Standards – Unterricht – Prüfungen

Plenarvortrag auf dem 4. Magdeburger Lehrertag
(gekürzt)

Prof. Dr. Michael Fothe
Casio-Stiftungsprofessur
für Didaktik der Informatik/Mathematik
Friedrich-Schiller-Universität Jena
fothe@minet.uni-jena.de

Uni Magdeburg / 5. März 2008

Standards – Unterricht – Prüfungen

Bildungsstandards hätte Wort des Jahres 2003 oder 2004 werden können.

Es gibt sie bereits führ mehrere Schularten, Fächer und Klassenstufen.

Mit ihnen beginnt sich die Schullandschaft zu verändern.

Ausgangssituation (2003/2004)

- Deutsche Schüler sind im internationalen Vergleich eher mittelmäßig.
- Lehrpläne sind überfrachtet und unklar.
- Unterricht ist zwischen Schulen nicht vergleichbar.
- Der Wert der Bildungsabschlüsse ist uneinheitlich.
- Medien befassen sich verstärkt mit dem Thema.

> Man streitet sich endlos um Schulform und Betreuungszeiten,
> aber nie um das Entscheidende: Wie gut ist der Unterricht?
> Das Klassenzimmer ist eine Art Black Box,
> in der auf wundersame Weise Unterricht geschieht.
>
> DER SPIEGEL 46/2003, S. 48

Schulpolitische Reaktionen

- Nationale Bildungsstandards werden erarbeitet und eingeführt.
- Das Institut für Qualitätsentwicklung im Bildungswesen wird aufgebaut.
- Standards für die Lehrerbildung der 1., 2. und 3. Phase werden entwickelt.

Bildungsstandards Informatik für die SI

Von 2004 bis 2007 entwickelten Fachdidaktiker und Schulpraktiker Bildungsstandards Informatik für die SI.

Vorstellen der Entwurfsfassung auf der INFOS 2007

Beschluss des GI-Präsidiums am 24. Januar 2008

- GI-Fachausschuss „Informatische Bildung in Schulen“
- GI-Fachgruppe „Didaktik der Informatik“
- Landes-Fachgruppen
- Fachdidaktische Gespräche Königstein
- Fachzeitschrift LOG IN
- Federführung: H. Puhlmann, Altdorf

Klassenstufen 7, 10

Mindeststandards, gültig für alle Schularten

Grundlage: Pflichtfach Informatik (1 Wochenstunde in den Kl. 5-10)

Bildungsstandards Informatik für die SI

Inhaltsbereiche (Themenfelder):

- Information und Daten
- Algorithmen
- Sprachen und Automaten
- Informatiksysteme
- Informatik, Mensch und Gesellschaft

Prozessbereiche (Arten des Arbeitens mit informatischen Inhalten):

- Modellieren und Implementieren
- Begründen und Bewerten
- Strukturieren und Vernetzen
- Kommunizieren und Kooperieren
- Darstellen und Interpretieren

Bildungsstandards Informatik für die SI

- Standards sollen sich durch Klarheit, Knappheit und Anspruch auszeichnen.
- Die Arbeit von Schulen kann auf der Grundlage der nationalen Bildungsstandards verantwortlicher, abrechenbarer und transparenter werden.
- Den Lehrerinnen und Lehrer der 11./12. Klassen, den Universitäten und berufsbildenden Schulen wird künftig klarer sein, auf welchem Wissen und Können sie aufbauen können.

Memorandum zur Schulinformatik

Digitale Spaltung verhindern – Schulinformatik stärken!

- Einführung eines durchgängigen Pflichtfaches Informatik in der Sekundarstufe I an allen allgemein bildenden Schulen aller Bundesländer
- Verankerung der Informatik in der gymnasialen Oberstufe
- Zulassung von Informatik als vollwertiges Prüfungsfach in allen Abschlussprüfungen an Schulen
- Erteilung von Unterricht im Fach Informatik nur durch ausgebildete oder entsprechend weitergebildete Lehrkräfte

➢ GI-Fachausschuss „Informatische Bildung in Schulen“,
Beschluss des Präsidiums der Gesellschaft für Informatik e. V. vom 21. September 2004

Nachwuchs für die Informationsgesellschaft! Plädoyer für eine zukunftsorientierte Schulbildung

- Jede Schülerin und jeder Schüler einer allgemeinbildenden Schule erhält eine fundierte und breite Allgemeinbildung auch auf dem Gebiet der Informatik. So wird in angemessener Weise auf Berufsausbildung bzw. Studium vorbereitet.
- An allgemeinbildenden Schulen ist mindestens ein Drittel der Unterrichtsstunden für Mathematik, Informatik, Naturwissenschaft und Technik vorzusehen.
- In der Sekundarstufe I wird Informatik als Pflichtfach mit durchschnittlich einer Wochenstunde pro Schuljahr in allen Schulformen eingerichtet. Länder, die dies bereits realisiert haben, können als Beispiele für die Integration dieses Pflichtfachs in die Stundentafel dienen.
- In der gymnasialen Oberstufe sind die Fächer Biologie, Chemie, Informatik und Physik gleichwertig anzubieten und mindestens zwei dieser vier Fächer bis zum Abitur zu belegen.
- Biologie, Chemie, Informatik und Physik sind in der Wahl der Prüfungsfächer in der Abiturprüfung gleichwertig. GI/BITKOM 18. September 2007

Nächste Arbeitsschritte

- Die Bildungsstandards Informatik sind dort zu implementieren, wo immer dies geht.
- Aufgaben und Kompetenzmodelle sind zu entwickeln und zu erproben.

