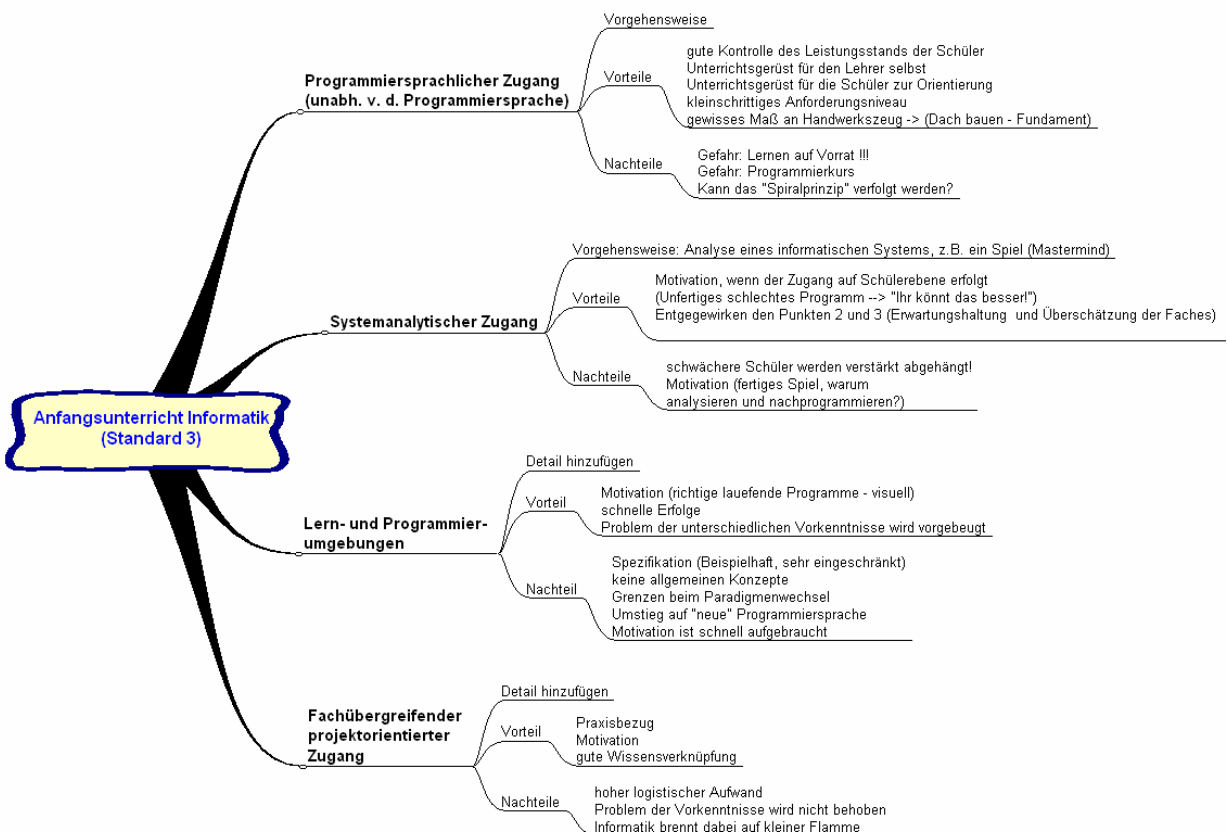
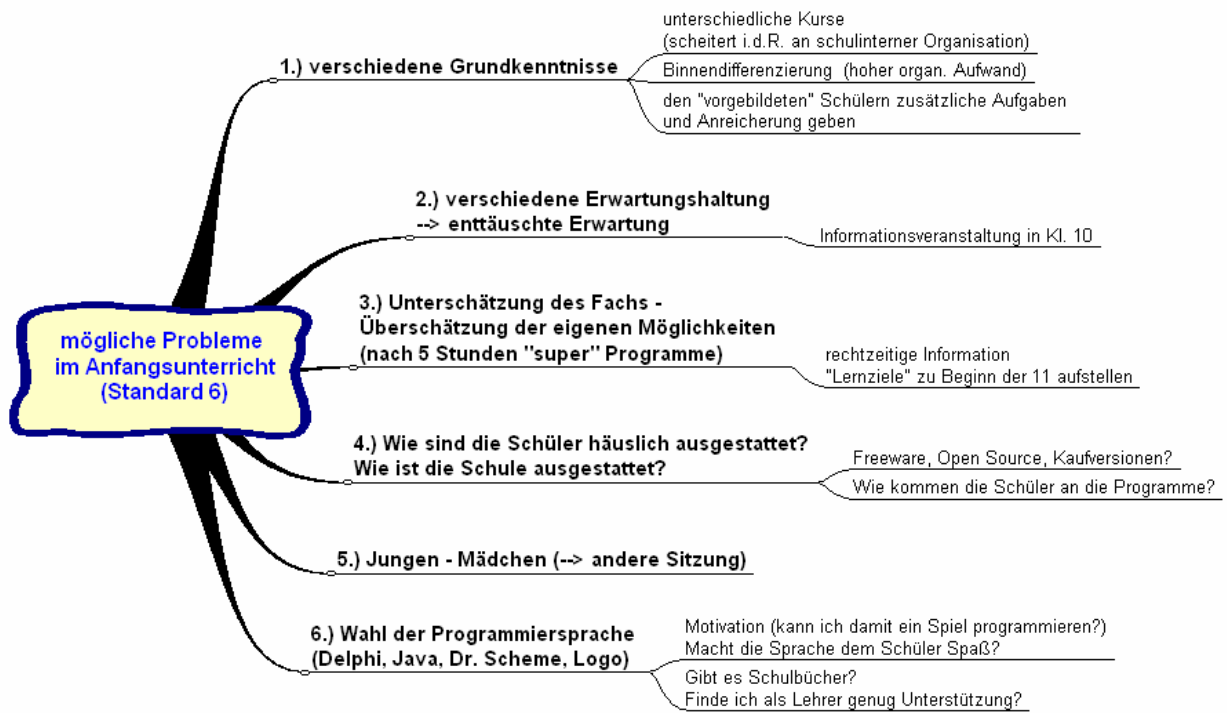


