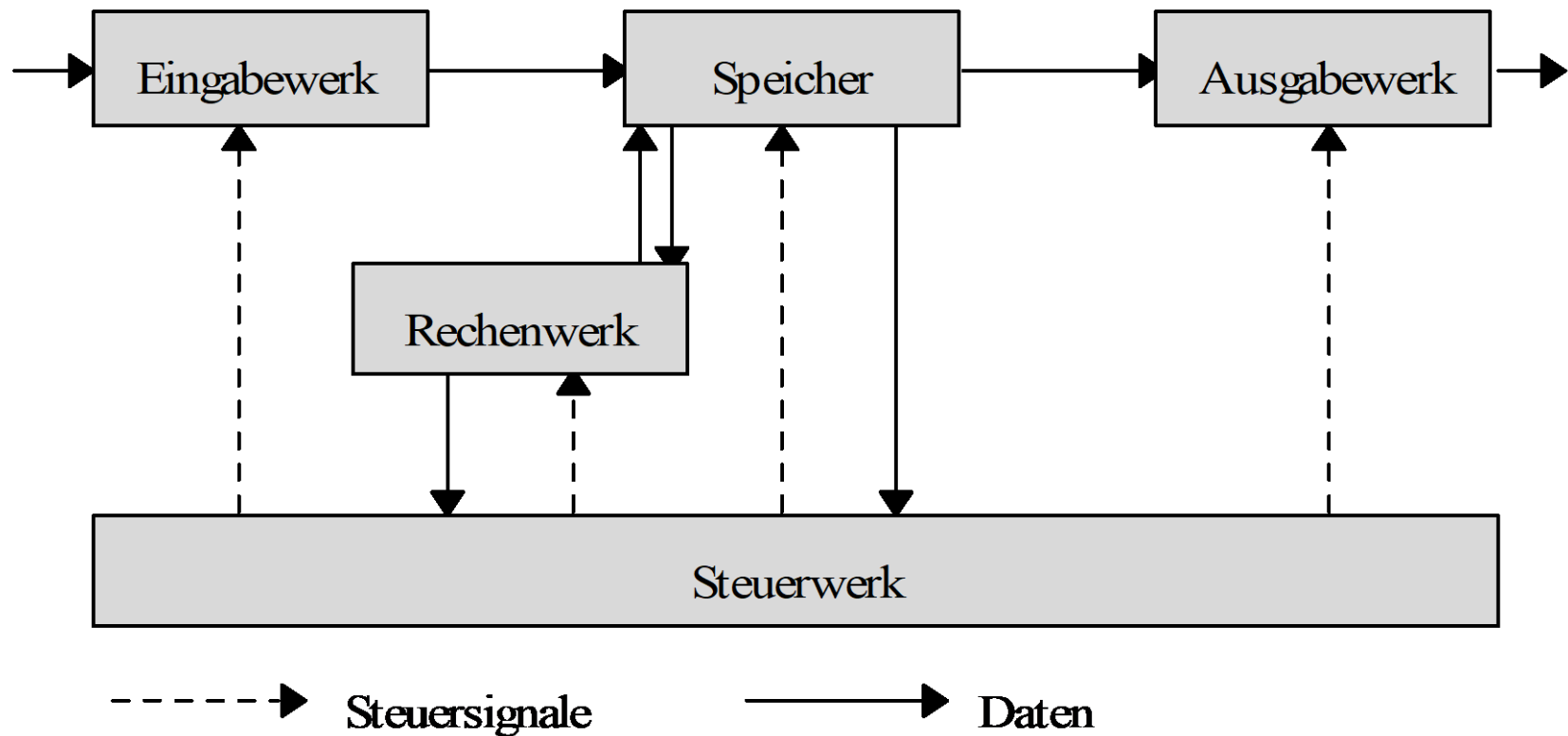
**Programmbibliothek:** Sammlung fertiggestellter Programme