Stufen / Kompo-nenten		Stufe I **Die Schülerinnen und Schüler haben grundlegende Kompetenzen zu Algorithmen.** **Die Schülerinnen und Schüler…**	Stufe II **Die Schülerinnen und Schüler haben vertiefte Kompetenzen zu Algorithmen.** **Die Schülerinnen und Schüler…**	Stufe III **Die Schülerinnen und Schüler haben umfassendere Kompetenzen zu Algorithmen.** **Die Schülerinnen und Schüler…**
A	**Eigenschaften von Algorithmen**	-erklären den Algorithmusbegriff und die wesentlichen Eigenschaften von Algorithmen -überprüfen die wesentlichen Eigenschaften von Algorithmen in einfachen Fällen -nennen Probleme, die mithilfe von Algorithmen lösbar bzw. nicht lösbar sind	-erklären den Algorithmusbegriff und die wesentlichen Eigenschaften von Algorithmen an bekannten Beispielen -begründen anhand dieser Eigenschaften, ob gegebene Handlungsabläufe Algorithmen sind -nennen Probleme, die mithilfe von Algorithmen lösbar bzw. nicht lösbar sind	-erklären den Algorithmusbegriff und die wesentlichen Eigenschaften von Algorithmen an selbst konstruierten Beispielen -begründen anhand dieser Eigenschaften, ob gegebene Handlungsabläufe Algorithmen sind -nennen Probleme, die mithilfe von Algorithmen lösbar bzw. nicht lösbar sind
B	**Algorithmische Grundbausteine und Datentypen**	-erklären die algorithmischen Grundbausteine wie Variablen, Wertzuweisungen, Verzweigungen und Wiederholungen und wenden diese Erklärungen an -stellen die algorithmischen Grundbausteine als Pseudocode dar -verwenden einen numerischen Datentyp	-erklären die algorithmischen Grundbausteine wie Variablen, Wertzuweisungen, Verzweigungen und Wiederholungen und wenden diese Erklärungen an -stellen die algorithmischen Grundbausteine in verschiedenen Darstellungsformen dar -verwenden verschiedene Datentypen	-erklären die algorithmischen Grundbausteine wie Variablen, Wertzuweisungen, Verzweigungen, Wiederholungen und Unterprogramme mit Parametern und wenden diese Erklärungen an -stellen die algorithmischen Grundbausteine in verschiedenen Darstellungsformen dar und wechseln zwischen Darstellungsformen -verwenden verschiedene Datentypen
C	**Arbeit mit Algorithmen**	-lesen in Pseudocode gegebene einfache Algorithmen -prüfen schrittweise einfache Algorithmen mit gegebenen Beispielen -setzen gegebene einfache Algorithmen in Programme um -modifizieren und ergänzen einfache Algorithmen bzw. Programme nach	-analysieren die Funktionsweise und den Leistungsumfang gegebener Algorithmen -prüfen Algorithmen mit gegebenen Beispielen mithilfe von Durchlauftabellen (Schreibtischtest) -setzen gegebene Algorithmen in Programme um -modifizieren und ergänzen Algorithmen bzw. Programme nach Vorgaben	-analysieren die Funktionsweise und den Leis-tungs-umfang gegebener komplexer Algorithmen -prüfen Algorithmen mithilfe von Durchlauftabellen (Schreibtischtest) und wählen dazu typische und untypische Beispiele selbst aus -setzen gegebene komplexe Algorithmen in Programme um …aben und …men
D	**Programmentwicklung**	-entwerfen einfache Programme skizzenhaft -implementieren einfache Programme mit einem Programmiersystem -testen einfache Programme anhand gegebener Eingaben auf ihre Grundfunktionalität	-fertigen einen schriftlichen Entwurf für Programme an -implementieren Programme mit einem Pro-grammiersystem benutzungsfreundlich -testen Programme anhand gegebener Einga-ben auf ihre Funktionalität -reflektieren über den Lösungsweg	-fertigen einen schriftlichen Entwurf für komplexe Programme an -implementieren komplexe Programme mit einem Programmiersystem benutzungsfreundlich -testen komplexe Programme anhand selbst gewählter Eingaben auf ihre Funktionalität -reflektieren über den Lösungsweg sowie über

L. Kohl / M. Fothe in: LOG IN Heft Nr. 146/147 (2007), S. 20-22

Nächste Arbeitsschritte

- Lehrpersonen sind zu befähigen, Bildungsstandards bei der Konzeption, Durchführung und Reflexion ihres Unterrichts gewinnbringend einzusetzen.

Ziel ist das Entwickeln einer Kultur des sinnvollen Umgangs mit Bildungsstandards Informatik.

Nächste Arbeitsschritte

- Eine wichtige Aufgabe der Schulen wird darin bestehen, die ergebnisorientierten Standards in prozessorientierte Planungen auf unterschiedlichen Ebenen umzusetzen.
- Wie können Inhalte und Prozesse gleichermaßen berücksichtigt werden?
- Welche Rolle besitzen künftig Lehrpläne?

Nächste Arbeitsschritte

- Ein Thema muss nicht nur gut unterrichtet sein, die Schüler müssen auch etwas dauerhaft gelernt haben.
- Was muss ein leistungsschwacher Schüler wissen und können, um erfolgreich zu sein?
- Die vorhandene Ungleichheit zwischen Schülern ist mehr als bisher produktiv zu nutzen.

Nächste Arbeitsschritte

- Offen ist, wie mit der unterschiedlichen Qualität von Schulen umzugehen ist.
- Wie kann verhindert werden, dass die Kulturhoheit der 16 Länder durch die Kulturhoheit der 10.000 deutschen Schulen ersetzt wird?
- Schule ist mehr, als das Erfüllen von Standards.

Bildungsstandards Informatik für die SII

Informatik ist an deutschen Gymnasien in der gymnasialen Oberstufe als Grundkursfach und teilweise auch als Leistungskursfach verankert.

Folgerichtig gibt es EPA für Informatik, in denen die Prüfungs-anforderungen und damit die Kompetenzen beschrieben werden, über die die Abiturienten am Ende ihrer Schulzeit verfügen sollen.

Einordnung der EPA Informatik

- Fachlichkeit
- Fokussierung
- Kumulativität
- Verbindlichkeit für alle
- Differenzierung
- Verständlichkeit
- Realisierbarkeit

- Klieme-Expertise
- M. Fothe: LOG IN Heft Nr. 135 (2005), S. 46-49

Kompetenzbereiche

- Erwerb und Strukturierung informatischer Kenntnisse
- Kennen und Anwenden informatischer Methoden
- Kommunizieren und Kooperieren
- Anwenden informatischer Kenntnisse, Bewerten von Sachverhalten und Reflexion von Zusammenhängen

Kompetenzbegriff (nach Weinert)

Kompetenzen sind die bei Individuen verfügbaren oder von ihnen erlernbaren kognitiven Fähigkeiten und Fertigkeiten, bestimmte Probleme zu lösen, sowie die damit verbundenen …Bereitschaften und Fähigkeiten, die Problemlösungen in variablen Situationen erfolgreich und verantwortungsvoll nutzen zu können.