Ergebnisse der Seminarsitzung vom 03. September 2007



Zusammenfassung aus Sekundärquellen:

1. Zugang über die Textverarbeitung

- Funktionsweise eines Textverarbeitungssystems
- Kommunikation mit Textverarbeitungssystemen (Menü- und Kommandosteuerung)
- Grundlegende Arbeitstechniken (Texte eingeben/laden, Bewegen im Text, Markieren, Speichern und Drucken usw.)
- Fortgeschrittene Arbeitstechniken (stilistische Gestaltung, Fehlerkorrektur, Einbindung anderer Objekte usw.)
- gesellschaftliche Auswirkungen von Computern

Vorteile

- entspricht meist den Wünschen der S. mit geringen Vorkenntnissen
- systematische Gliederung des Unterrichts
- eigenständiges, projektartiges Arbeiten ist möglich
- zu behandelnde Probleme sind meist von einer praktischen Relevanz
- angenehm zu unterrichten für Lehrer mit unzureichender Ausbildung
- vorzeigbare Ergebnisse entstehen

Gefahren

- löst nur bedingt das Problem des Anfangsunterrichts (unterschiedliche Vorkenntnisse),
- demotivierend für S. mit guten Vorkenntnissen
- liefert kein umfassendes Bild der Informatik
- zeigt nur eine Anwendung der Informatik
- bietet begrenzte Möglichkeiten für gesellschaftliche Aspekte

2. Programmiersprachlicher Zugang

- Prinzip 'Anfangsunterricht = Programmierkurs'
- meist durch schrittweise **Einführung** einer imperativen Programmiersprache beginnend mit elementaren werden immer komplexere Sprachkonstruktionen der jeweiligen Programmiersprache an fiktiven (meist sehr einfachen) Problemstellungen eingeführt

Vorteile

- entspricht meist den Wünschen der S. mit guten Vorkenntnissen
- Erwerb von Spezialkenntnissen
- systematische Gliederung des Unterrichts
- schrittweise Einführung entspricht dem gewohnten Unterricht (z.B. in Fremdsprachen)
- angenehm zu unterrichten für Lehrer mit unzureichender Ausbildung

