

## Übersicht

### Ein Vorschlag zum Themenbereich Algorithmen

Beschreibung der allg. Zielrichtung und Informationen für Kollegen . . . . .	S.1
Vorwort . . . . .	S.3
Aufgabensammlung zum Thema Algorithmen im Anfangsunterricht . . . . .	S.4
Lösungshinweise . . . . .	S.9
1.Klausur (Schuljahr 2000/2001) . . . . .	S.20

#### **Ziel:**

Bedeutung des Algorithmus zeigen;  
Beschreibung von Algorithmen angehen;  
dazu nötige Sprachelemente aus der Alltagswelt herausarbeiten und weiter präzisieren;  
Grundelemente von Programmiersprachen verdeutlichen; (Beachtung des Zusammenhangs mit den Datentypen, auf denen operiert wird)  
Verschiedenartigkeit von Sprachen verdeutlichen

#### **Wann:**

Zu Beginn der Klasse 12; sofort nach dem ersten Kennenlernen, erstem Umgang mit der vorhandenen Software bzw. parallel dazu. (Wechsel: Theorie/Arbeit am Computer)

Wiederaufgreifen in 13:

Die herausgearbeiteten Sprachelemente finden sich als WHILE-Sprache auch in der theor. Informatik wieder und zeigen sich äquivalent zu anderen zentralen Berechenbarkeitsmodellen. Effizienz von Algorithmen (Suchen, Sortieren) und Grenzen für Algorithmen.

#### **Wie:**

Ausgehend von Aufgaben (vgl. Skript/Aufgabenblätter)  
Umsetzung in einer Programmierwelt (Hamster, Turtle, Kara)

#### **Warum?**

Pro:

Algorithmen stellen einen Ansatzpunkt aus der Erfahrungswelt der Schüler dar und gleichzeitig einen zentralen (zeitunabhängigen) Begriff der Informatik.

Algorithmen als Einstieg in Sprachen und ihre Grenzen.

"Programmieren als Allgemeinbildung", denn "zeitliche Abläufe streng zu spezifizieren, gehört im Zeitalter der IT zum allgemeinen gedanklichen Rüstzeug." (Prof. Nievergelt, ETH Zürich)

Kontra:

"Die Konstruktion von Software -im Großen wie im Kleinen- ist ingenieurmäßiges Tun und nicht allgemeinbildend. Programmieren in prozeduralen Sprachen wird immer unwichtiger." (z.B. Burkert, HIBS Wiesbaden)

**Literatur/Material:**

Eine Orientierung bieten z.B.

**Appelrath/Boles/Claus/Wegener** "Starthilfe Informatik" (1998) Stuttgart, Teubner

**Bauknecht/Zehnder** "Grundlagen für den Informatikeinsatz" (1996<sup>5</sup>) Stuttgart, Teubner

**Goldschlager/Lister** "Informatik, eine moderne Einführung" (1990<sup>3</sup>) München, Hanser

**Kursskizze:**

Man kann in Anlehnung an Goldschlager/Lister den Zugang zu Algorithmen und der präzisen Beschreibung zeitlicher Abläufe als Einstieg in den Kurs nutzen. Dies ermöglicht eine parallele Arbeit ohne Computer wie auch mit Computer. vgl. Skript.

Im rechnerunabhängigen Teil empfiehlt es sich den Begriff Sprache und nicht bereits Programmiersprache zu nutzen. Zu diesem frühen Zeitpunkt sollte man die Unterscheidung Syntax / Semantik noch nicht problematisieren. Das führt nur zu einer Überbewertung der Syntax-Regeln. Ziel sind aber Ausdrucksweisen, allgemein gültige Sprachelemente.

(vgl. Goldschlager/Lister:

S.11-12 oben (Computer, Algorithmus, Prozess, Prozessor)

S.16 §1.2 Prozessor, interpretieren

S.20-23 §1.4 Bedeutung von Algorithmen;

S.32-35 ein Alltagsalgorithmus und die schrittweise Verfeinerung

Kl.13:

S.94ff Aufwandsabschätzungen, Komplexität

S.79ff Grenzen der Berechenbarkeit: Halteproblem, PKP,...)

-----

Erfahrungen Anfangsunterricht:

(Alltagsalgorithmen und Programmierumgebung TURTLE)

Es werden sehr schnell die drei wesentlichen Sprachelemente (Sequenz, Alternative, Iteration) herausgearbeitet sowie die Modularisierung, die bereits bei den ersten Algorithmen genutzt wird. Es entsteht das Bedürfnis nach einer sinnvollen Notation. Schüler sind bereit die Normsprache z.B. aus G/L zu übernehmen. Schüler fragen sofort nach, bis zu welcher Präzision die Ausformulierung gehen muss.

Die Rekursion wurde erst zu Beginn der Klassenstufe 13 anhand grafischer Darstellungen (Koch-Kurve, DrachenKurve, Streckenzeichnen, ...) genutzt - wieder mittels der Turtle und anschließend auf nicht grafische Probleme (z.B. Türme von Hanoi, Quicksort, ...) übertragen. Die Fakultät sollte hier nicht das Einstiegsbeispiel darstellen.

## Algorithmen im Alltag und in der Informatik:

Im Gegensatz zur rasenden Geschwindigkeit, mit der sich Einsatzbereich, Einsatzart und das sichtbare Umfeld informatischer Geräte und Systeme verändern, mit der sich insbesondere Software-Updates gegenseitig (ver-)jagen, bleiben die Grundlagen der Informatik zeitlich fest. Einer dieser Grundpfeiler der Informatik ist der Algorithmusbegriff. Dieser bietet darüberhinaus durch Analogien im Alltag und der Erfahrungswelt der Schüler einen Anknüpfungspunkt für den Unterricht und er bietet einen sinnvollen Anknüpfungspunkt an überschaubaren Problemstellungen.

Dieser Weg soll hier beschritten werden.

Zum einen durch einen Vorschlag für ein Schülerskript, das ausgehend von Alltagsalgorithmen sich mit Grundlagen der Informatik befasst und einen intuitiven Begriff der Berechenbarkeit sehr früh mitaufnimmt, zum anderen durch eine Aufgabensammlung, die so angelegt ist, dass an Beispielen aus dem Alltag die wesentlichen Sprachelemente zur präzisen Lösung algorithmisierbarer Probleme herausgearbeitet werden - unabhängig von einer speziellen Implementationssprache. In diesen Algorithmen werden von Anfang an **handelnde** Objekte identifiziert und herausgestellt. Hier bieten Turtle, Hamster, Roboter, ... einen Übergang von realen Objekten zu denen der Informatik.

Dies unterscheidet sich von Wegen, mit statischen und sehr abstrakten Objekten beginnend (z.B. Textbereichen, Textdokumenten, ... ) dem Schüler einen Zugang zur (statischen) Modellierung zu verschaffen.

## Aufgabensammlung: Algorithmen im Alltag und in der Informatik:

### A1) Wörter suchen:

In solchen Rätseln sind in einem oft quadratischen Feld Worte versteckt, die horizontal, vertikal oder diagonal (auch rückwärts) geschrieben sind.

In einem kleinen Feld lassen sich gesuchte Worte noch schnell selbst finden. Schon bei 50x20 Zeichen ist es aber sehr ermüdend.

Erläutern Sie präzise, wie Sie einen Suchknecht instruieren, damit er alle versteckten Worte findet. Versteckt wurden fünf Worte:

Leim, Informatik, Abi, Kurs, Eimer

P.S. Sie sind auch in diesen 20x8 Zeichen versteckt.

Wie stellen sie sich den Suchknecht vor?

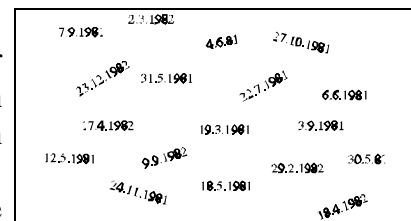
Könnten Sie die fünf Worte auch finden lassen, wenn nur bekannt ist, dass irgendwelche deutschen Worte versteckt sind?

```
ASDFERFGTHZJUKIJERTX
TIERNKMENOAERSOIMMEL
KJHZRUINNVHMGSEBOLE
OKDFJRUEHZCCRBOHDARI
IRGENSJISEREMIELIERN
LPOUJNASHZRTEUFIFNEF
EHSZEURKITAMROFNIAF
AGREHZTUECJNBHEEIOWT
```

### A2) Sprecher suchen:

Für eine große Gruppe von Leuten ist ein Sprecher zu benennen. Die Gruppe einigt sich darauf, den Ältesten mit dieser Aufgabe zu betrauen. Erläutern Sie schriftlich, wie Sie den ältesten ermitteln.