Lern- und Prüfungsbereiche

- Grundlegende Modellierungstechniken
  - Objektorientierte Modellierung
  - Datenmodellierung
  - Zustandsorientierte Modellierung
  - Modellierung von Abläufen mit Algorithmen
  - Funktionale Modellierung
  - Regelbasierte Modellierung

  (mindestens zwei im Grundkurs- und mindestens drei im Leistungskursfach)

- Interaktion mit und von Informatiksystemen
- Möglichkeiten und Grenzen informatischer Verfahren

Anforderungsbereiche

Die EPA Informatik legen fest, welchen Schwierigkeits- und Komplexitätsgrad Abituraufgaben haben sollen. Diese Festlegungen besitzen eine längerfristige Gültigkeit.

Das Schwergewicht der zu erbringenden Prüfungsleistungen liegt im Anforderungsbereich II.
Daneben sind die Anforderungsbereiche I und III zu berücksichtigen, und zwar Anforderungsbereich I in höherem Maße als Anforderungs-bereich III.

Anforderungsbereich I

- Wiedergeben eines bekannten Modells in geübter Darstellung
- Identifizieren von Objekten und ihren Beziehungen in einem bekannten Sachzusammenhang
- Beschreiben von Daten- und Kontrollstrukturen
- Verwenden einfacher Modellierungen und bekannter einfacher Algorithmen

Anforderungsbereich II

- Durchführen einer objektorientierten Analyse und Entwickeln eines objektorientierten Designs für eine vergleichbare neue Problem-stellung
- Erstellen eines ER-Diagramms für eine vergleichbare neue Problemstellung
- Implementieren von Prozeduren, Funktionen und Methoden im vorgegebenen Kontext
- Begründen von bestimmten Eigenschaften (z. B. Terminierung, Zeit- und Speicheraufwand) eines gegebenen Algorithmus durch nicht formale Überlegungen

Anforderungsbereich III

- Durchführen einer komplexen Problemanalyse
- Formulieren einer begründeten Stellungnahme zu einem authentischen Text in Bezug auf Möglichkeiten, Angemessenheit und Grenzen des Einsatzes von Informatiksystemen
- Beurteilen der eigenen Modellierung und Problemlösung im Anwendungskontext
- Entwickeln einer Sprache (z. B. Angabe der Syntax und Semantik einer einfachen Steuersprache für einen Roboter)

Weitere Schwerpunkte der EPA

In den EPA Informatik sind Festlegungen angegeben:

- zum Erarbeiten der Aufgaben,
- zum Formulieren des Erwartungshorizontes und
- zum Bewerten bei der schriftlichen Prüfung, der mündlichen Prüfung und bei der fünften Prüfungskomponente („mündliche Prüfung in neuer Form“).

Informatikunterricht heißt auch:

Hinter die Kulissen schauen.

Suche nach einem Muster

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
aber
```

Suche nach einem Muster

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
 aber
```

Suche nach einem Muster

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
  aber
```

Suche nach einem Muster

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
   aber
```

Suche nach einem Muster

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
    aber
```

Suche nach einem Muster

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
     aber
```

Suche nach einem Muster

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
      aber
```

Suche nach einem Muster

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
       aber
```

Suche nach einem Muster

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
        aber
```

Suche nach einem Muster

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
          aber
```

Suche nach einem Muster

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
          aber
```

Suche nach einem Muster

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
           aber
```

Suche nach einem Muster

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
            aber
```

Suche nach einem Muster

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
                                      aber
```

Suche nach einem Muster

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
                                      aber
```

Suche nach einem Muster

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
                                      aber
```

Anwendungen

Komprimieren von Daten

00000 → +
11111 → -

00111111100000000000101000000

Anwendungen

Komprimieren von Daten

00000 → +
11111 → -

00111111100000000000101000000

Anwendungen

Komprimieren von Daten

00000 → +
11111 → -

00-1100000000000101000000

Anwendungen

Komprimieren von Daten

00000 → +
11111 → -

00-11000000000010100000

Anwendungen

Komprimieren von Daten

00000 → +
11111 → -

00-11+00000101000000

Anwendungen

Komprimieren von Daten

00000 → +
11111 → -

00-11+00000101000000

Anwendungen

Komprimieren von Daten

00000 → +
11111 → -

00-11++101000000

Anwendungen

Komprimieren von Daten

00000 → +
11111 → -

00-11++101000000

Anwendungen

Komprimieren von Daten

00000 → +
11111 → -

00-11++101+0

Anwendungen

DNS-Sequenzierung

AACCTGATGGGTACTAAGTCGTAACCGTTGCATGTACGTTA

Enzym: A|TG

AACCTGA TGGGTACTAAGTCGTAACCGTTGCA TGTACGTTA

Anwendungen

Ausschließen von Schimpfwörtern in E-Mail-Adressen

Pech für alle Arschaks, Barschs, Darscheids, Marschalls und Zarschitzkys

Verfahren von Boyer-Moore

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
aber
```

Verfahren von Boyer-Moore

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
    aber
```

Verfahren von Boyer-Moore

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
     aber
```

Verfahren von Boyer-Moore

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
         aber
```

Verfahren von Boyer-Moore

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
            aber
```

Verfahren von Boyer-Moore

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
                 aber
```

Verfahren von Boyer-Moore

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
                      aber
```

Verfahren von Boyer-Moore

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
                      aber
```

Verfahren von Boyer-Moore

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
                           aber
```

Verfahren von Boyer-Moore

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
                               aber
```

Verfahren von Boyer-Moore

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
                                    aber
```

Verfahren von Boyer-Moore

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
                                     aber
```

Verfahren von Boyer-Moore

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
                                      aber
```

Verfahren von Boyer-Moore

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
                                      aber
```

Verfahren von Boyer-Moore

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
                                      aber
```

Verfahren von Boyer-Moore

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
                                      aber
```

Verfahren von Boyer-Moore

```
Text:  er sagte abrakadabra, es bewegte sich aber nichts
Muster: aber

er sagte abrakadabra, es bewegte sich aber nichts
                                      aber
```

Standards – Unterricht – Prüfungen

Wie sieht ein Informatikunterricht aus, der Bildungsstandards als Grundlage hat?