Gefahren

- löst nicht das Problem des Anfangsunterrichts (unterschiedliche Vorkenntnisse), verstärkt es eher
- zu behandelnde Probleme sind meist von geringer praktischer Relevanz
- demotivierend für S., da erst 'auf Vorrat' gelernt wird und kaum vorzeigbare Ergebnisse entstehen
- liefert kein umfassendes Bild der Informatik (eher den falschen Eindruck: Informatik = Programmieren)
- bietet kaum Möglichkeiten für gesellschaftliche Aspekte

3. Systemanalytischer Zugang

ausgehend von einem komplexen Softwaresystem (z.B. ein Spiel, das in Pascal programmiert wurde) erlangen die Schüler über die Phasen:

- **Blick auf das System** → Computerbedienung, Struktogramm über den Programmverlauf
- **Blick in das System** → Bedeutung bestimmter Sprachelemente (Prozedur, Funktion)
- **Modifikation des Systems** → Programm wird geringfügig ergänzt und aktuellen Anforderungen angepaßt, erste Programmiererfahrungen

- **Konstruktion des Systems** → unter Verwendung der vorhandenen Bauelemente wird ein Programm zu einem ähnlichen Problem erarbeitet

Vorteile des Ansatzes

- Informatiknah, führt zu verständnisvoller Nutzung und Konstruktion von Informatiksystemen
- Fächerübergreifende Anwendung
- Wissenserwerb mit Hilfe des Spiralprinzips

Gefahren

- ist intellektuell anspruchsvoll
- erhebliche Vorarbeiten für den Lehrer (bezogen auf eine analyse-freundliche Aufbereitung und Strukturierung des Programms)
- löst das Problem der unterschiedlichen Vorkenntnisse nicht

4. Zugang über Lern- und Programmierumgebungen

- durch Verwendung leistungsfähiger, kommerzieller, objektorientierter Entwicklungsumgebungen (Hypertextsysteme z.B. ToolBook und Programmiermodelle, z.B. LOGO) sammeln die Schüler erste Erfahrungen mit Informatiksystemen
- Problemstellungen und Operationen orientieren sich an realen Vorlagen des Alltags
- Programmieren reduziert sich auf das Konfigurieren der vorhandenen Umgebung auf persönliche Bedürfnisse

Vorteile des Ansatzes

- orientiert sich an den kognitiven Voraussetzungen von Anfängern
- mit geringen Vorkenntnissen lassen sich leistungsfähige Produkte konstruieren
- realisiert einen stufenweisen Einstieg in die Programmierung
- vereinigt Entwicklungsrichtungen wie Objektorientiertheit und Multimedia

Nachteile des Ansatzes

- beschränkt die Tiefe der zu gewinnenden Informatikkenntnisse
- ein Umstieg auf 'richtige' Programmiersprachen ist notwendig, da z.B. Datentypen wie Listen, Bäume, Records fehlen
- löst das Problem unterschiedlicher Vorkenntnisse nur teilweise

5. Projektorientierter fächerübergreifender Zugang

- ein komplexes fächerübergreifendes Projekt wird innerhalb einer Projektwoche bzw. eines Kurshalbjahres behandelt
- Aufgabenstellungen werden durch die Schüler mitbestimmt und knüpfen an ihren Erlebnis- und Erfahrungsbereich an
- denkbare fächerübergreifende Verbindungen wären z.B. Mathematik / Physik / Informatik bzw. Politik / Informatik

Vorteile des Ansatzes

- spricht alle Schüler an
- fördert außerinformatische Kompetenzen
- löst Problem der unterschiedlichen Vorkenntnisse
- Arbeitsformen wie Team-, Gruppenarbeit fördern soziale Kompetenzen und demonstrieren zentrale Arbeitsweisen von Informatikern
- fördert fächerübergreifende Bildung

Nachteile des Ansatzes

- informatische Elemente des Projekts bilden oft nur einen geringen Anteil
- mehr Nutzung der Informatik als Verständnis von informatischen Systeme

Quellen:

- 1.) *Schwill, Andreas und Sigrid Schubert. Didaktik der Informatik. Berlin 2004. S. 277ff.*
- 2.) <http://www.informatikdidaktik.de/Projekt/anfang3.html>