Lässt sich ihr Lösungsverfahren auch nutzen, um die größte von mehreren Zahlen zu bestimmen?



### A3) Die Wahl-Aufgabe:

Bei Wahlen müssen die Stimmzahlen anschließend in Sitze umgerechnet werden. Beschreiben Sie das oder ein Verfahren, nach dem diese Mandatsverteilung [in Baden-Württemberg] geschieht. Wo finden Sie geeignete Quellen?

Bei einer Gemeinderatswahl am 7.5.2000 wurden für die sechs angetretenen Parteien folgende Stimmzahlen abgegeben:

A	B	C	D	E	F
20651	21716	8414	5304	4351	47848

Welche Sitzverteilung erwarten Sie bei 22 zu vergebenden Sitzen?

### A4) Die Abi-Aufgabe:

Im Leitfaden für die gymnasiale Oberstufe steht im Abschnitt 5.3. über den Abiturprüfungsblock:

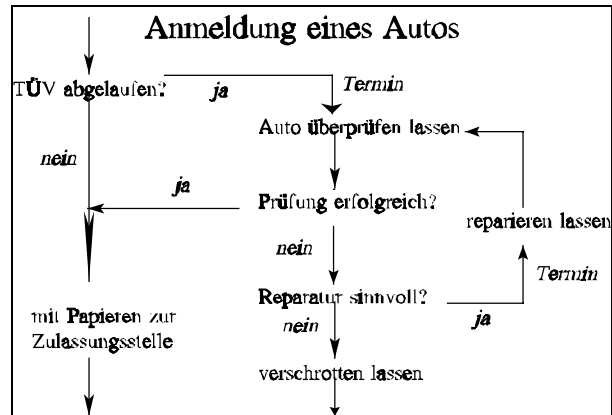
*"In den vier Prüfungsfächern müssen zusammen mindestens 100 Punkte erreicht werden. Dabei müssen in zwei Prüfungsfächern, darunter einem Leistungsfach, mindestens je 25 Punkte erreicht werden."*

Geben Sie Beispiele an, in denen die geforderte Qualifikation erreicht wird und solche, in denen sie nicht erreicht wird.

Geben Sie Beispiele an, in denen die geforderte Qualifikation nicht erreicht wird, der

Schüler aber mehr als 100 Punkte im Abiturprüfungsblock hat. Geben Sie ein extremes Beispiel an, bei dem die geforderten Werte gerade nicht erreicht werden. Schreiben Sie präzise auf, wie Sie überprüfen, ob ein Schüler diese Bedingungen des Abiturprüfungsblockes erfüllt.

- A5) Auto-Anmeldung:  
Verbalisieren Sie das grafisch dargestellte Verfahren zur Anmeldung eines PKW bei der Zulassungsstelle. Bietet die grafische Darstellung Vorteile?



- A6) Schemen und Regeln:  
Viele Vorgänge des täglichen Lebens sind schematisiert und folgen festen Regeln; aus Gewohnheit und Bequemlichkeit, wegen Gesetzen und Geboten oder aus innerer Notwendigkeit. Diese Vorgänge ähneln damit Algorithmen.

- a) Beschreiben Sie möglichst präzise die Alltagsalgorithmen:
- Herstellen einer Telefonverbindung
  - Schalten vom 1. in den 2. Gang
  - Binden beider Schuhe
  - Suchen eines Begriffs im Lexikon
  - Kochen zweier Frühstückseier
- b) Woran scheitert die Präzision z.B. im letzten Fall?

- A7) Verschlüsselung:  
Sie tauschen häufiger Mitteilungen aus z.B. über das Handy. Leider können diese kurzen Botschaften auch von anderen Familienmitgliedern oder Freunden gelesen werden, wenn sie gerade das Handy haben. Sie benutzen daher eine Geheimschrift. Eine einfache Verschlüsselung für Texte könnte so lauten:

Verschlüsseln Sie damit: 'Ein erster, kurzer Test.'

Welche Schwächen hat das Verfahren? Geben Sie ein anderes Verfahren an.

Welche Eigenschaften sollte Ihr Verfahren besitzen?

Unterscheide nicht zwischen Groß- und Kleinschreibung;  
Beginne mit dem ersten Zeichen des Textes;  
**solange** noch Zeichen da sind **wdh**:  
  **wenn** es ein Buchstabe außer 'z' ist,  
    **dann** ersetze ihn durch den alphabetisch nächsten,  
  **sonst wenn** es ein 'z' ist,  
    **dann** ersetze ihn durch 'a';  
**wenn** es ein Satzzeichen **oder** Leerzeichen ist,  
  **dann** übergehe es.  
**ende-solange**

- A8) Leerstellen:  
In einem Text sollen alle doppelten Leerstellen entfernt werden. Wie geht das? Was setzen Sie bei dem voraus, der Ihr Verfahren ausführt.

A9) Welche der folgenden Probleme lassen sich nach Ihrer Meinung nicht durch Algorithmen lösen? Warum glauben oder woher wissen Sie dies?

- Flicken eines Lochs im Fahrradschlauch
- Beheben einer Fahrradpanne
- Addieren zweier Brüche
- Lösen der Mathe-GK-Abiaufgabe A1 von 1997
- Lösen einer Mathe-GK-Abiaufgabe
- Das Übersetzen eines Geschäftsbriefes
- Das Finden eines Freundes / einer Freundin
- In der nächsten Deutsch-Klausur 14 Punkte schreiben
- Den Hunger auf der Welt beseitigen
- Echte von falschen Entschuldigungen unterscheiden
- Echte von falschen Hundertern unterscheiden
- Echte von falschen Freunden unterscheiden
- Echte von falschen Überweisungen unterscheiden
- Einen Sechser im Lotto erzielen
- Durch Lotto-Spielen reich werden
- Das Spielen einer Sonate
- Das Komponieren einer Sonate
- Einen Hit produzieren

Welche der Aufgabenstellungen sind nicht präzise? Welche können nicht präzise sein?

A10) Durchschnittsalter:

"Das Lehrerkollegium wird immer älter!", klagen viele Kollegen (und Schüler?). Wie würden Sie das Durchschnittsalter des Kollegiums feststellen? Welche Angaben müssen Ihnen zur Lösung der Aufgabe bekannt sein?

Schreiben Sie Ihr Vorgehen so auf, dass jedermann danach in der Lage wäre, das Durchschnittsalter des Kollegiums zu berechnen oder berechnen zu lassen.

A11) Der Sammel-Roboter

Die Buchstaben (vgl. A1) liegen innerhalb eines rechteckigen Feldes, das ein kleiner Roboter Platz für Platz ablaufen kann. Er kann aber immer nur das Feld *direkt vor ihm* betrachten und den dortigen Buchstaben aufnehmen, identifizieren und sich merken. Instruieren Sie ihn, die versteckten Worte zu finden.

Ein \* markiert eine unzugängliche Stelle.

Was an der Aufgabenstellung erscheint Ihnen unpräzise? Präzisieren Sie es ggfs.

Können Sie Ihr Lösungsverfahren aus A1) oder Teile davon wieder nutzen? Was ändert sich?

b) Erläutern Sie präzise, wie Sie den Roboter beauftragen, um alle 'E' in obigem Feld zu finden. Gibt es dazu verschieden gute Lösungen?

Neuere Roboter können nun sogar um sich schauen. Ändern sich Ihre Lösungsanweisungen dadurch?