EPA-Anforderungsbereich II:

- Begründen von bestimmten Eigenschaften (z. B. Terminierung, Zeit- und Speicheraufwand) eines gegebenen Algorithmus durch nicht formale Überlegungen

Zeit- und Speicheraufwand von Algorithmen

- Eine wesentliche Eigenschaft von Algorithmen ist der funktionale Zusammenhang zwischen der Problemgröße und der benötigten Zeit.
- Die Schülerinnen und Schüler müssen erkennen, dass es nicht um einzelne Zeitmessungen geht und dass die Eigenschaft gleich bleibt, auch wenn der Computer schneller wird.
- Betrachtet werden meist nur die aufwändigen Operationen. Diese werden alle als gleich schnell angenommen.

➢ M. Fothe: Unterricht – bald nur noch mit Computer? In: informatica didactica, Ausgabe Nr. 7 (2006)

Naives Sortieren

- Nach und nach alle möglichen Reihenfolgen (Permutationen) der zu sortierenden Zahlen erzeugt und jedes Mal wird überprüft, ob die Zahlen in der sortierten Reihenfolge vorliegen.
- Ist dies der Fall, so wird das Erzeugen und Überprüfen beendet.
- Die Rechenzeit steigt sehr schnell an. Schon für kleine n dauert es ewig, bis das Ergebnis feststeht.
- Man ist geneigt, von Slowsort zu sprechen, denn es gibt n! Permutationen von n verschiedenen Zahlen, von denen genau eine die gesuchte ist.
- Im Mittel sind daher n!/2 Permutationen zu erzeugen und zu überprüfen

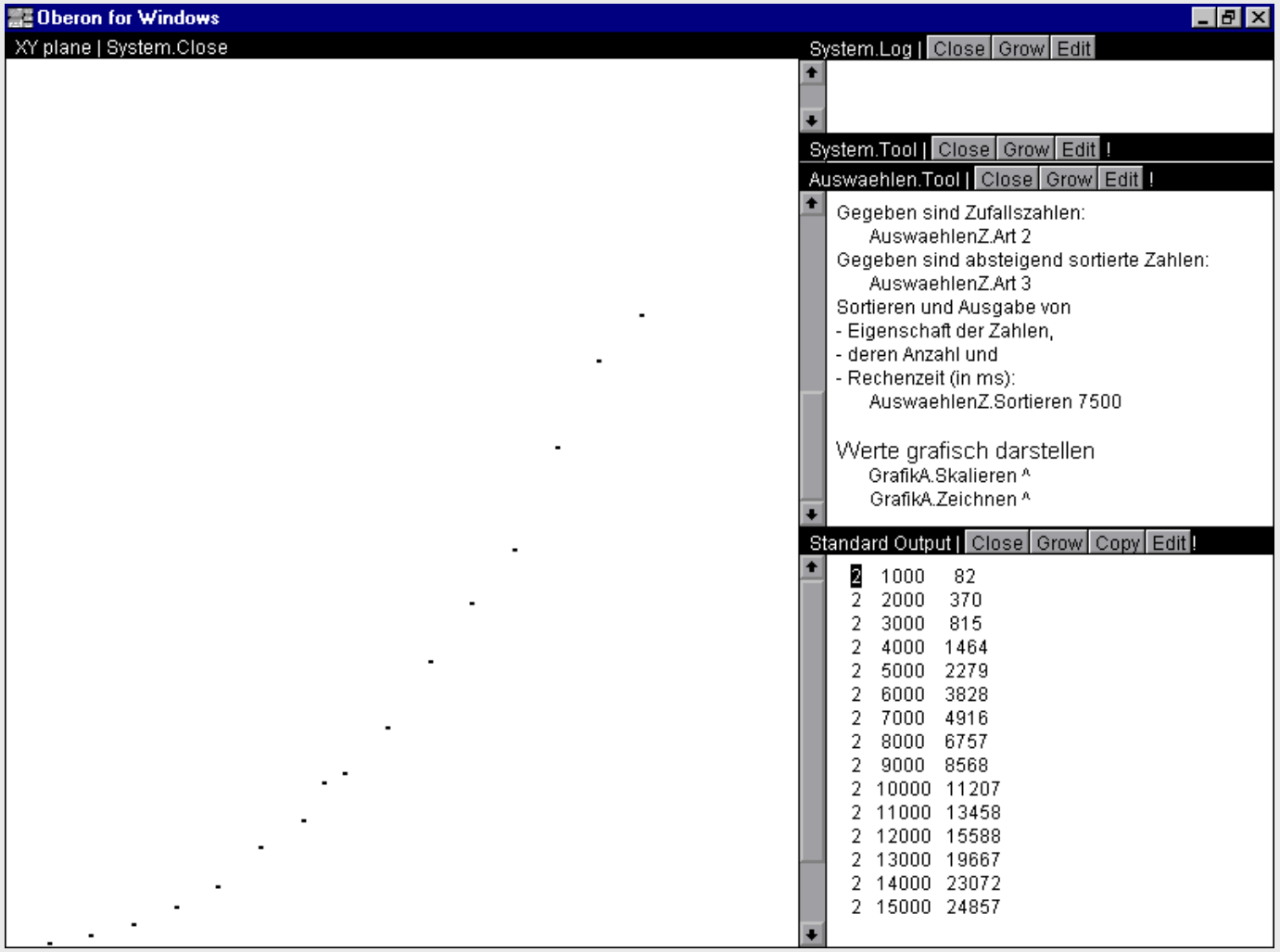
Sortieren durch Auswählen

```
PROCEDURE Auswahl*;
VAR r, s, t, x: INTEGER;
BEGIN
  FOR r := 1 TO n-1 DO
    t := r;  x := a[r];
    FOR s := r+1 TO n DO
      IF a[s] < x THEN
        t := s;  x := a[s]
      END
    END;
    a[t] := a[r];  a[r] := x
  END
END Auswahl;
```

Beim Untersuchen des Zeitverhaltens erfolgen als erstes Zeitmessungen. Dabei werden drei Eigenschaften der gegebenen Zahlen unterschieden:

1. Die Zahlen liegen bereits aufsteigend sortiert vor.
2. Die Zahlen sind Zufallszahlen.
3. Die Zahlen liegen falsch herum, also absteigend sortiert vor.

