

Telecooperation/RGB

Grundlagen der Informatik I

Thema 0: Einführung

Prof. Dr. Max Mühlhäuser
Dr. Guido Rößling

Copyrighted material; for TUD student use only



Worum es in der Informatik nicht geht

Um den Computer!

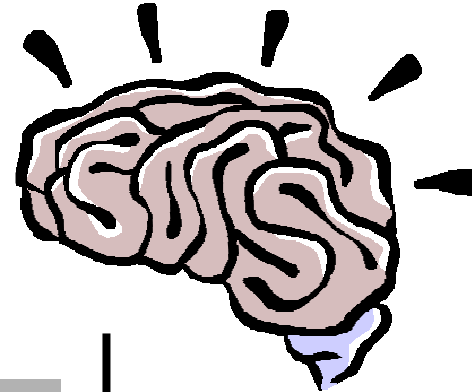


Genauso wie es in der Biologie nicht um Mikroskope
und in der Literatur nicht um Kugelschreiber geht...



Worum geht es bei Gdl-1?

Problem und
Beschreibung
mentales Model



Herausforderung: Die Lösung so zu formulieren, dass sie für die Maschine ausführbar, aber auch für den Menschen verständlich ist

völlig
unterschiedliche
Abstraktionsebenen

ausführbares
Programm
Maschinenebene

```
01010110110
01101001011
10010110100
10111011110
00010011110
10001001010
```



Die Rolle von Gdl-1

- Einführung in die Programmierung -

- Inhaltliche Randbedingungen
- Soziologische Randbedingungen
- Zeitliche Randbedingungen



Die Rolle von Gdl 1

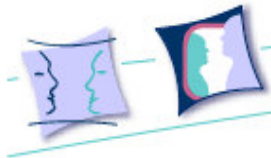
**Eltern: “N-TV
berichtet viel über Java.
Wieso unterrichten Sie
meinen Sohn in
Scheme?”**

**Student: “Ich weiß
alles und Perl
ist das Beste”**

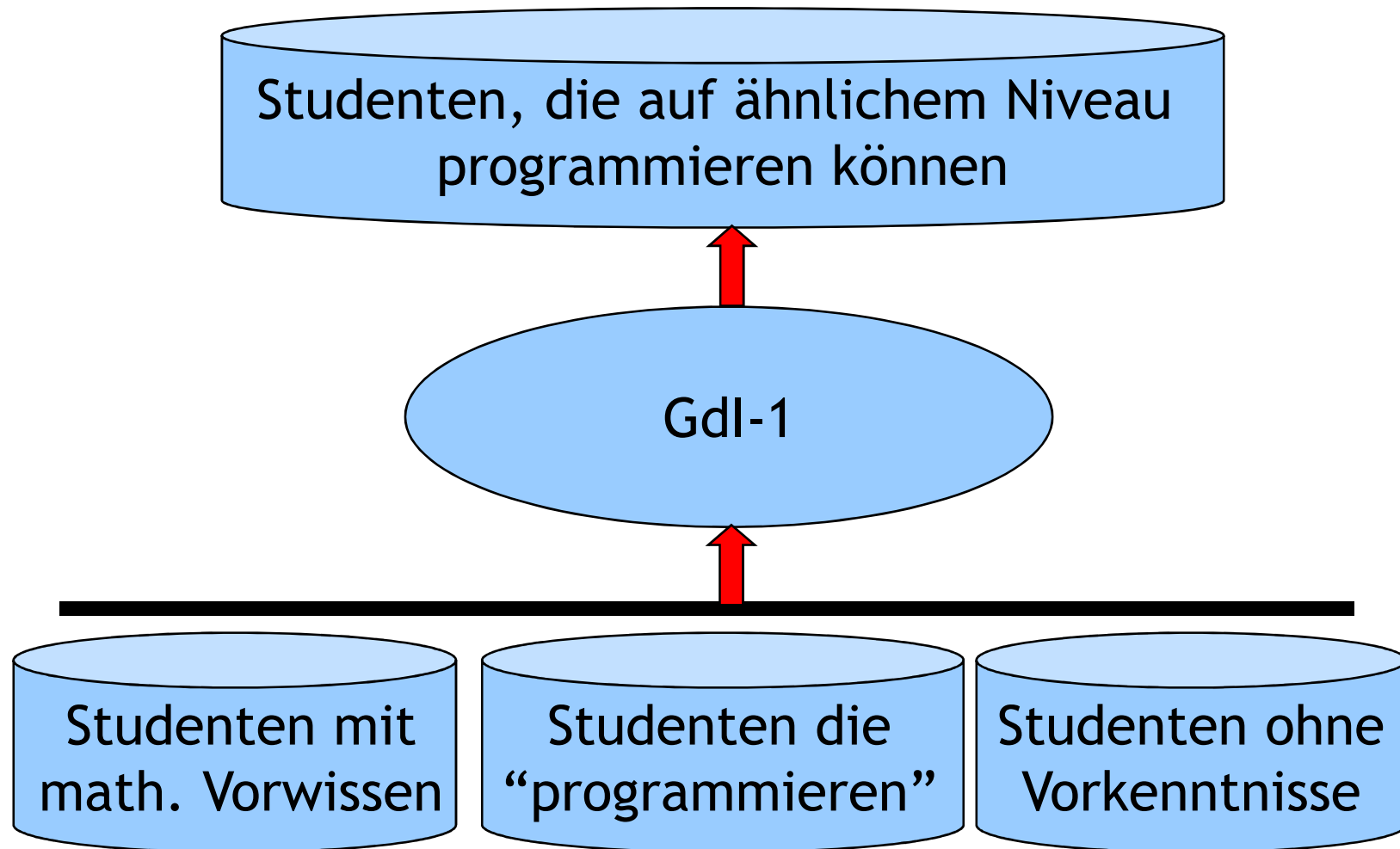
**Gdl 1 Einführung
in die
Programmierung**

