

Gedanken zu den Gegenständen von Grund- und Leistungskursen Informatik aus Sicht der Fachwissenschaft

Andreas Schwill
Institut für Informatik
Universität Potsdam

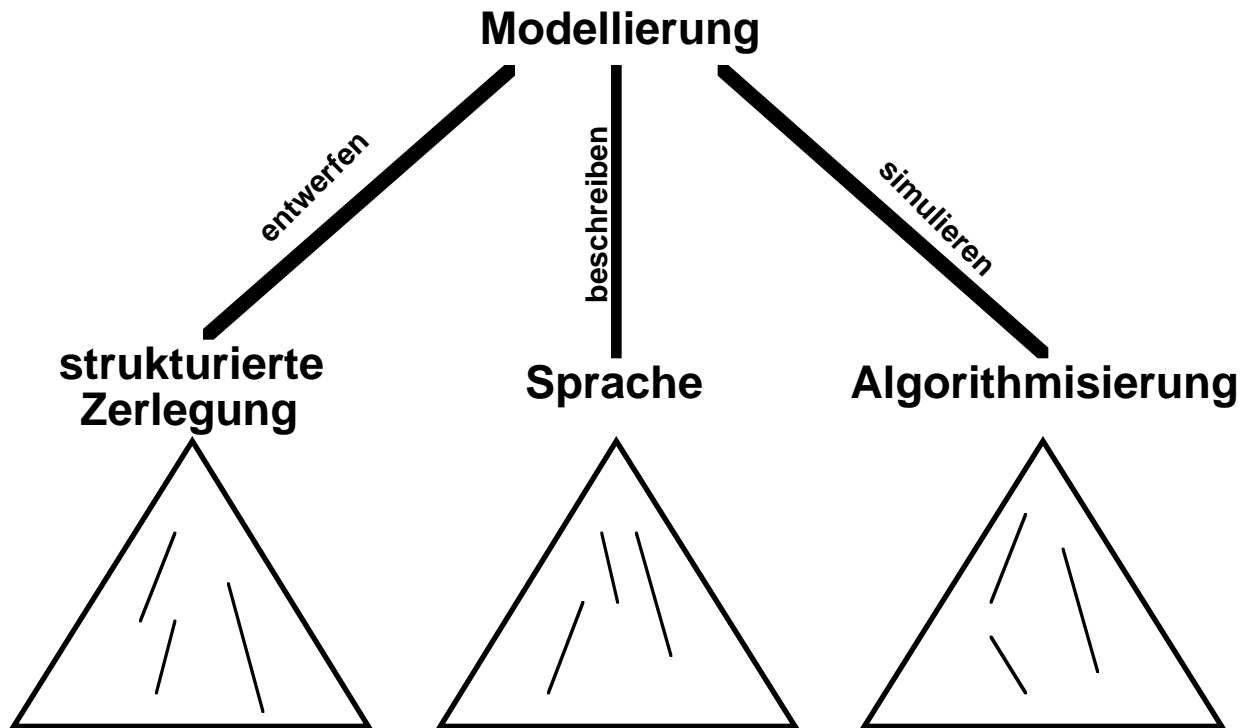
Überblick

- **Annahmen**
- **Fundamentale Ideen als Entscheidungskriterium**
- **Weitere Aspekte**
 - **Arbeitsformen**
 - **Theorie**
 - **Grenzbereiche der Informatik**