Vermutung: quadratischer Zeitaufwand



Analyse des Quelltextes

- Die Vermutung soll durch Analyse des Quelltextes bestätigt werden.
- Die Analyse des Quelltextes soll computerunterstützt erfolgen.
- In dem Programm zum Sortieren durch Auswählen wurden zu diesem Zweck zusätzliche Ausgabeanweisungen aufgenommen.
- Nach jedem Vergleich bzw. nach jeder Bewegung gibt das Programm ein Zeichen aus.
- Jedes Mal, wenn ein Element an die richtige Position gebracht wurde, wird eine neue Zeile begonnen.
- Veränderliche Parameter sind die Festlegung, ob Vergleiche oder Bewegungen ausgegeben werden sollen, die Anzahl zu sortierender Zahlen und die Eigenschaften dieser Zahlen.

Analyse des Quelltextes

Gegeben: 10 Zahlen absteigend sortiert

Ausgabe der Vergleiche

****
****
****
******
*****
****
***
**
*

Anzahl: 45

Ausgabe der Bewegungen

****
****
****
******
****
***
***
***
***

Anzahl: 52

Analyse des Quelltextes

Gegeben: 20 Zahlen absteigend sortiert

Ausgabe der Vergleiche

```
********************
*******************
******************
*****************
****************
***************
**************
*************
************
***********
**********
*********
********
*******
******
*****
****
***
**
*
```

Anzahl: 190

Ausgabe der Bewegungen

```
**********************
*********************
******************
*****************
***************
*************
**********
********
******
****
***
***
***
***
***
***
***
***
***
```

Anzahl: 157

Analyse des Quelltextes

Gegeben: 20 Zahlen absteigend sortiert

Ausgabe der Vergleiche

Anzahl: 190

Ausgabe der Bewegungen

Anzahl: 157

Analyse des Quelltextes

Gegeben: 20 Zahlen absteigend sortiert

Ausgabe der Vergleiche

Ausgabe der Bewegungen

$V = (n^2 - n) / 2$

$B \approx 3 * (n - 1) + n^2 / 4$

Gewonnene Erkenntnisse

- Die Anzahl der Vergleiche wächst stets quadratisch mit der Problemgröße.
- Sind die gegebenen Zahlen bereits aufsteigend sortiert, so wächst die Anzahl der Bewegungen linear.
- Sind die gegebenen Zahlen absteigend sortiert, so wächst die Anzahl der Bewegungen quadratisch.
- Werden Zufallszahlen sortiert, so wächst die Anzahl der Bewegungen zwischen linear und quadratisch.
- Die minimale Anzahl an Operationen ist dann auszuführen, wenn die gegebenen Zahlen bereits aufsteigend sortiert sind.
- Die maximale Anzahl an Operationen ist vermutlich dann auszuführen, wenn die gegebenen Zahlen absteigend sortiert sind.

Quicksort

- Aus der Folge von Elementen, die zu sortieren ist, wird ein Element X ausgewählt.
- Die Elemente, die kleiner sind als das Element X, werden links von ihm angeordnet.
- Die anderen Elemente werden rechts von ihm angeordnet.
- Nun wird in gleicher Weise mit der linken und der rechten Teilfolge verfahren – aber nur dann, wenn die jeweilige Teilfolge aus mindestens zwei Elementen besteht.
- Zur linken Teilfolge gehören alle Elemente vom ersten Element bis zum Element X.
- Zur rechten Teilfolge gehören alle anderen Elemente.

Finden einer Hypothese

Beispiel: Quicksort – BESTER FALL

4 Zahlen

xxxx
xx
  xx

xxxx
xxxx

Finden einer Hypothese

Beispiel: Quicksort – BESTER FALL

8 Zahlen

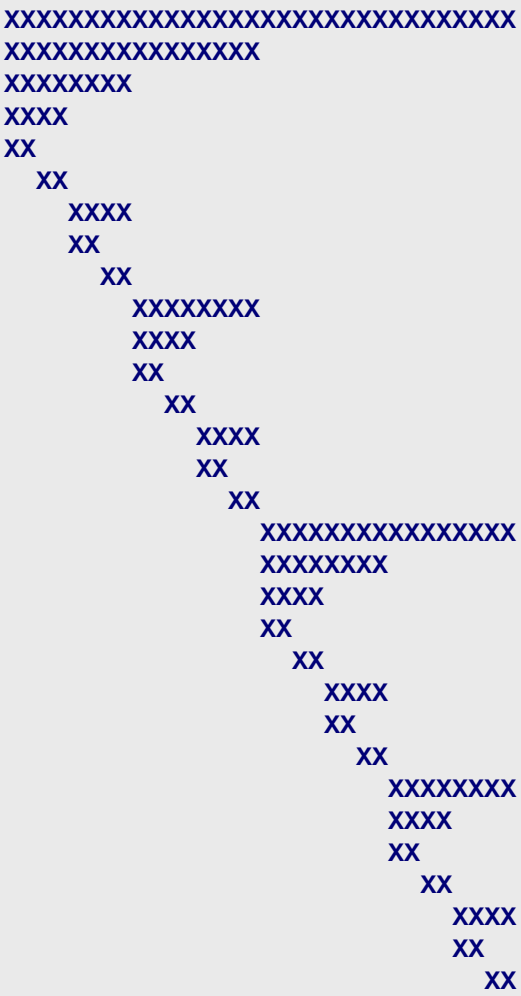
Finden einer Hypothese

Beispiel: Quicksort – BESTER FALL

16 Zahlen

Finden einer Hypothese

Beispiel: Quicksort – BESTER FALL



32 Zahlen

Finden einer Hypothese

$$t \sim n \cdot \log_2 n$$

Überprüfen der Hypothese

Beispiel: Quicksort – BESTER FALL

Die Funktion $f(x) = x \cdot \log_2 x$ ($x \in \mathbb{R}$, $x > 0$) ist fast linear.
Sie hat die Eigenschaft, dass sich beim Verdoppeln des Argumentes der Funktionswert auf das $(2 \cdot (1 + 1 / \log_2 x))$-fache vergrößert.
Der Quotient $f(2x) / f(x)$ konvergiert mit wachsendem x (langsam) gegen 2.