**Industrie: “Heute
benutzen wir
C# v4.538. Wieso
bringen Sie es Ihren
Studenten nicht bei?”**

**Andere Profs:
“Jeder in meiner
Welt braucht
Pointer. *Das*
sollten sie in
Gdl 1 lernen!”**



Die Rolle von Gdl-1





In vielen Gdl-1 Kursen...

- Klassisch:
 - Ihnen wird die Syntax einer Programmiersprache beigebracht, die gerade in Mode ist
 - Sie benutzen eine kommerzielle Entwicklungsumgebung
 - Sie bekommen Codebeispiele und sollen diese imitieren
 - Sie diskutieren einige algorithmische Ideen
- Die genaue Bedeutung der Programmiersprache bleibt im Dunkeln
 - Oberflächliche Erklärungen
 - Erklärungen anhand von Low-Level Hardware Details
- Wie **Sie selber** völlig neue Programme schreiben, ist und bleibt unklar



Frickeln vs. Design

Wozu führt dieser Ansatz?

- Syntax: Frickeln bis es funktioniert
- Design: ebenfalls frickeln bis es funktioniert
- Sie kennen zwar ein paar Standardalgorithmen, aber dies ersetzt kein Vorgehensmodell für Programmdesign
- Ihr Verständnis von der Bedeutung der Programmiersprache ist eine Mischung aus Bauchgefühl und Halbwissen über Low-Level-Implementationsdetails



Arbeitsumgebung von Profi-Piloten



Gut geeignet für die erste Flugstunde?



Arbeitsumgebung eines Software-Profis

Java - L2Object.java - Eclipse Platform

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer: Hierarchy: L2Object

Outline: vcl1, import declarations, L2Object 1.1 (ASCII-1kv), mixins: LexicalPath[], outer: Addr, L2Object(LexicalPath[], Addr), fields: HashMap, setField(String, Value), getField(String), toString()

```
HashMap fields = new HashMap();
public void setField(String name, Value value) {
    fields.put(name, value);
}
public Value getField(String name) {
    return (Value) fields.get(name);
}
public String toString() {
    StringBuffer s = new StringBuffer("[[");
    for (int i=0; i< mixins.length; i++) s.append(mixins[i].
    s.append("|");
    s.append(outer.toString());
    s.append("|");
    Iterator it = fields.entrySet().iterator();
    while (it.hasNext()) {
        Map.Entry e = (Map.Entry) it.next();
        s.append(e.getKey() + "=" + e.getValue() + ",");
    }
    s.append("]]");
    return s.toString();
}
```