ASDFERFGTHZJU*****
TIERNKMENOAERSO****
*JHZRUINNVHMGS****
***FJRUEHZCCRBOHD**
***ENSJISEREMIELIRE
**OUJNASHZRTEUFIFNE
*HSZEURKITAMROFNIA
AGREHZTUECJNBHEEIO

- A12) Geben Sie einen Lösungsweg an für das Sortieren der Namensliste des GK Informatik. Wie unterscheidet sich das vom Sortieren einer beliebigen Namensliste? Was sind hierbei Eingabe- und Ausgabedaten? Wer versteht Ihren Lösungsweg? Gibt es andere Lösungen?
- A13) **Prozess und Prozessor:**  
Was versteht man in der Informatik unter dem EVA-Prinzip?  
Welche Eingabedaten verarbeitet der Prozess "Frühstückseier kochen"? Welche Ausgabedaten liefert er?  
Welche Eingabedaten werden beim Berechnen des ggT benötigt? Welche Ausgabedaten liefert der Prozess? Wer kann das Verfahren durchführen?  
Wie ist es beim Prozess "Beethovens 5. spielen"? Sind hier die Eingabedaten präzise? Wer ist der Prozessor?  
Warum steht in diesen Fragen jeweils 'Prozess' und nicht 'Algorithmus'?  
Wie unterscheidet sich der formulierte Algorithmus, wenn Sie ihn für einen Koch bzw. für einen Strohvitwer aufschreiben, der bisher nie kochen musste?
- A14) **Der häufigste Buchstabe:**  
Es soll derjenige Buchstabe bestimmt werden, der in einem Textstück am häufigsten auftaucht. Überlegen Sie sich ein Verfahren zur Bestimmung des häufigsten Buchstabens.  
Was muss ein Computer können, um Ihren Algorithmus ausführen zu können?  
Kann man diesen Algorithmus auf die Suche nach der häufigsten Buchstabenkombination aus zwei (drei) Buchstaben erweitern?
- Der größte Freiraum:**  
Viele Buchstaben werden in dem betrachteten Textabschnitt dreifach oder noch öfter auftreten. Dabei werden die Vorkommen i.a. verschieden weit voneinander entfernt sein. Geben Sie ein geeignetes Maß für diese Entfernung an.  
Als 'Freiraum' eines Buchstabens wird nun die kleinste solche Entfernung festgelegt. Beschreiben Sie ein Verfahren, mit dem Sie den Buchstaben mit dem größten Freiraum ermitteln (lassen).  
Würden Sie den 'Freiraum' anders festlegen?

## A15) Ein Rätsel:

Wie gehen Sie als Gruppe daran, diesen Text zu entschlüsseln?

Geben Sie einen Weg an, auf dem Ihre Gruppe die Teil-Aufgabe aus dem 98er-Sommer-Rätsel der ZEIT löst. Unterscheidet sich der Weg von dem, den Sie alleine einschlägen?

RM WRVHVI SZFKGHGZWG VRMVH AF QVMVI AVRГ HVSI  
IVELOFGRLMHUIVFWRTVM OZMWVH DFIWV VRM RMHVOTVYLIVMVI  
UZHГ-ADVIT RN OVGAGVM NLMZG WVH ZXSGAVSMGVM  
QZSISFMWVIGH VIHГVI PLMHFO (33) FMW YVSVIIHXSGV HKZVГVI  
UZHГ TZMA VFILKZ. WVI SVIIHXSVI WVH TVHFXSGVM HGZZGVH RM  
VRMVI YVHGRNNGVM HGZWG RHG YRHXS LU (32) WRVHVI HGZWG  
FMW LYVISZFKG VRMVH DVOGIVRXSVH. ZM WRVHV N ELM VRMV N  
TVDRHHVM KSRORKK AFI SZFKHGZWG VIPOZVIGVM LIG  
IVHRWRVIGVM OZMTV AVRГ HGZGG KIRMA(37)VM MFI RMUZMGVM  
FMW YVMZSNVM HRXS LUG TVMFT DRV HLOXSV. ZM WRVHV N LIG  
EVIGIRVY HRXS HL NZMXSVI KZKHG (38) ELM POVNVMH YRH TIVTLI  
WRV VCROAVRG AFNVRHG ZFU IVXSG ZMTVMVSNV DVRHV. SRVI  
DFIWV WVI YVIFVSNGV HLSM VRMVH YVIFVSNGVM EZGVIIH  
TVYLIVM FMW HXSFU VRM NFHRPHGFVXP NRG WVN GRGVO  
HVRMVH PZRHVI(40)H. RM WRVHVI FMGVI WIVR EVIHXS RVWVMVM  
MZNVM YVPZMMGVM HGZWG VIOZT HL NZMXSVI KZHXSZ (39) DL SO  
AF LUG WVM EVIU FVSIFMTHPFVMHGVM HVRMVI SZIVNHWNVM,  
DZH NRG VRM TIFMW WZUFVI DZI, WZHH WZH LHNZMRHXS V IVRXS  
HKZVГVI AFN VIORVTVM PZN. MVSVM HRV WVM PFVIAVHGVM  
MZNVM I WRV RM HGVGGRM TVYLIVMV HLKSRV SVRIZGVGV ZM  
WRVHV N LIGV VRMVM TVDRHHVM KVGVI, DZH RSI HKZVГVI WRV  
NLVTORXSPVRG YLG, FMGVI ZMWVIVN ELIMZNV M ZOH AZIRM (34)  
YVIFVSNG AF DVIWVM. WRVHV ZOH EVGVIZMVMPLOLMRV FMGVI  
WVN MZNVM XZVHZIZFTFHGZ ELM VRMV N PZRHVI (40)  
TVTIFVMWVG V HGZWG YVUZMW HRXS OZMTV AVRГ FMGVI WVI  
SVIIHXSZUG ELM PZORUVM, WVIVM GRGVO WVN HFOGZM (36) RN  
LHGIVRXS VMGHKIZXS. VRM VCHXSZFHKRVOVI ZTRVIGV ZM Nб  
WRVHV N LIG OZVMTVIV AVRГ ZOH YRHXS LU (32), YVELI VI  
HKZVГVI DVTVM VRMVI YVHHVI WLGRVIGVM ILOOV MZXS HFVWVM  
TRMT. SRVI YIZXSGV VRM IVXSG ZFUNFVKURTVI QFMT-PZRHVI (40)  
HVRMVM PZMAOVI (31) UZHГ AFI EVIADVRUOFMT FMW OVGAGORXS  
AFN NVSI LWVI DVMRTVI VIADFMTVMVM IFVXPGIRGG, WVI ZOH  
"ZYTZMT WVH OLGHV M" DVOGDVRG YVPZMMG DFIWV. VRM ZFU  
VRMV N HXSOLHH YVR SZMMLEVI TVYLIVMVI UIVRSVII (35), WVI ZOH  
YVMRNNLMPVO TZMAVI TVMVIZGRLMVM YVPZMMG DFIWV, HGZIV  
RM WRVHVI SZUVMHGZWG.

Quelle: (ZEIT-Magazin 4.10.98)



## Lösungshinweise:

## Ziel der Aufgaben:

Häufig ist es sinnvoll, sich auf wenige Beispiele zu beschränken und diese in die Tiefe zu besprechen. Die Aufgabenbeispiele A1-A14 sind nicht dafür ausgelegt, dass die Schüler zu diesem frühen Kurszeitpunkt bereits zugehörige Lösungsalgorithmen auf dem Rechner realisieren. Die Aufgaben sollen dazu Anlass geben, Lösungsalgorithmen zu finden, zu beschreiben, zu vergleichen. Bei diesen Vergleichen sollen die Sprachelemente SEQUENZ, VERZWEIGUNG und WIEDERHOLUNG herausgearbeitet werden sowie die PROZEDUR als erste Möglichkeit einer Modularisierung von Problemlösungen.

Natürlich bieten sich zu den Aufgaben A1,A2,A4,A7,A10,A11,A12,A14 Lösungen auf dem Computer mittels einer Programmiersprache an.

Viele dieser Aufgaben können von den Schülern später in Programmiersprachen umgesetzt werden. Hier im frühen Stadium können fertige Programme zur Veranschaulichung genutzt werden. Der Schüler arbeitet als Programm-Benutzer und als Gestalter (umgangs-)sprachlicher Lösungen, nicht als Gestalter in einer Programmiersprache. Man kann hierbei auch schon Quelltexte betrachten. Dies bietet sich vor allem an, wenn sie die gleiche Idee nutzen, wie die sprachlichen Lösungsformulierungen der Schüler. Diese wird man dann im Programmtext zu identifizieren versuchen.

## Präzisionsniveau:

Generell werden die Antworten verschiedene Präzisierungsebenen enthalten, die im Gespräch verdeutlicht werden können. Dabei ist immer wieder zu klären: Welche Voraussetzungen muss der Prozessor mitbringen, also derjenige, der diese Anweisungen ausführen soll.