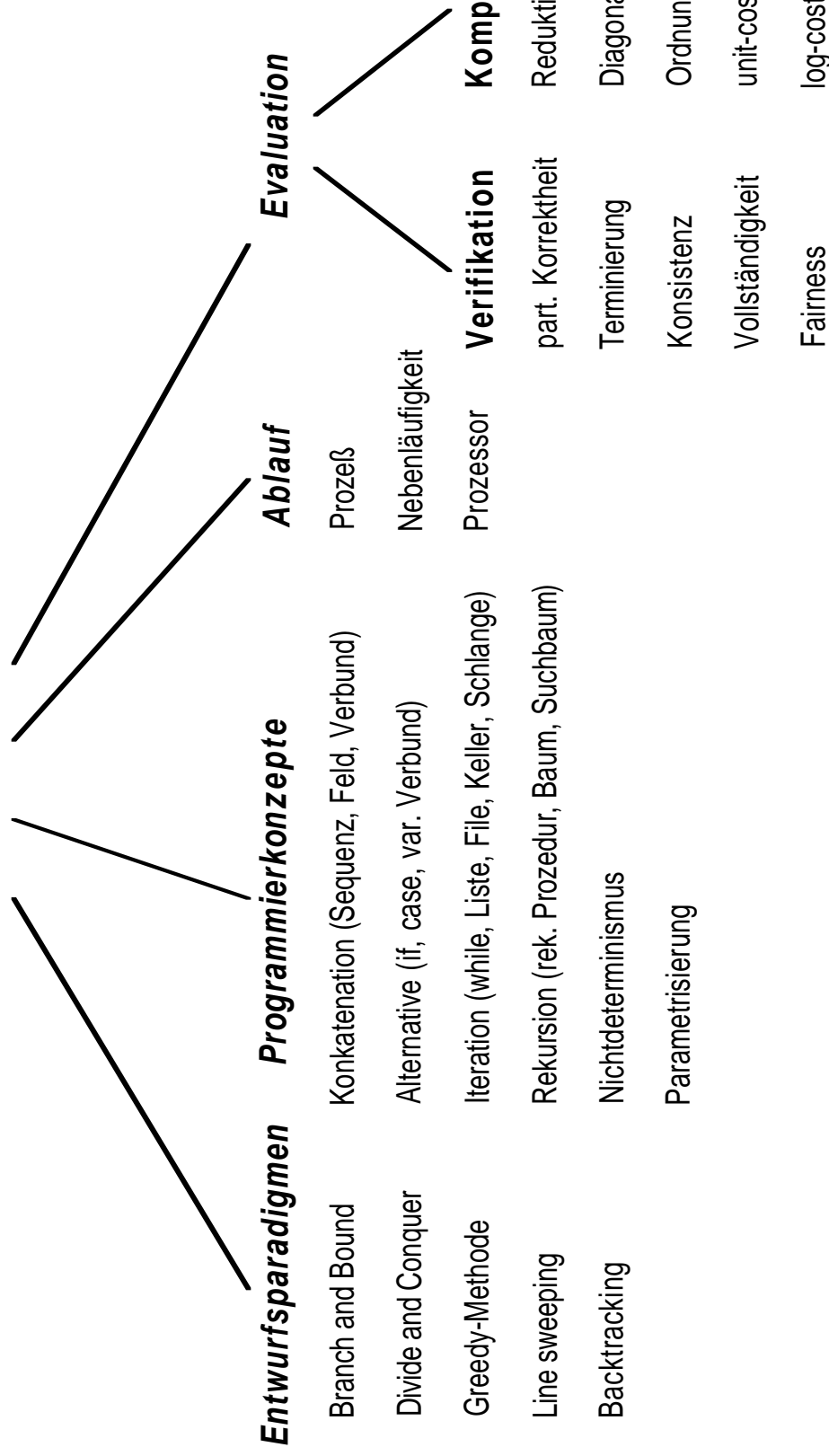
Vorgehensweise und Annahmen

- **keine Berücksichtigung von länderspezifischen Eigenarten**
- **keine Machbarkeitsaspekte bezgl. Ausstattung, Schülermaterial, Lehrerkenntnissen**
- **Informatik als Wissenschaft von der systematischen Verarbeitung von Information, insbesondere mithilfe von Digitalrechnern (Rechner im Nebensatz!)**
- **Informatik als Strukturwissenschaft**
- **Vergleich mit anderen Wissenschaften, insbesondere Physik, Mathematik**

Fundamentale Ideen



Algorithmisierung



worst/average/
amortized case

Sprache

```
graph TD; Sprache --> Syntax; Sprache --> Semantik; Syntax --> Erkennen; Syntax --> Erzeugen; Semantik --> Konsistenz; Semantik --> Vollständigkeit; Semantik --> Transformation;
```

Syntax

Erkennen

Erzeugen

Semantik

Konsistenz

Vollständigkeit

Transformation

strukturierte Zerlegung

Modularisierung

- Methoden
- Top-down-Methode
- Bottom-up-Methode
- Geheimnisprinzip (black box Denken)
- Hilfsmittel
- Lokalität von Objekten
- Spezifikation
- abstrakter Datentyp
- Teamarbeit

Hierarchisierung

- Darstellung
- Schachtelung
- Baum
- Klammerung
- Einrückung
- Realisierung
- Übersetzung
- Interpretation
- operationale Erweiterung

Orthogonalisierung

Emulation

Divide and Conquer

Grundkurs	Leistungskurs
Erklärung des Prinzips	Notation des Verfahrens in einem Algorithmenschema
Verständnis für die Funktionsweise	Beherrschung des Verfahrens, Darstellung durch Bäume
Anwendung an kleinen Beispielen (z.B. binäres Suchen)	Anwendung in unterschiedl. Zusammenhängen (Suchen, Sortieren, algor. Geometrie)
Grenzen des Verfahrens	Formale Effizienzanalyse des Verfahrens für ausgewählte Beispiele (z.B. Quicksort)