Problems: 79 errors, 21 warnings, 0 infos (Filter matched 100 of 238 items)

Description	Resource	In Folder	Location
The declared package does not match the expected pac...	Addr.java	JavaSandbox/gamma	line 1
The declared package does not match the expected pac...	Advice.java	JavaSandbox/gamma	line 7
The declared package does not match the expected pac...	Assignment.java	JavaSandbox/gamma	line 1
The declared package does not match the expected pac...	ASTElement.java	JavaSandbox/gamma	line 1
The declared package does not match the expected pac...	ASTVisitor.java	JavaSandbox/gamma	line 1
The declared package does not match the expected pac...	BinExpr.java	JavaSandbox/gamma	line 1

Gut geeignet für die ersten Programme?

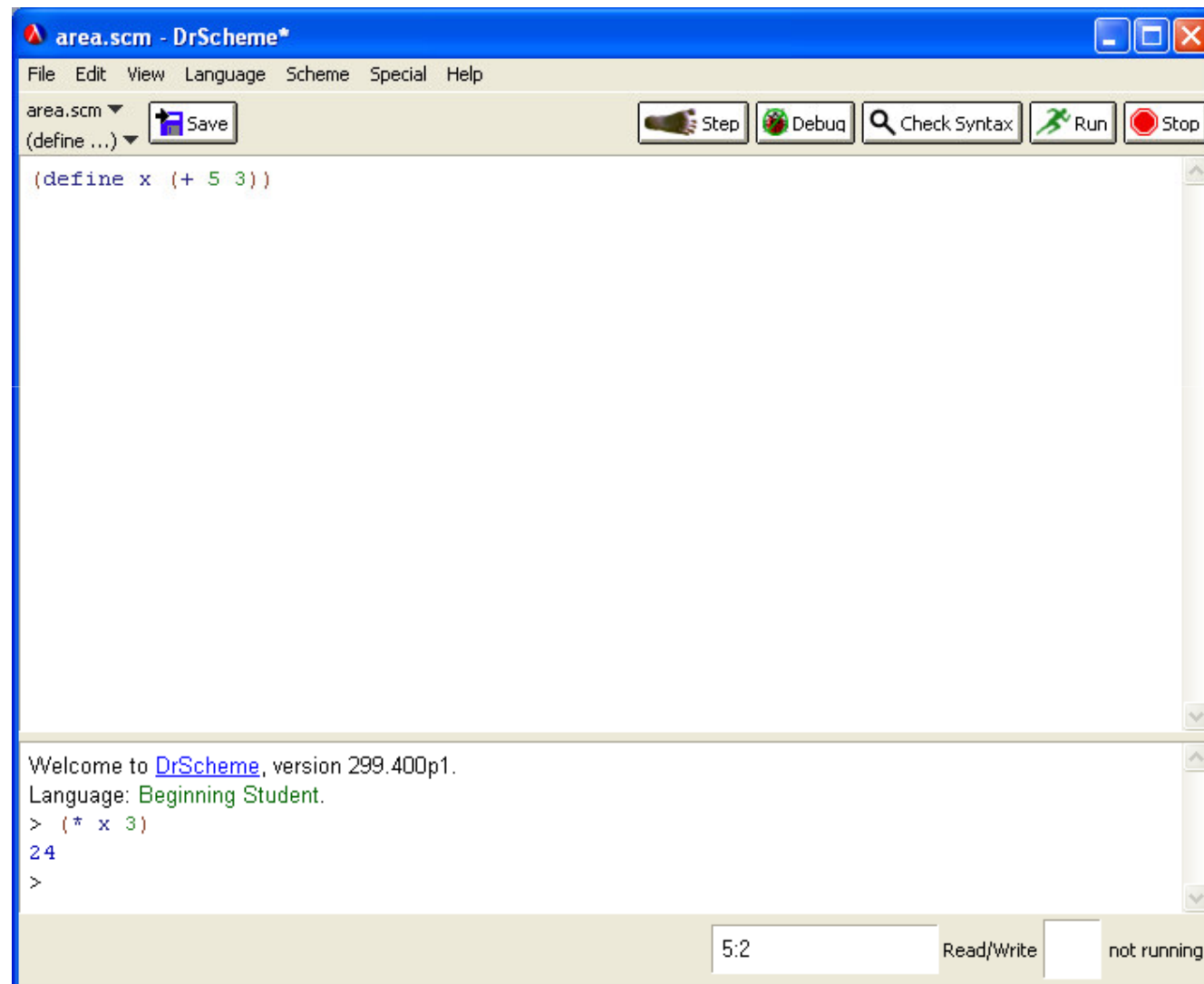


Unser Ansatz: 4 Säulen

1. Entwicklungsumgebung für Anfänger
2. Einfache Programmiersprache
3. Präzise Modelle, um die Bedeutung der Programmiersprache zu verstehen
4. Leitfaden, wie Sie neue Programme entwickeln sollten



Entwicklungsumgebung für Anfänger





Einfache Programmiersprache

- Sie werden nicht eine, sondern viele abgestufte Programmiersprachen kennen lernen
 - Unterschiedliche Level, nach *pädagogischen* Gesichtspunkten angeordnet
 - Teilmengen der Sprache *Scheme*
 - Benutzung von Konstrukten, die Sie noch nicht kennen, wird unterbunden
 - Keine bizarren Fehlermeldungen, mit denen Sie nichts anfangen können
- Wir können uns auf die Designkonzepte und Ideen, die Ihrem jeweiligen Wissensstand angemessen sind, konzentrieren!



Präzise Modelle für die Programmiersprache

Auf jedem Sprachlevel werden wir Ihnen ein Modell vorstellen, anhand dessen Sie die Bedeutung ihrer Programme nachvollziehen können:

- Sie können Programme, wenn sie wollen, auf einem Blatt Papier “ausführen”.
- Diese Modelle orientieren sich nicht an der Hardware, sondern daran, wie sie am einfachsten die **Bedeutung** verstehen können.
- Sie werden auch Modelle kennen lernen, die illustrieren, wie eine Programmiersprache auf typischer Hardware implementiert ist.



Leitfaden zur Entwicklung von Programmen

- Sie bekommen nicht einfach Programme vorgesetzt
- Sie bekommen “Rezepte”, wie Sie Schritt für Schritt selber ein Programm entwerfen sollten
 - Diese Rezepte sind fast unabhängig von der Programmiersprache, die Sie benutzen



Typischer Ansatz für Programm-“Design”

So werden wir es **nicht** machen... ☺





Von Scheme nach Java

- In der 2. Hälfte des Semesters wird Java verwendet
- Die Designrezepte funktionieren Schritt für Schritt auch für objektorientierte Programme
- Viele Unterschiede sind lediglich Notation
- Das Studium fundamentaler Unterschiede ist sehr instruktiv



Wieso nicht gleich Java?

- Komplexe Notation, komplexe Fehler
- Keine Entwicklungsumgebung unterstützt Sprachlevel
- Direkte Interaktion mit dem Interpreter macht den “Start” einfacher
- Designrezepte ertrinken in umständlicher Syntax
 - K.I.S.S. - Keep It Simple Syntactically
 - Java/C++/Pascal: **90% Syntax**, 10% Problemlösung
 - Scheme: 10% Syntax, **90% Problemlösung**
- Scheme (bzw. Lisp) ist die “Mutter” der meisten modernen Sprachen
 - Sprachfeatures können in Scheme in Reinform studiert werden
 - Viele “neue” Features moderner Sprachen können sehr leicht verstanden werden, wenn man Scheme kennt
- Untersuchungen zeigen, dass nach unserem Ansatz ausgebildete Studenten **die besseren OO-Programmierer** werden!