Da zu diesem Zeitpunkt keine Umsetzung in eine Programmiersprache geplant ist, wird das Präzisionsniveau nicht das einer Programmiersprache sein.

Alle vorgeschlagenen Lösungsverfahren werden zuerst darauf untersucht werden, ob sie korrekt sind, d.h. in allen (erlaubten) Fällen funktionieren. Dabei erscheinen sprachliche Formulierungen wie 'bei Fehleingaben wird nichts untersucht' oder 'dann wird abgebrochen' zulässig und sogar sinnvoll.

## Besonderheiten bei den Sprachelementen:

Es tauchen sehr rasch die drei konstituierenden Sprachelemente auf. Die Sprachelemente Verzweigung und Wiederholung werden i.a. verschiedenartig umschrieben. Häufig ist eine alltagssprachliche Umschreibung der Verzweigung zu finden, die sich nicht ganz mit der programmiersprachlichen deckt:

Prüfe, ob ... Wenn ja, dann ... Wenn nein, dann...	ist zu übertragen nach:  Man vereinbart, dass die Frage, die Prüfung mit <b>wenn</b> eingeleitet wird.	<table border="1"> <tr> <td> <b>wenn ...</b>  <b>dann ...</b>  <b>sonst ...</b> </td> </tr> </table>	<b>wenn ...</b> <b>dann ...</b> <b>sonst ...</b>
<b>wenn ...</b> <b>dann ...</b> <b>sonst ...</b>			

Damit können "Wenn ja..." sowie "Wenn nein..." entfallen.

Zu den einzelnen Aufgaben:

- A1) Es gibt Lösungen, die ein Wort nach dem anderen suchen und solche, die alle fünf Worte 'auf einmal' suchen. Insbesondere bei den 'Lösungen durch Draufschaun' sollte man versuchen herauszuarbeiten, was beim Draufschaun alles passiert. Man nutzt bereits ein Modularisierungskonzept, indem die Details unter der Hülle des Draufschauns subsummiert werden.

Suche LEIM:

Suche nach einem L.

Wenn<sup>(1)</sup> du ein L gefunden hast, dann betrachte die vier Nachbarfelder.

Wenn du ein E<sup>(2)</sup> gefunden hast, dann gehe in diese Richtung<sup>(3)</sup> noch zwei Schritte weiter. Wenn dabei I und M auftauchen, dann ist das Wort LEIM gefunden, sonst nicht.

Wenn du kein L gefunden hast, dann gibt es das Wort LEIM nicht.

Suche ein *Wort*:

Durchsuche der Reihe nach alle Zeilen von der ersten bis zur letzten;  
innerhalb einer Zeile jeweils von links nach rechts Spalte für Spalte.

Prüfe dabei jeweils<sup>(4)</sup>, ob der Anfangsbuchstabe (*des Wortes*) vorliegt.

JA: prüfe, ob in der Umgebung<sup>(5)</sup> der 2. Buchstabe vorliegt

JA: prüfe, ob in gleicher Richtung<sup>(6)</sup> der 3. Buchstabe vorliegt

JA: ...<sup>(7)</sup>

BIS alle Buchstaben in einer Richtung gefunden

NEIN: nächste Spalte auf Anfangsbuchstaben prüfen.

Bemerkungen:

<sup>(1)</sup>: Hier kann die grafische Darstellung verdeutlichen, dass die letzte Zeile 'Wenn du kein L...' den Alternativzweig sonst darstellt.

<sup>(2)</sup>: Ggfs nachfragen: Was ist, wenn mehrere E auf den Nachbarfeldern vorkommen?

<sup>(3)</sup>: Eine Beschreibung der Richtung auf Gitterebene ist nicht notwendig, wenn ein Mensch als Prozessor fungiert.

<sup>(4)</sup>: Damit wird die Wiederholung verdeutlicht.

<sup>(5)</sup>: Hierbei ist der Begriff Umgebung weiter zu präzisieren.

<sup>(6)</sup>: Gemeint ist dabei 'direkt anschließend'. Es wird aber nicht gesagt. So etwas passiert häufig in Arbeitsaufträgen. Es werden unausgesprochene (für den Auftraggeber selbstverständliche) Annahmen verwendet. Dies kann zu Missverständnissen führen.

<sup>(7)</sup>: Die Pünktchen bilden zusammen mit dem Schlüsselwort **bis** die noch sehr unscharfe Wiederholungsformulierung.

Weiter präzierte Lösung:

**für** alle Zeilen von der ersten bis zur letzten **wdh**:

**für** alle Spalten der aktuellen Zeile von der ersten bis zur letzten **wdh**:

**wenn** Anfangsbuchstabe (*des Wortes*) vorliegt,

**dann** suche die vier Nachbarfelder ab<sup>(2)</sup>

**wenn** hier der 2. Buchstabe vorliegt,

**dann** gehe in gleiche Richtung weiter;<sup>(1)</sup>

**wenn** der 3. Buchstabe vorliegt,

**dann** gehe in gleiche Richtung weiter; ...

das ist zu wiederholen **bis** das Wortende erreicht ist  
**oder** das Buchstabenfeld zu Ende ist<sup>(3)</sup>  
**sonst** ist nichts zu tun; d.h. zur nächsten Spalte gehen.

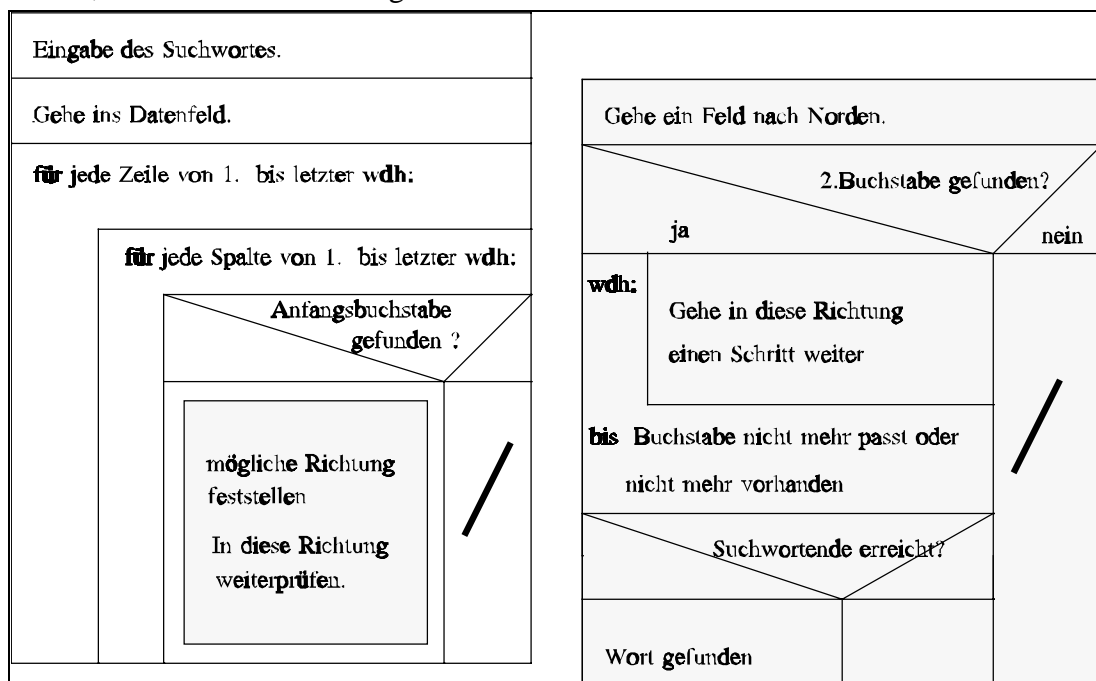
<sup>(1)</sup>: Sobald der zweite Buchstabe gefunden ist, wiederholen sich die immer gleichartigen Prüfungen auf den nächsten Buchstaben. Die Umwandlung dieser wiederkehrenden Prüfungen (=Verzweigungen) in die Schleifenbedingung fällt häufig nicht leicht. Bereits an dieser Stelle lässt sich darauf eingehen. Das 'wenn' innerhalb des Schleifenkörpers ist überflüssig.