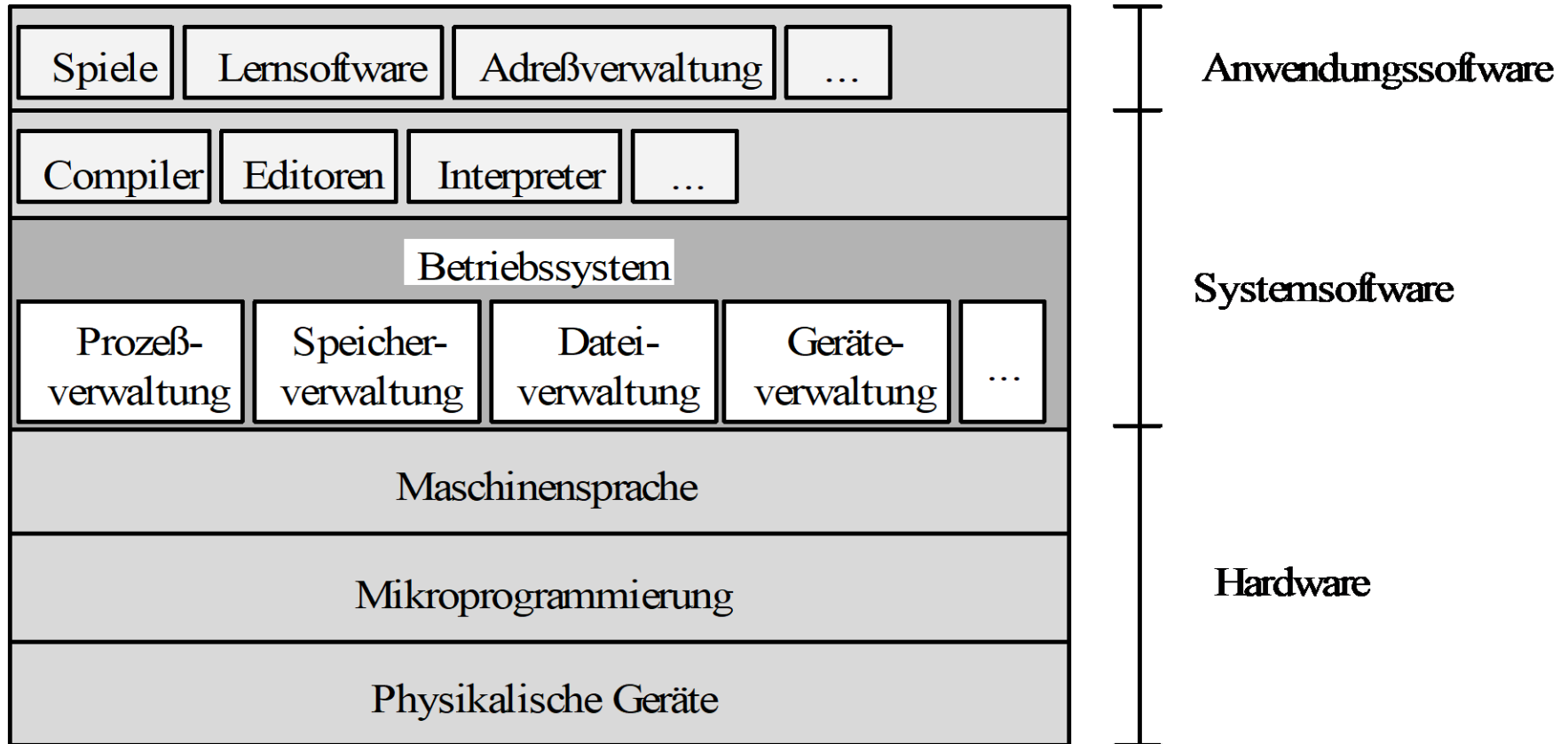


- Gerät zur automatischen Verarbeitung von Daten
- Computer setzen sich zusammen aus
  - Hardware (physikalische Geräte; Zentraleinheit plus periphere Geräte)
  - Software (Programme)
- Arbeitsweise:



- Von-Neumann-Rechnerarchitektur:



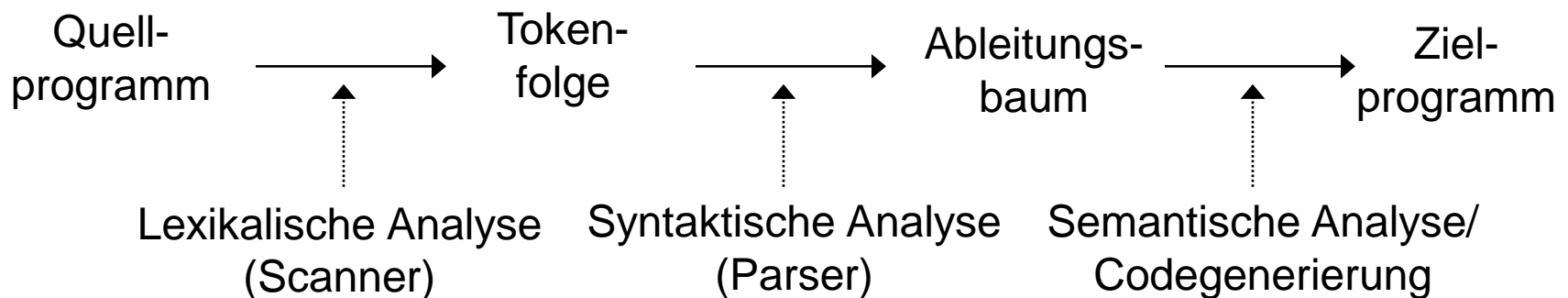


**Lexikalik:** gültige Zeichen und Wörter der Sprache

**Syntax:** korrekter Aufbau von Sätzen der Sprache

**Semantik:** Bedeutung von Sätzen der Sprachen

```
int n = IO.readInt();
int erg = 0;
int aktZahl = 1;
while (aktZahl <= n) {
    erg = erg + aktZahl++;
}
System.out.println(erg);
```



- Entfernung von Trennzeichen und Kommentaren
- Erkennung von **Token** (Zeichen, die bedeutungsmäßig zusammengehören):
  - Schlüsselwörter
  - Bezeichner
  - Symbole
  - Zeichenketten
  - ...

```
while (zahl >= 17) {  
    zahl = zahl-1;  
}
```

Java



```
while      zahl  
(          =  
zahl       zahl  
>=        -  
17         1  
)         ;  
{         }
```

- Überführung einer Tokenfolge in Ableitungsbaum
- Untersuchung auf syntaktische Korrektheit
- Darstellung: Syntaxdiagramme, BNF, EBNF

## Backus-Naur-Form (BNF):

- Technik zur textuellen Darstellung der Syntax einer Sprache
- Verwendung von Ersetzungsregeln (Produktionen)
- besitzen linke und rechte Seite
- linke Seite: Nichtterminalsymbol
- Nichtterminalsymbol: durch  $\langle \rangle$  gekennzeichnet
- Alternativen: durch  $|$  gekennzeichnet
- $\epsilon$  (Epsilon): leere Alternative

## EBNF:

- Erweiterung der BNF (Abkürzungsmöglichkeiten)
- $[ \dots ]$  bedeutet: Symbole in Klammern können auch wegfallen
- $\{ \dots \}$  bedeutet: Symbole in Klammern können beliebig oft (auch Null mal) wiederholt werden

## EBNF:

```

<programm>          ::= [ "void" ] "main" <block>
<block>            ::= "{" <anweisungen> "}"
<anweisungen>     ::= <anweisung> { <anweisung> }
<anweisung>       ::= "anw1" | "anw2" | <repeat-anweisung>
<repeat-anweisung> ::= "repeat" <block>
    
```

## Gültige Programme:

```

main {
  anw1
}
    
```

```

void main {
  anw2
  repeat { anw1 anw2 }
  anw1
}
    
```



- 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: computerverständliche Notation zur Formulierung von Algorithmen