Überprüfen der Hypothese

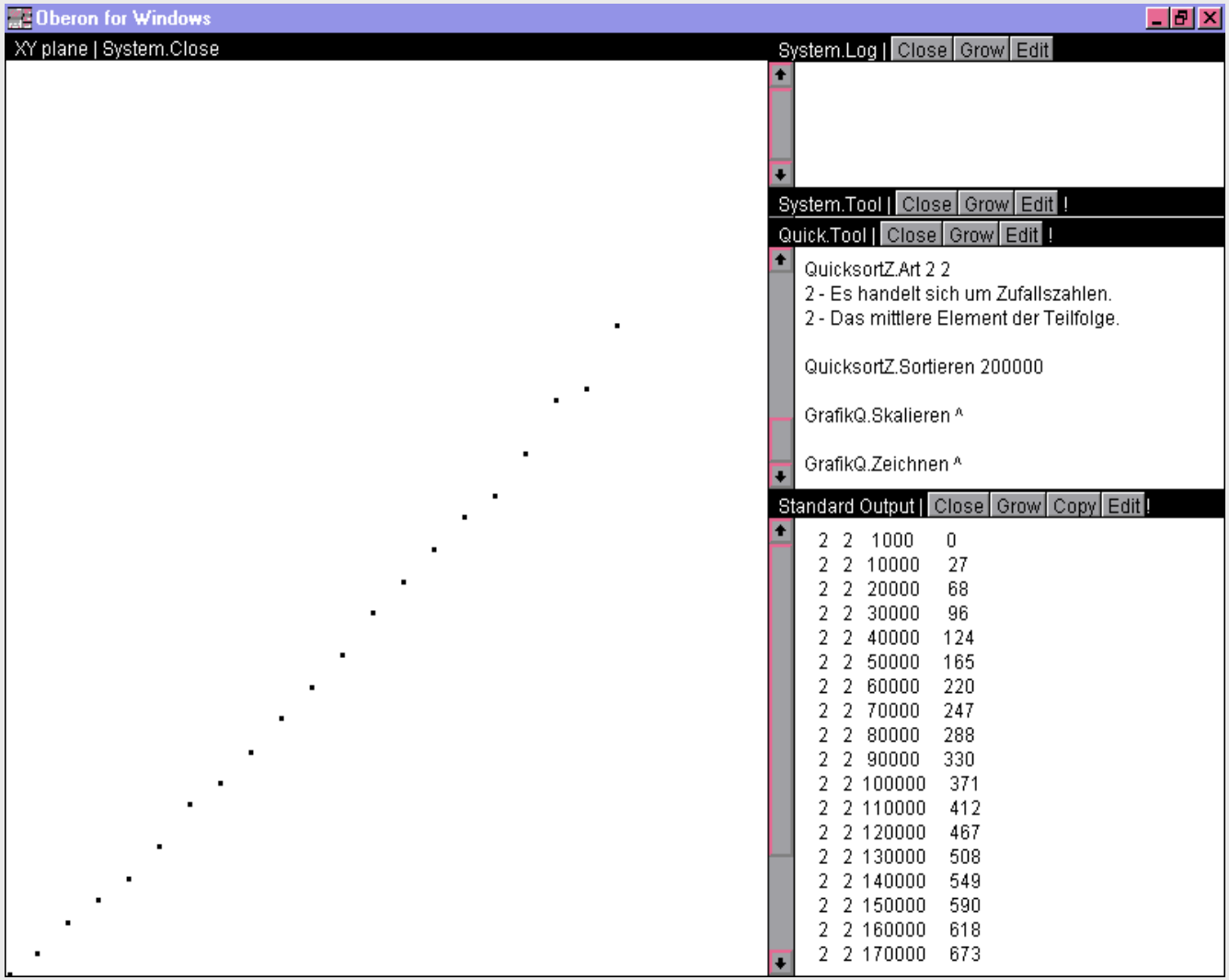
Beispiel: Quicksort – BESTER FALL

Die Funktion $f(x) = x \cdot \log_2 x$ ($x \in \mathbb{R}$, $x > 0$) ist fast linear.
Sie hat die Eigenschaft, dass sich beim Verdoppeln des Argumentes der Funktionswert auf das $(2 \cdot (1 + 1 / \log_2 x))$-fache vergrößert.
Der Quotient $f(2x) / f(x)$ konvergiert mit wachsendem x (langsam) gegen 2.

Die Funktion $f(x) = x \cdot \log_2 x$ ($x \in \mathbb{R}$, $x > 0$) ist jedoch nicht linear.
Die Schülerinnen und Schüler sollen wissen, dass lineares Zeitverhalten nicht erreichbar ist.

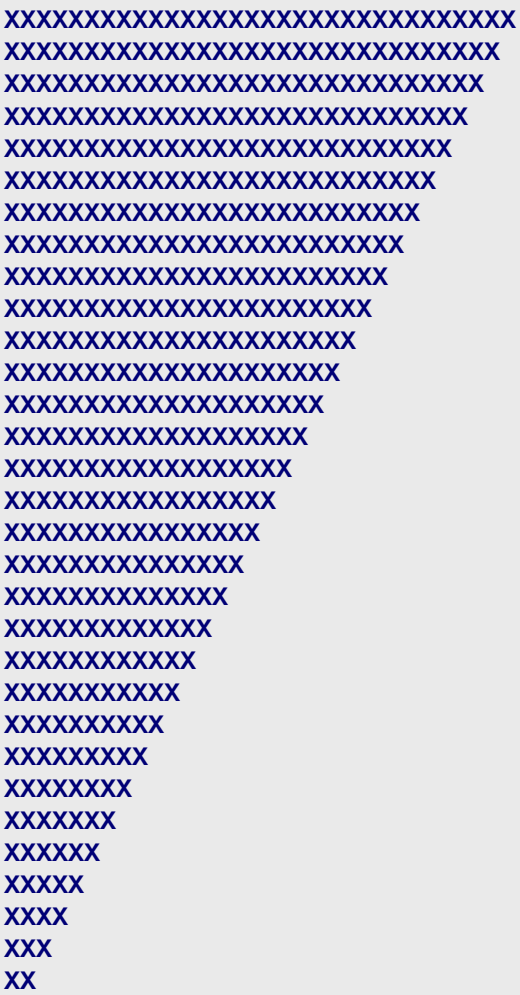
Überprüfen der Hypothese

Ergebnisse von Zeitmessungen bei Quicksort
Zufallszahlen – Trennelement: mittleres Element einer jeden Teilfolge