<sup>(2)</sup>: Hierbei ist 'Suche die vier Nachbarfelder ab' wieder eine verborgene Wiederholungsanweisung.

<sup>(3)</sup>: Nach dem Ende dieser Schleife ist zu prüfen, wieso sie beendet wurde, da sie durch eine zusammengesetzte Bedingung überwacht wurde.

Auf diesem Präzisionsniveau wird nicht erläutert, wie erreicht wird, dass der jeweils passende Buchstabe des Suchwortes im Vergleich verwendet wird.

Man kann die Lösungen auch grafisch darstellen. Teillösungen können z.B. so aussehen und dabei auch verdeutlichen, dass sie bei vielen Details vor allem dazu dienen können, hierarchische Ordnungen sichtbar zu machen.



Die rechts skizzierte Verfeinerung enthält nur den ersten der vier nötigen Teile. Anschließend sind genauso die andern drei Richtungen zu untersuchen. In den Struktogrammen werden die durchzuführenden Prüfungen, die Bedingungen nicht so deutlich, wie sie zur Umsetzung in eine Programmiersprache werden müssen. Das kann an dieser Stelle herausgearbeitet werden.

Ebenso ist zu klären, wo (im Buchstabenfeld) das Verfahren weitergeht, ob nach dem Finden eines Wortes aufgehört wird (ist es zweimal vorhanden?).

## A2) Sprecher suchen

Das Verfahren dazu kann genauso auch für das Finden der größten bzw. kleinsten Zahl angewandt werden.

Idee1: Turniermethode

Sie funktioniert wie das k-o-System z.B. bei Tennisturnieren.

Immer zwei Daten hernehmen; das ältere Datum merken, das jüngere kann weg.

Ein Datum ohne Partner kommt zur Gruppe der Älteren (=Sieger)

Nun mit allen Älteren die nächste Runde des gleichen Spiels wieder durchführen.

Dieses k-o-System wird solange durchgehalten, bis nur noch ein Pärchen übrig ist.

Der Sieger hieraus ist der Älteste.

präziser:

- 1) Schreibe alle Daten der Reihe nach hin.
- 2) **solange** mehr als ein Datum übrig ist **führe durch**:
  - 2.1) eine Spielrunde
  - 2.1.1) eine Spielrunde
  - 2.1.2) Vergleiche beide. Stelle das ältere Datum fest. Merke es.<sup>(1)</sup>
  - 2.1.3) Wenn ein Datum ohne Partner ist, merke es auch.
- 3) Gib den Ältesten aus.

Verfeinerung der Spielrunde:

- 2.1.1) Nimm ein Datum und das nächste der Liste.
- 2.1.2) Vergleiche beide. Stelle das ältere Datum fest. Merke es.<sup>(1)</sup>
- 2.1.3) Wenn ein Datum ohne Partner ist, merke es auch.

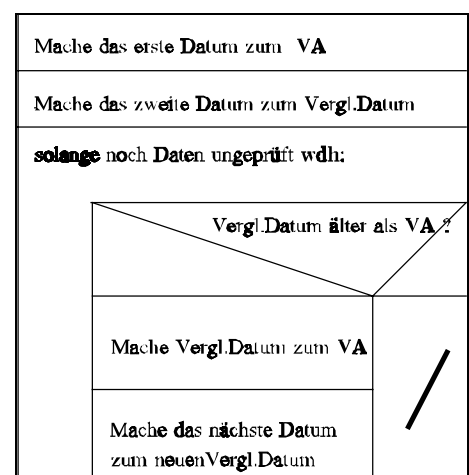
{<sup>(1)</sup>: 'Merke es' ist der Aufbau einer neuen Menge/Liste von Daten. Hier wäre bei weiterer Präzisierung auf Ausgangswerte zu achten. Damit liegen neue Daten für die nächste Spielrunde vor.}

Idee2: Lineare Auswahl

- 1) Nimm das erste Datum. Betrachte es als das vorläufig älteste (**VA**).
- 2) Beginne beim zweiten Datum:
- 3) **wenn** das Datum älter ist als das **VA**,  
**dann** haben wir ein neues **VA** gefunden.  
**sonst** ist nichts zu tun.
- 4) Betrachte das nächste Datum;  
**wenn** es vorhanden ist **dann** fahre bei 3) fort.

Präzisierung (verbal und grafisch):

- 1) Nimm das erste Datum. Betrachte es als das vorläufig älteste (**VA**).
- 2) Beginne nun mit dem zweiten Datum
- 3) **solange** noch Daten nicht geprüft sind **wdh**
  - 3.1) **wenn** das Datum älter ist als das **VA**.  
**dann** haben wir ein neues **VA** gefunden.
  - 3.2) Rücke ein Datum weiter



- A3) Die Schüler sollen sich z.B. mittels GK-Bücher, Politik-Lexika, Internet-Recherche über die Verfahren informieren und sie dann beschreiben. Man kann z.B. aus dem GK-Buch die entsprechenden Seiten kopieren oder über einen Scanner diese Seiten einlesen, so dass das dort beschriebene Verfahren besprochen werden kann. Ggfs. lässt sich hier die Unterscheidung zwischen Texterkennung und Scannen von Bildern erläutern (Referat).

**Das d'Hondtsche Höchstzahlverfahren.**

Die Aufgabe ist, das Verfahren zu beschreiben, das die Sitzverteilung berechnet. Dazu müssen die (gültigen!) Stimmenzahlen sowie die Zahl der zu vergebenden Mandate bekannt sein werden.

Andere Aufgaben wären:

- prüfen, ob das Verfahren kleine Parteien benachteiligt.
- zeigen, warum die Mandatsverteilung überhaupt zu Ungerechtigkeiten führen kann.

Auch wenn man eine erste Beschreibung des Verfahrens Beispiel entwickelt, so wird man verdeutlichen, dass ein Algorithmus allgemein für alle möglichen Beispiele zu formulieren ist.

Zur Erinnerung ein überschaubares Zahlenbeispiel (vgl. GK-Buch!):

Partei	A	B	C	D	Reihenfolge der Mandate
Stimmenzahl	28617	16244	3118	19828	
Teiler	-----				
1	28617	16244	3118	19828	01 03 00 02
2	14308	8122	1559	9914	04 07 05
3	9539	5415	1039	6609	06 11 09
4	7154	4061	779	4957	08 12
5	5723	3249	623	3966	10
6	4769	2707	520	3305	13
7	4088	2320		2832	
8	3577	2030		2479	
9	3180			2203	

Es werden nacheinander die (noch verbliebenen) Höchstzahlen ausgewählt, mit einem Mandat bewertet und gestrichen. So erhält die Partei A das 1., 4., 6., 8., 10. und 13. Mandat, wenn 13 Mandate zu vergeben sind; also insgesamt 6 von 13 Mandaten! Die Partei B erhielt das 3., 7. und 11. Mandat; also 3 von den 13 Mandaten. Partei C erhält kein Mandat. Partei D erhält 4 von 13 Mandaten, nämlich, das 2., 5., 9. und 12. Mandat.

Mögliche Schülerbeschreibungen:

Für jede Partei teilt man ihre Stimmenzahl nacheinander durch 1,2,3, usw..

Aus den so entstandenen Zahlen wählt man die größten aus; und zwar so viele, wie es Mandate gibt. Für jede dieser Zahlen erhält die zugehörige Partei ein Mandat.

Bemerkung:

Das Beschreibungsniveau geht von einem Betrachter aus, der die Zahlen überblickend (also vergleichend) die größten auswählen kann.

Man könnte weiter präzisieren lassen: Wie geschieht das Auswählen der größten?

Es treten deutlich die Wiederholungen "für jede Partei.." auf sowie die Ungenauigkeit 'teilen durch 1,2,3, usw.'. Es muss klargemacht werden wie weit dieses usw. geht; und zwar im allgemeinen Fall.

Andere Lösung:

Groblösung:

- 1) Zahl und Namen der Parteien eingeben.
- 2) Zahl der Mandate eingeben.
- 3) Der Reihe nach die gültige Stimmenzahl für jede Partei eingeben.
- 4) Der Reihe nach alle Mandate zuteilen.
- 5) Mandatsverteilung ausgeben.

Weiter präzisieren wird man nur Schritt 4): Zuteilen der Mandate.

