

---

# Einführung in Softwaretechnologien

## Zusammenfassung

Christian Lengauer

Fakultät für Informatik und Mathematik



# Syntax und Semantik

- **Programmbeschreibung:**
  - Syntax = textueller Aufbau, bestimmt Zugehörigkeit oder Nichtzugehörigkeit zur Programmiersprache (d.h. Legalität des Programms)
  - Semantik = Bedeutung, entscheidet Korrektheit oder Unkorrektheit des Programms
- **Übersetzer (Compiler):** `>javac Quelldateiname.java`
  - überprüft Zugehörigkeit des Programms zur Programmiersprache
  - generiert ausführbaren Code (bei Java: für die Java Virtual Machine)
- **Java Virtual Machine (JVM):** `>java Quelldateiname`
  - führt das Programm aus
  - es gibt JVM-Installationen für viele verschiedene Prozessortypen

# Sprachelemente von Java

- **Datentypen:**
  - Typ = (Menge von Werten, Menge von Operationen)
  - primitive Typen: `int`, `double`, `char`, `boolean`
  - Klassen (Objekttypen):
    - vordefiniert: `String`, `Array`, `Integer`, `Exception`, ...
    - benutzerdefiniert: siehe BlueJ-Projekte
- **Vererbung:** bildet Klassenhierarchien (`extends`)
- **Abstraktion:** gibt nur Signatur, ohne Implementierung (`abstract`)
- **Interface:**
  - Klasse mit nur Signaturen (`interface`)
  - Implementierungen müssen in Subklassen folgen (`implements`)
- **Exception:** Sonderfall
  - muss angezeigt (geworfen) werden (`throw`)
  - kann behandelt werden (`try-catch`)

# Datenstrukturen in Java

## Zentrale Herausforderungen

- **Repräsentation der Daten:**
  - Zahlen: `int` / `double` / `String` (Gleitpunktproblematik)
  - Textuelle Strukturen: `String` / Syntaxbaum
- **Aggregation der Daten:**
  - konkret:
    - `Array`: geordnet, endlich indiziert (`[0..n]`)
    - `List`: geordnet, unindiziert, unbeschränkt
    - `Tree`: halbgeordnet, unindiziert, unbeschränkt
    - `Hash`: geordnet, unbeschränkt indiziert (`String`)
  - abstrakt: Sammlungen
    - `Set`: ungeordnet, keine Wiederholung
    - `Map`: ungeordnet, Urbild/Bild-Paare
- **Iteratoren:**
  - konkret: `for` (Zähl), `while` (Prätest), `do` (Posttest)
  - abstrakter: `while` mit Iterator über Sammlung (Prätest)
  - noch abstrakter: `for-each` über Sammlung (Zähl)

# Eigenschaften von Programmen

- **Statische Eigenschaften:** (z.B. Klassendiagramm)
  - für alle Läufe des Programms gültig
  - vom Übersetzer feststellbar (im Prinzip!)
- **Dynamische Eigenschaften:** (z.B. Objektdiagramm)
  - nur für bestimmte Läufe des Programms gültig
  - nicht von Übersetzer feststellbar
  - relevant für:
    - Vererbung: dynamische Methodensuche
    - Sonderfallbehandlung: Laufzeitsonderfälle
    - Programmoptimierung: “just-in-time” Kompilation

# Eigenschaften der Sprache Java

- **Strenge Typisierung:**
  - Jedem Ausdruck in einem Javaprogramm kann vom Übersetzer ein eindeutiger Typ zugeordnet werden.
  - Vorteile:
    - Hohe Fehlererkennungsrate des Übersetzers
    - Hohe Effizienz und Robustheit des generierten Codes
- **Erweiterbarkeit:** über Klassenbibliotheken
  - garantiert vorhanden: Standardpakete
  - zusätzlich: weit verbreitete Pakete ausserhalb des Standards
  - für kleinere Benutzergemeinden: domänenspezifische Pakete (z.B. effizientere Arraypakete für Hochleistungsanwendungen)
  - persönliche Klassenbibliotheken