Finden einer Hypothese

Beispiel: Quicksort – SCHLECHTESTER FALL 32 Zahlen



Untersuchung von zwölf Fällen

10.000 Zahlen (Datentyp REAL) Notebook von 1997

Gegebene Zahlen	Trennelement	Zeit (ms)
aufsteigend sortiert	erstes	6834
	mittleres	18
	letztes	6927
	zufälliges	21
Zufallszahlen	erstes	41
	mittleres	39
	letztes	39
	zufälliges	27
absteigend sortiert	erstes	6767
	mittleres	9
	letztes	6372
	zufälliges	28

Speicheraufwand

Beobachtung:
Jede Folge von x repräsentiert einen rekursiven Unterprogramm-Aufruf.

Benötigt wird Speicherplatz für die zu sortierenden Elemente und zusätzlich:

im besten Fall:	$s \sim \log_2 n$
im schlechtesten Fall:	$s \sim n$

Spagetti-Computer

- Der Funktionalität eines Analogcomputers liegt eine physikalische Größe zu Grunde.
- Beim Spagetti-Computer ist dies die Länge.
- Für jede Zahl wird eine Spagetti-Stange entsprechender Länge hergestellt. Zum Beispiel ist für die Zahl 85 eine Spagetti-Stange der Länge 85 cm herzustellen.
- Zum Sortieren wird das Bündel an Spagetti-Stangen auf einer Ebene kräftig aufgestoßen und dann werden die Spagetti-Stangen der Länge nach von oben nach unten weggenommen und gemessen. Die Zahlen werden aufgeschrieben.
- Dieser Analogcomputer besitzt lineare Zeitkomplexität.
- Der interessante Ansatz scheitert leider bei einer größeren Anzahl von zu sortierenden Zahlen.

Aufgaben mit Praxisbezug nehmen

- Komplex 1: Tauschen zweier Werte, Addieren zweier Zeiten,

  Variablen, Ein- und Ausgabeanweisungen, ganze Zahlen, arithmetische Operationen, Wertzuweisung
- Komplex 2: Bruchrechnung mit dem PC, gregorianischer Kalender
  Wahrheitswerte, logische Operationen, bedingte Anweisung, bedingte Schleife, Laufanweisung, Funktionen, Algorithmus
- Komplex 3: Prüfbit, ISBN, Suchen in Texten, Cäsar-Code

  Zeichenketten, String-Operationen
- Komplex 4: schnelles Potenzieren, Zufall, Bisektion

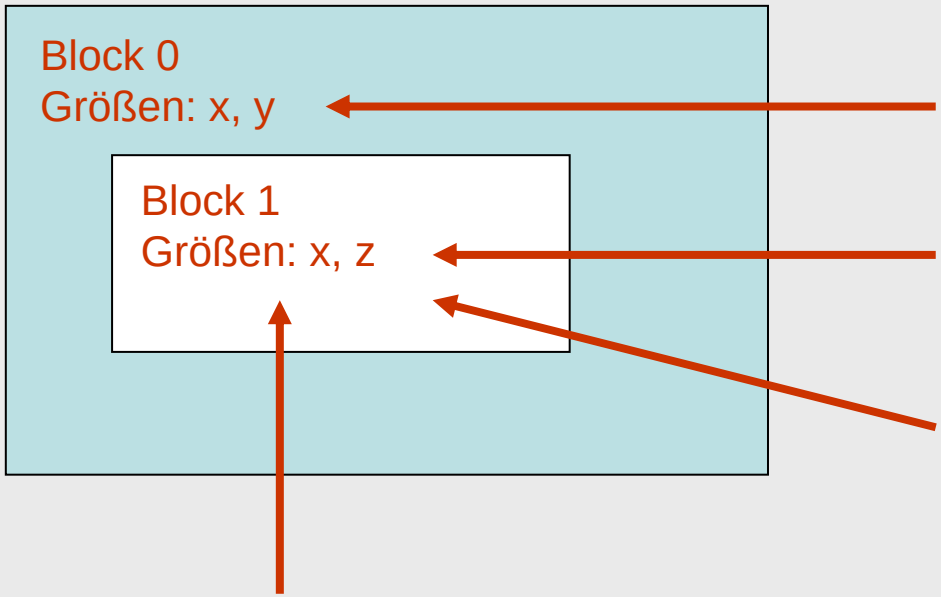
  Gleitpunktzahlen, arithmetische Operationen
- Komplex 5: Bubblesort, Sortieren von Datensätzen, Suchen von Datensätzen

  Listenverarbeitung
- Komplex 6: Überarbeiten von Programmen aus den Komplexen 1-5

  Module

Mit Visualisierungen arbeiten

Beispiel „Konzept der Lokalität“



Die Größen x und y sind hier lokal.

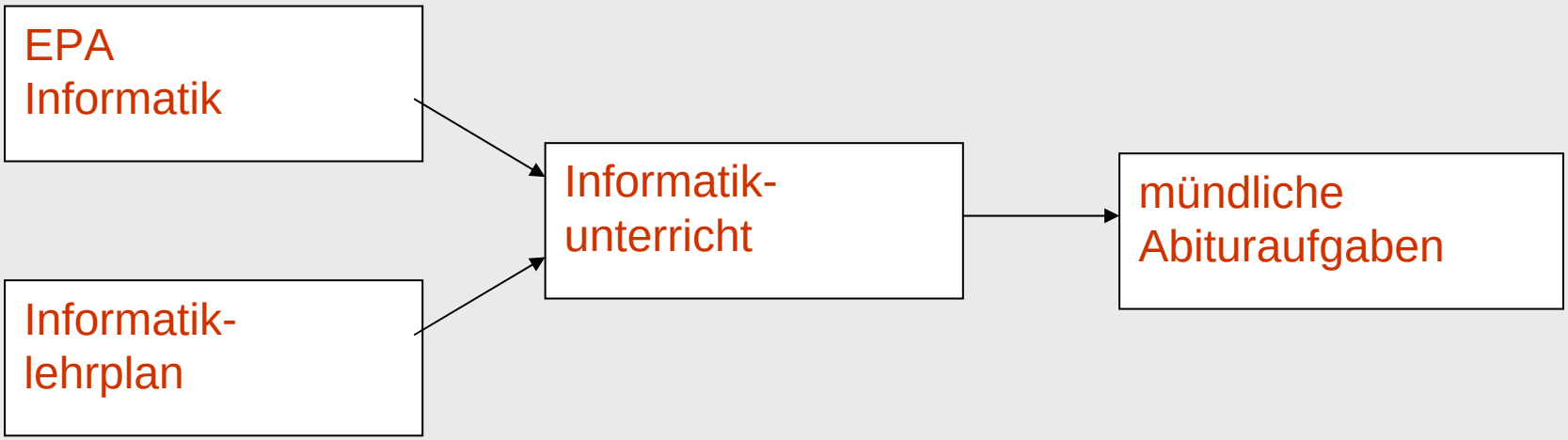
Die Größen x und z sind hier lokal.