Zuteilen des n.Mandates bei 4 Parteien:

(Voraussetzung: Vergleichszahlen sind ermittelt) <sup>(1)</sup>

- 4.1) Ermitteln zu welcher Partei die größte der 4 Vergleichszahlen gehört.
- 4.2) Anzahl der Mandate der dieser Partei um eins erhöhen.
- 4.3) Nächste Vergleichszahl (mit erhöhtem Teiler) für diese Partei berechnen.

Weitere Präzisierung von Schritt 4.1)

- 4.1) Gehe von der ersten bis zur letzten Partei der Reihe nach die Vergleichszahlen der Parteien durch. Suche die größte und merke, bei welcher Partei sie vorkommt.

... und weiter ...

- 4.1.1) Nimm die erste der Vergleichszahlen als vorläufig größte.

- 4.1.2) **wdh** für jede Partei von der 2. bis zur letzten der Reihe nach:  
**wenn** die jetzige Vergleichszahl größer als die vorläufig größte ist,  
**dann** merke dir die neue Vergleichszahl als vorläufig größte,  
ebenso die Stelle an der sie stand. (=zugehörige Partei)  
**sonst** tue nichts

{Am Ende der Wiederholung ist die größte Vergleichszahl bekannt, ebenso die Stelle an der sie stand.}

<sup>(1)</sup>: Dies wird später als Vorbedingung für Prozeduren auftauchen.

<sup>(2)</sup>: Ein Algorithmus zum Finden der größten Zahl lässt sich als eigenständige Aufgabe stellen.

Lösung:

Anzahl der Parteien N eingeben;

Anzahl der Mandate M eingeben;

Für jede Partei von der 1. bis zur N-ten: jeweilige Stimmenzahl eingeben;

Anfangswerte setzen:

d.h. Für jede Partei von der 1. bis zur N-ten: jeweilige Vergleichszahl festsetzen auf die Stimmenzahl dieser Partei;

Für jedes Mandat vom 1. bis zum letzten: akuelles Mandat zuteilen;

d.h.: Position der Höchstzahl im jeweiligen Vergleichsfeld feststellen;

Mandatszahl der entsprechenden Partei erhöhen;

Vergleichszahl dieser Partei neu berechnen;

Mandatsverteilung ausgeben.

Betrachten wir das Zuteilen des 5.(nicht des 1.) Mandates:

Dafür sind im Beispiel zu vergleichen:

Partei A	Partei B	Partei C	Partei D
9539	8122	3118	9914

Hiervon ist 9914 die größte Zahl, also erhält D das 5.Mandat. Partei D hat nun 2 Mandate. Aus der Tabelle oben sind die ersten beiden Zeilen der Partei D somit abgearbeitet. Für die folgenden Vergleiche ist bei Partei D die nächstfolgende Stimmen-teilerzahl, die 6609 zu benutzen. Die vorherigen sind bereits berücksichtigt.

Es sind bei den vier Parteien immer vier Zahlen zu vergleichen. Diese sind für jede Partei die abgegebene Stimmenzahl dividiert durch einen Teiler. Dieser Teiler ist i.a. für jede Partei ein anderer. Aber welcher? Was hat er mit den bisher erhaltenen Stimmen zu tun?

Wenn eine Partei bisher 2 Mandate hat, so ist die Zahl in der 3.Reihe für die nächsten Vergleiche zu benutzen. Bei n bisher erhaltenen Mandaten also Stimmenzahl dividiert durch (n+1). Für die Zuteilung des ersten Mandates kann man somit genau die vier abgegebenen Stimmzahlen nutzen.

Eine weitere Präzisierung ist zu diesem Zeitpunkt nicht nötig, da es noch nicht um die Umsetzung in die Programmiersprache geht. Bei der Behandlung von Reihungen (eindimensionalen Feldvariablen) kann man auf diese Lösung zurückkommen und die weitere Präzisierung durch Indizes darstellen.

Man wird herausheben, dass für jede Partei ähnliches durchzuführen ist. Dazu müssen die Daten für eine Partei in geeigneter Weise beschrieben werden (Reihung bzw. Feldvariable), damit man dies auch in einer Wiederholungsanweisung abarbeiten lassen kann. Damit lässt sich z.B. die Schreibweise  $S[i]$  für die Stimmzahlen der Partei näherbringen. Sie sollte zu diesem Zeitpunkt nicht weiter ausgebaut werden. Spätestens hier dringt die Programmiersprache ein. Denn solche Formulierungen wurden durch sie beeinflusst. Die später genutzte Programmiersprache muss uns hier soviel Freiheit wie möglich lassen, eigene Ideen zu formulieren.

-----

Nach dem angewandten Zuteilungsverfahren (d'Hondt) wurden im genannten Zahlenbeispiel vergeben (Gemeinderatswahl Hemsbach):

A	B	C	D	E	F
4	4	1	1	1	11

Nach Hare-Niemeyer hätte sich eine andere Mandatsverteilung ergeben.

Man kann ein fertiges Programm zur Ermittlung der Sitzverteilung nach d'Hondt zur Verfügung stellen (Java/Delphi/Tabellenkalkulation) und hierbei mit verschiedenen Mandatszahlen experimentieren (Zusammenarbeit mit Gemeinschaftskunde/Politik).

A4) Die Abi-Aufgabe: Beispiele:

Name:	LK1:	LK2:	GK_sch:	GK_mdl:	Summe:	best.
Anton	24P	44P	<b>32P</b>	<b>30P</b>	<b>130P</b>	<b>ja</b>
Bert	24P	22P	<b>50P</b>	38P	<b>134P</b>	nein
Claus	<b>62P</b>	18P	6P	13P	99P	nein
Diana	<b>30P</b>	24P	<b>26P</b>	19P	99P	nein

Die 24P im LK1 können dabei z.B. so zustandekommen: 4P in der schriftlichen Prüfung; keine mdl.Prüfung (ergibt  $4 \cdot 4P = 16P$ ) und 8P im Halbjahr 13.2.

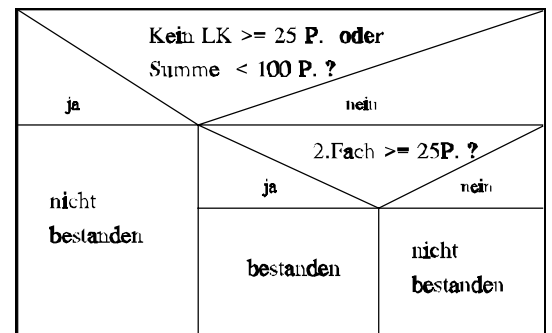
- Erster Formulierungsversuch:
- (1) **wenn** (kein LK über 25 P),  
    **dann** nicht bestanden.
  - (2) **wenn** (Summe unter 100 P),  
    **dann** nicht bestanden.
  - (3) **wenn** zwei LK  $\geq 25$  P,  
    **dann wenn** Summe  $\geq 100P$ ,  
        **dann** bestanden
  - (4) **wenn** (ein LK  $\geq 25$  P) **und** (ein weiteres Fach  $\geq 25P$ ),  
    **dann wenn** Summe  $\geq 100P$ ,  
        **dann** bestanden.

Werden durch diese 4 Abfragen alle Fälle erfasst? Wie baut man Abfragen systematisch auf, um dies zu erreichen? Muss man in (3) nochmals prüfen, ob die Summe  $\geq 100P$ . ist? Was geschieht sonst?

Diese Formulierung lässt sich vereinfachen. (3) und (4) bedeuten, dass ein LK und ein weiteres Fach mind. 25P. benötigen. ('Ein weiteres Prüfungsfach' in (4) schließt aber den zweiten LK mit ein, daher ist (3) überflüssig.)

(1) und (2) kann man zusammenziehen. Wenn die eine Bedingung ODER die andere Bedingung nicht erfüllt ist, dann ist das Abitur nicht bestanden.

Überprüfung:  
**wenn** (kein LK über 25P) **oder** (Summe unter 100P),  
    **dann** nicht bestanden;  
**sonst wenn** zweites Fach über 25P,  
    **dann** bestanden,  
    **sonst** nicht bestanden.



Diese Lösung benutzt wie der erste Lösungsversuch in (4) eine zusammengesetzte Bedingung. Durch **und** sowie durch **oder** lassen sich zwei Bedingungen zu einer zusammenfügen.