Vorteile einer einfachen Sprache

Wir lassen den Interpreter die Summe von 2 und 3 berechnen

DrScheme:

Wir tippen das Programm (+ 2 3) ein und bekommen als Resultat "5"

Die in Java/Eclipse benötigten Schritte sind nicht "Unsinn", sondern haben alle ihre Berechtigung.

Für einen **Anfänger** sind sie jedoch verwirrend und unnötig!

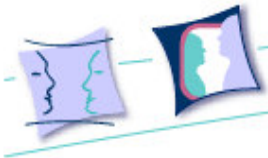
Java/Eclipse:

Wir starten ein neues **Projekt**
Wir legen ein neues **Package** an
Wir legen eine neue **Klasse** an
Wir schreiben folgendes Programm:

```
package test;  
public class Test {  
    public static void main(String[] args) {  
        System.out.println(2+3);  
    }  
}
```

(Sie brauchen Wochen, bis Sie alle Teile dieses Programms verstanden haben)

Wir **kompilieren** das Programm
Wir starten das Programm
Im Ausgabefenster erscheint das Resultat "5"

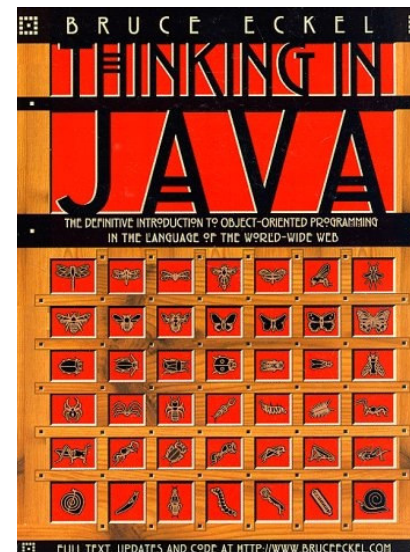
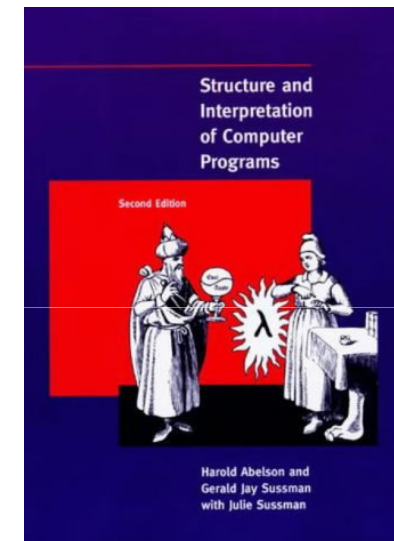
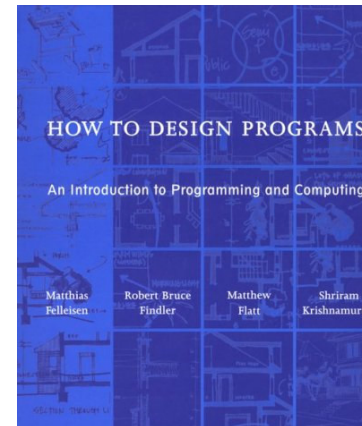


Fazit

- Die Ausbildung guter Programmierer heißt nicht, dass man mit Werkzeugen für professionelle Programmierer beginnt, die gerade in Mode sind
- Bei uns bekommen Sie zunächst eine fundierte, rigorose Basis der Grundkonzepte der Programmierung
- **Dann, und erst dann**, werden wir sie auch mit der neuesten Mode vertraut machen!

Bücher

- Scheme
 - Wichtigstes Buch: How To Design Programs (HTDP)
 - Structure and Interpretation of Computer Programs (SICP)
- Java
 - Thinking in Java



Alle Bücher umsonst online verfügbar!



Weitere Infos

- Alle Neuigkeiten, Links, Unterlagen etc. finden Sie in der Lernplattform

<http://proffs.tk.informatik.tu-darmstadt.de/teaching>