Die Größen x und y aus Block 0 sind hier global.

x global, x lokal
Welche Größe x wird hier genommen?
Was passiert mit der anderen Größe x?

LOOK!
(H. Zemanek)

Standards – Unterricht – Prüfungen



Mündliche Abiturprüfung

Die Suchmaschine GUTSUCH verwendet beim Beantworten von Anfragen eine Datenbank. Die Datenbank verwaltet unter anderem Schlüsselwörter und Internet-Adressen.

a) Entwickeln Sie ein ER-Diagramm, das die Daten, die die Suchmaschine beim Beantworten von Anfragen benötigt, modelliert.

b) Eine Anfrage an die Suchmaschine GUTSUCH kann mehrere Schlüsselwörter enthalten. Erläutern Sie, wie solche Anfragen von der Suchmaschine bearbeitet werden können.

c) Erläutern Sie einen Algorithmus, der Schlüsselwörter aus einem Text herauslöst. Der Algorithmus soll solche Wörter wie „der“ und „eine“ als Schlüsselwörter ausschließen.

Mündliche Abiturprüfung

Definieren Sie den Begriff „Matroschka“ auf rekursive Art.

Eine Matroschka besteht aus einer Puppe, die eine Matroschka enthält, oder es ist die kleinste Matroschka.

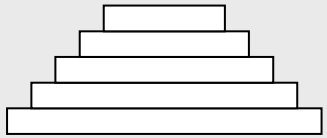
Mündliche Abiturprüfung

Erläutern Sie einen Algorithmus zur Lösung des Problems „Türme von Hanoi“.

Mündliche Abiturprüfung

Begriffliche Modellierung: rekursiver Begriff „Turm“

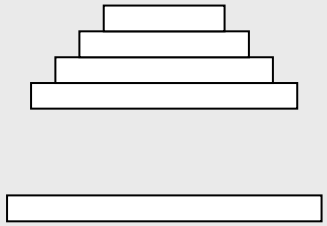
Ein Turm besteht aus der größten Scheibe und dem Rest-Turm,
oder es ist der leere Turm.



Mündliche Abiturprüfung

Begriffliche Modellierung: rekursiver Begriff „Turm“

Ein Turm besteht aus der größten Scheibe und dem Rest-Turm,
oder es ist der leere Turm.



Mündliche Abiturprüfung

Das Versetzen eines Turmes vom Startfeld zum Zielfeld erfolgt in drei Schritten:

1. Schritt:
Der Turm wird ohne seine größte Scheibe vom Startfeld zum Hilfsfeld versetzt.

2. Schritt:
Die größte Scheibe wird vom Startfeld zum Zielfeld versetzt.

3. Schritt:
Der Turm (siehe 1. Schritt) wird nun vom Hilfsfeld zum Zielfeld versetzt.

Die drei Schritte werden rekursiv auf jeden Turm, der zu versetzen ist, angewandt.

Der Abbruch (Basisfall) liegt vor, wenn der Turm leer ist.

Entwurf

Aufgabe:
Erarbeiten Sie ein Programm, das für ein beliebiges Jahr aus dem Zeitraum 1701-2100 einen Jahreskalender erstellt.

Beispiel: 2008

Monat: Januar

Mo	Di	Mi	Do	Fr	Sa	So
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Monat: Februar

Mo	Di	Mi	Do	Fr	Sa	So
				1	2	3
4	5	6	7	8	9	10

…

Entwurf

Recherche-Ergebnis zum gregorianischen Kalender:

- Im Jahr 1582 wurde von Papst Gregor XIII. eine Kalenderreform angeordnet.
- Nach den Festlegungen der Reform folgte auf den 4. Oktober 1582 der 15. Oktober. Es wurden also 10 Tage ausgelassen.
- Außerdem wurde eine neue Regelung für die Festlegung eines Schaltjahres eingeführt. Die Regelung ist auch heute noch gültig.
- Der gregorianische Kalender wurde in den meisten katholischen Ländern sofort eingeführt, in den protestantischen Ländern Deutschlands erst 1700.

Entwurf

Das Datum wird als Tupel (jahr, monat, tag) modelliert.

Zuerst sind die folgenden drei Funktionen zu realisieren:

- logische Funktion schaltjahr(jahr), die ermittelt, ob ein Jahr ein Schaltjahr ist
- Funktion anzahl(jahr, monat), die die Monatslänge ermittelt
- Funktion wochentag(jahr, monat, tag), die die Nummer des Wochentags für ein gegebenes Datum ermittelt (6 → Sonnabend)

Entwurf

Die drei Funktionen werden beim Erstellen des Jahreskalenders genutzt.

In einer while-Anweisung werden die Monate nacheinander bearbeitet.

Für jeden Monat wird die Nummer des Wochentags des Monatsersten ermittelt. Aus dieser Nummer ergibt sich, wie weit vor der Ausgabe der Zahl 1 einzurücken ist.

Dann werden nacheinander die Zahlen von 1 bis zum Monatsletzten ausgegeben.

Jedes Mal, wenn ein Sonntag ausgegeben wurde, wird eine neue Zeile begonnen.

Differenziertes Arbeiten: Das Programm ist ausbaufähig!
Motto: Wer Ostern hat, hat alles.

A prototype is a program
that nearly works.
(Koster)