Die Negation der zusammengesetzten Bedingung der 1. Zeile sollte man mit Schülern verdeutlichen. Warum ist bei Erreichen des 1. **sonst**-Teiles sichergestellt, dass die 100-Punkte-Grenze erreicht wurde?

Hier sehen Sie eine weitere Möglichkeiten, die ohne zusammengesetzte Abfragen auskommt.

Womit wird dies erkauft? Wie lautet die verbale Darstellung dieser Lösungsidee?

Ein LK $\geq$ 25 P. ?			
ja			nein
2. Fach $\geq$ 25 P. ?			
ja		nein	
Summe $>$ 100 P. ?			
ja			nein
<b>bestanden</b>	nicht bestanden	nicht bestanden	nicht bestanden

- A7) Die Anweisung "übergehe es" für Satzzeichen bzw. Leerstellen ist nicht eindeutig. Sie kann so verstanden werden, dass diese im verschlüsselten Text nicht auftauchen, aber auch so, dass sie nicht verändert werden. Verschlüsseln von 'Ein erster, kurzer Test.' ergibt mit der ersten Interpretation: fjojfstufslvsafsuftu. Mit der zweiten Interpretation: fjo fstufs, lvsafs uftu. wird die Entschlüsselung einfacher.

Schwachstellen sind z.B.:

- das direkte Ersetzen eines Zeichens durch immer genau das gleiche Zeichen, so dass z.B. vor allem bei längeren Texten über die rel. Häufigkeiten auf das zugehörige Klartextzeichen zurückgeschlossen werden kann;
- das Beibehalten der Wortlücken und Satzzeichen, wenn man "übergehen" im zweiten o.g. Sinne interpretiert. Dann können über häufige Buchstabenkombinationen z.B. in Drei-Buchstaben-Wörtern wie bestimmten Artikeln Ansatzpunkte zur Entschlüsselung gefunden werden.
- sind einige Beziehungen Klartextzeichen - verschlüsseltes Zeichen gefunden, können die anderen wegen der Regelmäßigkeit der Ersetzung sofort erschlossen werden.

- A10) Man kann z.B. den Kurs seinen Durchschnittsgeburtstag erraten lassen. Damit kann man dann das Durchschnittsalter bestimmen. Es bietet sich an, hierfür ein Programm zur Verfügung zu stellen oder durch ein/zwei Schüler (z.B. mit einer TK) vorführen zu lassen. (Das bietet sich z.B. als erstes Referat an. Die Schüler erläutern ihre Idee und die Umsetzung mittels einer TK).

Lösungsidee:

Alle Geburtsdaten eingeben. Als Zellenformat dabei 'Datum' benutzen.

Anschließend den Mittelwert aller eingegebenen Daten berechnen lassen.

Das ist das Durchschnittsgeburtsdatum.

Vielleicht können die Schüler darauf eingehen, wie der Computer(?) mit den Datumsangaben rechnet.

Eine Beispieltabelle (WinWorks3 bzw. EXCEL) liegt bei.

- A12) Die Aufgabenstellung sagt nicht, wonach sortiert werden soll. Die Lösungshinweise hier sollen eine Sortierung nach den Nachnamen (als 1.Kriterium) erzeugen.

Idee: Nach und nach sortierte Liste aufbauen.

Präzisierung:

- 1) Nimm den ersten Eintrag der Liste<sup>(1)</sup>.
- 2) Setze ihn an die erste Stelle der neuen sortierten Liste.
- 3) Streiche ihn in der alten Liste.
- 4) Wiederhole solange noch Einträge zu verarbeiten sind:
  - 4.1) Nimm den nächsten Eintrag.
  - 4.2) Setze ihn an passende Stelle der neuen sortierten Liste.
  - 4.3) Streiche ihn in der alten Liste.
- 5) Gib die neue sortierte Liste aus.

Nimm <b>den ersten</b> Eintrag der Liste	
Setze ihn an <b>die erste</b> Stelle einer neuen Liste	
Streiche ihn in der alten Liste	
<b>solange</b> noch Einträge in der alten Liste sind:	
<b>wdh:</b>	Nimm <b>den nächsten</b> Eintrag der alten Liste
	Setze ihn an <b>die passende</b> Stelle der neuen Liste Die ist damit immer sortiert.
	Streiche ihn in der alten Liste
Gib die neue, sortierte Liste aus.	

Idee: Liste umsortieren.

Präzisierung:

- 1) Suche den alphabetisch ersten Nachnamen.
- 2) Tausche ihn nach vorn. Tauschpartner vom ersten Platz kommt an seine Stelle.
- 3) Bearbeite weiter nur noch die Restliste (=ohne erste Stelle).
- 4) Wiederhole solange noch nicht am Ende der Restliste:
  - 4.1) Suche in ihr den alphabetsich kleinsten Nachnamen.
  - 4.2) Tausche ihn in der Restliste nach vorn. Tauschpartner wieder ...
  - 4.3) Bearbeite weiter nur noch die neue Restliste (=ohne vorderste Stelle).

Suche den <b>alphabetisch ersten</b> Namen in <b>der</b> Liste	
Tausche <b>diesen</b> ersten Namen nach vorn. und Tauschpartner von vorn auf <b>seinen alten Platz</b> ..	
<b>Restliste ist die</b> Liste ohne 1.Platz	
<b>solange</b> in <b>der jeweiligen</b> Restliste noch Einträge sind :	
<b>wdh:</b>	Suche den <b>alphab.ersten</b> Namen der Restliste
	Tausche <b>ihn</b> in der Restliste nach vorn und Tauschpartner von vorn auf <b>seinen alten Platz</b> ..
	<b>Neue Restliste ist die</b> Restliste ohne vordersten Platz
Gib die zurechtgetauschte <b>alte</b> Liste aus.	

A13) E-V-A - Prinzip meint:

Eingabe der Daten; Verarbeitung der Daten; Ausgabe der Resultate.

In diesen Fragen steht Prozess, weil hier die Ausführung gemeint ist, das Abarbeiten des Algorithmus.

Zum Begriff Prozess:

Ein Prozess in der Informatik ist der Vorgang einer algorithmisch ablaufenden Informationsbearbeitung. Grob also die Ausführung des Algorithmus. Die Unterscheidung wird vor allem dann interessant, wenn mehrere Prozesse ablaufen und sich gegenseitig beeinflussen.

A14) **Der größte Freiraum:**

Grobidee:

1) Den Text Zeichen für Zeichen prüfen und dabei Anzahl des Auftretens des jeweiligen Buchstabens notieren.

2) Für alle Zeichen, die mind. 2 mal vorkommen:

Text bis zum 1. Auftreffen dieses Zeichens durchlaufen;

Durchzählen bis zum nächsten Auftreffen (=Entfernung ermitteln und merken);

3) Bis zum Textende weiter den Text durchzählen; dabei:

**wenn** Buchstabe auftritt, **dann** Entfernung ermitteln

**wenn** neue Entfernung kleiner als gemerkte, **dann** die neue merken

4) Für alle Zeichen, die mind. zweimal vorkommen, die (endgültige) Entfernung vergleichen und das Zeichen mit der kleinsten Entfernung ausgeben.

Diese Idee macht deutlich, dass hierbei zwei Arten von Gegenständen vorkommen: der ganze Text und die Zeichen des Textes. Jedes Zeichen hat dabei Eigenschaften wie u.a. Häufigkeit des Auftretens, Entfernung.

Das Verfahren scheint selten auftretende Buchstaben zu bevorzugen. Häufig auftretende werden eher einmal in der Nähe voneinander vorkommen. Vielleicht wäre dann der relative Freiraum ein geeigneteres Maß.

## GK Informatik 12

## Klausur Nr.1

SJ 2000/2001

**Unterrichtsverlauf:**

1.DStd: Organisatorisches; Übersicht; Alltagsalgorithmen (Computerraum nicht verfügbar)

2./3./4.Woche sind wegen Fachkonferenz, Fortbildung, Feiertag ausgefallen

2.DStd: Rechnernutzung an der Schule; Benutzeroberflächen; (Delphi-)Programme ausführen; Sprachelemente von Algorithmen;

3.DStd: Delphi-Oberfläche; Beispielprogramme unter Delphi(Turtle); Struktogramme zur Beschreibung von (Alltags-)Algorithmen

4.DStd: Quelltexte und Formulare in Delphi; Lesen von Quelltexten(Turtle); Ändern von Quelltexten(Turtle); Fragen zum Skript

5.DStd: Klausur 1, also nach 4 Unterrichtsdoppelstunden(!)  
Die eingescannt wiedergegebene Tabelle bei Aufg.2 als auch das Struktogramm bei den Lösungen waren in der Vorlage für die Schüler direkt als Kopie aus der Zeitung bzw. Originalzeichnung vorhanden (in deutlich besserer Qualität).