Nichtdeterminismus

Grundkurs	Leistungskurs
"Das gibt es"	Sprachelemente für Nichtdeterminismus
Beispiel: Quicksort	Einfache Beispiele (Spiegeln einer Zahlenfolge)
	Nichtdeterminismus bei Suchproblemen (Klassenbildung)
	Vergleich der Effizienz

Rekursion

Grundkurs	Leistungskurs
Erklärung des Prinzips	Beherrschung der Idee, rekursiver Vertrauenssprung
Durchführung an enaktiven oder ikonischen Beispielen (Schachtelungsdiagramm)	Unterschiedliche Darstellungsformen (Bäume, Schachtelungen, textuelle Beschreibungen, unterschiedl. Programmiersprachen)
Beispiele für enaktiv nachvollziehbare Rekursionen (Fakultät, gerade/ungerade)	Einbettung in reale Probleme (Zugverbindungen, Graphik, Geometrie, Syntaxanalyse, Suchprobleme)
Umsetzung in eine Programmiersprache	Implementierung von Rekursion (Stack)
Vergleich iterativer und rekursiver Darstellungen	Zusammenhang zwischen Iteration und Rekursion, Umwandlung rekursiver in iterative Programme
	Effizienzbetrachtungen (Zeit, Speicher)

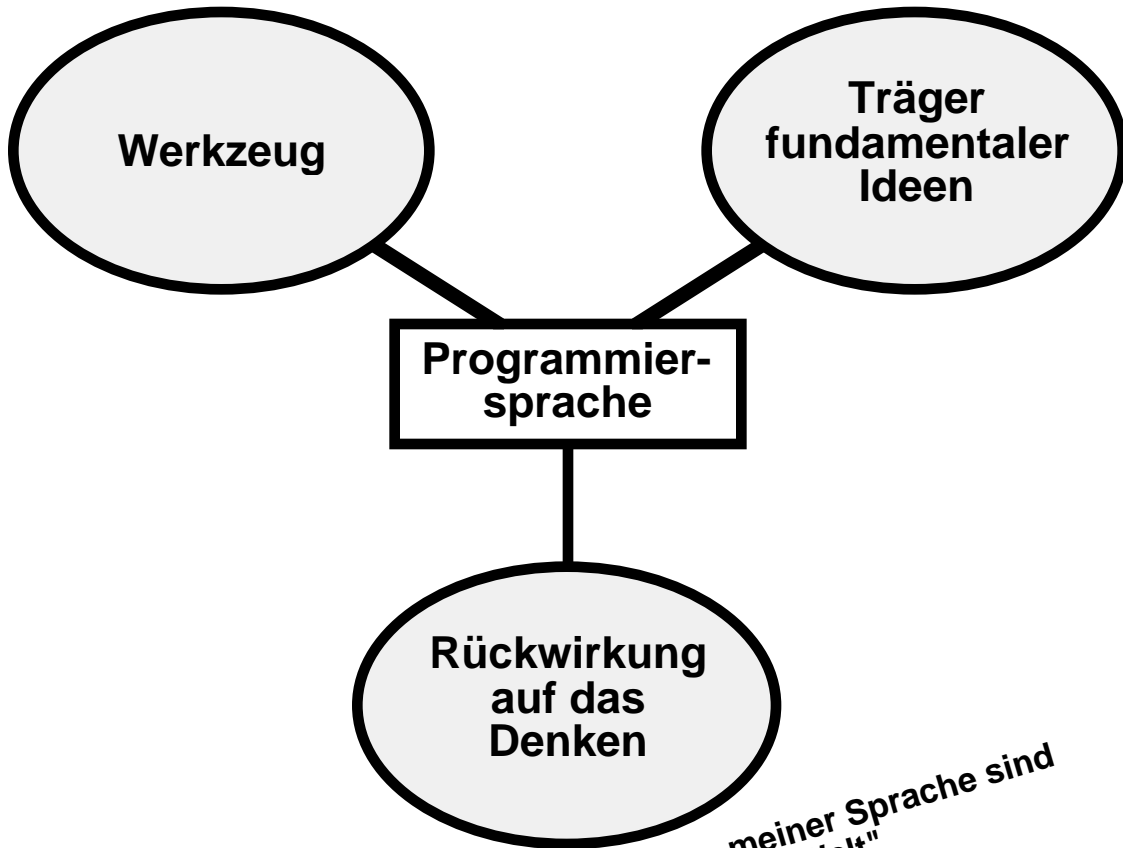
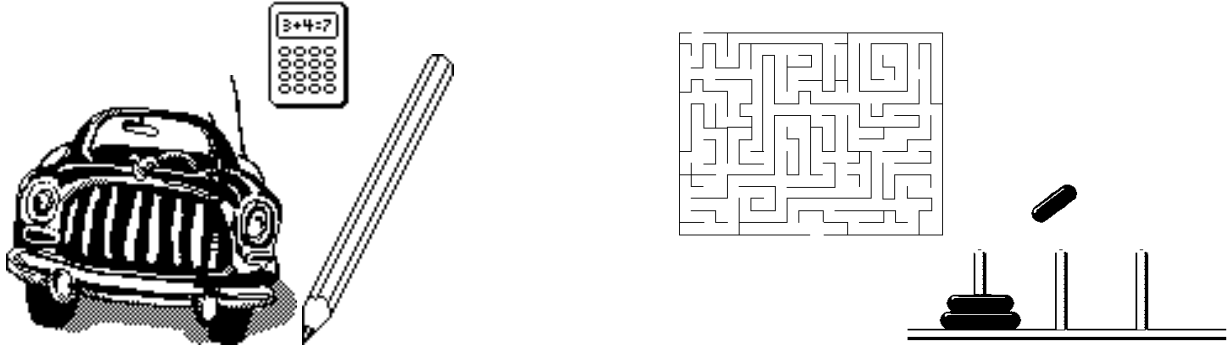
Verifikation

Grundkurs	Leistungskurs
Begriff der Korrektheit	Begriff der partiellen Korrektheit und der Terminierung, Zusammenhang zum Testen
Unterschiedliche Korrektheitsniveaus, Problem der Korrektheit, aktuelle Anforderungen an die Korrektheit von Programmen	Testmethoden (Anweisungsüberdeckung, Pfadüberdeckung)
Systematisches Testen	Formaler Nachweis von Korrektheit an einfachen Beispielen (Tausch von Variablen)
Grenzen des Testens und von Korrektheitsanforderungen	Zusammenhang zum Halteproblem

Komplexität

Grundkurs	Leistungskurs
Begriff der Betriebsmittel Speicher, Laufzeit	Quantifizierung von Speicher und Laufzeit
Einheitskostenmodell, schlimmster Fall	Formalisierung
Ordnung eines Algorithmus	O-Kalkül, untere Schranken für Probleme (Sortieren)
Komplexität einfacher Algorithmen	schwere Probleme, Reduktion

Bedeutung von Programmiersprachen



Wittgenstein: "Die Grenzen meiner Sprache sind die Grenzen meiner Welt"

Sapir-Whorf-These: Linguistisches Relativitätsprinzip



Sprache

Grundkurs	Leistungskurs
imperative Programmierung	ein weiterer Programmierstil, Bewertung von Programmiersprachen
Syntaxdiagramme für Programmier- und andere Sprachen	Grammatiken, Sprachklassen, Maschinenmodelle, Chomsky-Hierarchie
umgangssprachlich "exakte" Darstellung der Semantik, Idee der operationalen Semantik	Beschreibung der Semantik von Sprachkonzepten durch ein Maschinenmodell (operationale Semantik)
Syntaxanalyse von einfachen Sprachen	Übersetzerbau

Weitere Aspekte

- **Projektarbeit**
 - intensives Arbeiten in Projekten zu selbst gewählten Themen
 - Berücksichtigung pädagogischer und informatischer Aspekte
 - Dokumentation
 - Präsentation
 - Gesprächsführung
 - Protokollführung

- **Theorie**
 - Theoretische Informatik ist die Klammer des Leistungskurses
(-> Strukturwissenschaft)
 - keine Theorie oder Anwendungsbezug
 - Einbindung in den behandelten Zusammenhang
 - Algorithmen -> Laufzeit/O-Kalkül
 - schwierige Probleme -> NP-Vollständigkeit
 - Übersetzer -> Grammatiken
 - Computernetze -> Graphen
 - ecommerce -> RSA-Verfahren

- **wissenschaftliches Praktikum an einer Universität (14 Tage)**
 - **Mitwirkung am wiss. Erkenntnisprozeß in kleinen Bezirken**
 - **Erarbeiten von kleinen Softwarebausteinen**
 - **eigenverantwortliches Erstellen von Dossiers**
 - **Bearbeitung von Literatur/Kennenlernen einer Bibliothek/Suchen und Finden/Typische Fachliteratur/...**
 - **Präsentation**

- **Gefühl für die Grenzen der Informatik**
 - **Betonung der Nicht-Machbarkeit gegenüber der Machbarkeit (Bsp.: z.Zt. keine korrekte Software zu entwickeln)**
 - **Vermittlung der Grenzen des Fachs und ihrer eigenen**
 - **Wir stehen in der Informatik noch am Anfang**
 - **Wo sind aktuelle Forschungsaufgaben?**
 - **Was ist noch ungelöst?**
 - **Wie nah waren wir gerade an der Grenze der Wissenschaft?**