**Bemerkungen zur Klausur:**

Aufgabe Nr.1 bezog sich auf das ausgegebene Skript und prüfte u.a. ab, wie weit es gelesen und bearbeitet wurde.

Aufgabe Nr.2 sollte (an einem Alltagsbeispiel) zeigen, wie die entscheidende Bedingung herausgefiltert wird und dass Bedingungen ineinander verwoben sind. Es wurde meist sauber erkannt, dass der HSV gar nicht mehr erster werden kann. Das Lesen der Zeitungstabelle führte zu Schwierigkeiten. (Lt. Schüleraussagen ist die Darstellung nicht korrekt, weil bei Punktgleichheit der direkte Vergleich zählt! Diese weite Verschachtelung machte wie erwartet Probleme.)

Der erste Teil bei Aufgabe Nr.3 diente dazu, den Algorithmus selbst zu durchlaufen, um ihn anschließend sicherer formulieren zu können.

Gerade bei Aufgabe Nr.4 wirkte sich die bis dato geringe Erfahrung der Schüler aus. Dies wurde bei der Bewertung der Antworten berücksichtigt (u.a. Zusatzpunkte). Aufg.4 wurde im Schnitt am schlechtesten gelöst. Insbesondere Teil 4c fiel noch schwer.

Alle 13 Schüler haben mitgeschrieben. Klausur war konzipiert für 60-75min Bearbeitungszeit. Einige Schüler brauchten fast 90 min.

Durchschnitt 10,0NP;

Verteilung:

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	NP
2	-	1	1	1	2	2	1	1	1	1	-	-	-	-	-	Anzahl

GK Informatik 12

Nr.1

7.11.2000

NAME: \_\_\_\_\_

- 1a) Der Begriff INFORMATIK ist ein Kunstwort. Woraus? Was soll damit ausgedrückt werden?
- b) Was versteht man unter einem Algorithmus? Was unterscheidet Alltagsalgorithmen von denen der Informatik?
- c) "Lehrer Lämpel ist ganz fortschrittlich. Er führt sein Notenbuch mit dem Computer." Was bedeutet in dieser Aussage "mit dem Computer"?

2) **"Wieder Chance für den HSV"**

so titelten die Zeitungen nach dem 3:1 Sieg in Turin.

Nutzen Sie die Tabelle und geben Sie in Form eines Struktogrammes an, unter welchen Bedingungen der HSV in seiner Gruppe erster oder zweiter wird (und damit in der Champions League weiterkommt) Für einen Sieg gibt es 3 Punkte (letzte Spalte), für ein Unentschieden 1 Punkt. Bei Punktgleichheit entscheidet die bessere Tordifferenz über den besseren Platz. Bei Tordifferenzgleichheit die Anzahl der geschossenen Tore.

Champions League				
Gruppe E				
Juventus Turin - Hamburger SV				3
Deportivo La Coruña - Panathinaikos Athen				1
1 Deportivo La Coruña	5	2	0	5:0 = 3
2 Juventus Turin	3	1	2	3:1 = 6
3 Hamburger SV	3	1	2	3:3 = 3
4 Panathinaikos Athen	5	1	2	5:4 = 1

Letzter Spieltag: Hamburger SV - Deportivo La Coruña, Panath. Athen - Juventus Turin JME 8/11, 20:45 Uhr.

3) **"Das runde Dutzend"**

In einem Buch mit Knobelaufgaben findet sich (so ähnlich) diese hier:

- a) Suchen Sie in der Zahlenfolge jeweils aufeinanderfolgende Zahlen, deren Summe 12 ergibt. Markieren Sie sie geeignet. Wieviele solche 'Quersummen' 12 finden Sie?
- 3 2 3 2 6 1 7 8 1 3 4 5 6 4 8 2 7 7 5 4 2 4 6 2 1 3 4 4 9 1 2 4 4 5
- b) Beschreiben Sie, wie Sie vorgegangen sind, um die Aufgabe zu lösen.
- c) Entfernen Sie an einer Stelle obiger Folge eine Zahl, so dass die 'Quersumme' 12 weniger häufig auftritt. Welche Stelle wählen Sie?

4) **Delphi**

- a) Diese Programmierumgebung trennt (anfangs) zwei Teile der Programmerstellung. Welche?
- b) Auf der Rückseite finden Sie den Quelltext des Programms KLAUS1.  
Was zeichnet die Turtle bei Anklicken des Schalters mit dem nichtssagenden Namen Button1 ?
- c) Was zeichnet die Turtle wenn der Schalter BFast betätigt wurde?  
Wie ändert sich die Zeichnung, wenn die Zeile 32 auskommentiert wird?  
Wie ändert sich die Zeichnung, wenn in Zeile 34 RT(2); auskommentiert wird?
- d) Warum sind Zeile 35 und 39 ungeschickt? Wie sollte man dies besser machen?

-----

.../ 24 Punkten

NOTE:

zu Aufgabe Nr.4

```
1  unit Klaus1;
2  interface
3  uses
4      SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
5      Forms, Dialogs, StdCtrls, ExtCtrls, Turtle;
6
7  type
8      TForm1 = class(TForm)
9          Turtle1: TTurtle;          Panell: TPanel;
10         BFast: TButton;           BNeu: TButton;
11         Button1: TButton;        BEnde: TButton;
12         procedure BEndeClick(Sender: TObject);
13         procedure BFastClick(Sender: TObject);
14         procedure BNeuClick(Sender: TObject);
15         procedure Button1Click(Sender: TObject);
16     private { Private-Deklarationen }
17     public  { Public-Deklarationen }
18     end;
19
20 var Form1: TForm1;
21
22 implementation {$R *.DFM}
23
24 procedure TForm1.BEndeClick(Sender: TObject);
25 begin close end;
26
27 procedure TForm1.BFastClick(Sender: TObject);
28     var seitenlaenge: INTEGER;
29 begin
30     seitenlaenge:=100;
31     with Turtle1 do begin
32         { RT(90); FD(-200); }
33         while seitenlaenge < 140 do begin
34             PU; {RT(2);} FD(10); PD;
35             Pen.Color:=clgreen;
36             FD(seitenlaenge); RT(120);
37             FD(seitenlaenge); RT(120);
38             FD(seitenlaenge); RT(120);
39             Pen.Color:=clblack;
40             inc(seitenlaenge, 2);
41         end;
42     end;
43 end;
44
45 procedure TForm1.BNeuClick(Sender: TObject);
46 begin
47     Turtle1.CS;
48     Turtle1.HT;
49 end;
50
51 procedure TForm1.Button1Click(Sender: TObject);
52 begin
53     with Turtle1 do begin
54         FD(80); RT(90);
55         FD(50); RT(90);
56         FD(80); RT(90);
57         FD(50); RT(90);
58     end;
59 end;
60
61 end.
```

## GK Informatik 12

## Nr.1

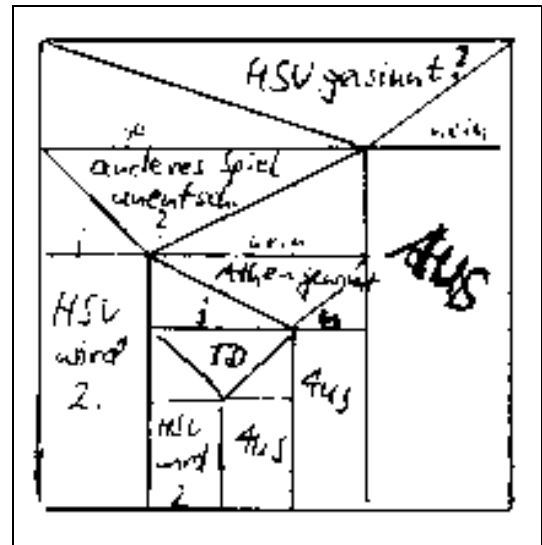
7.11.2000

Lösungen

- 1a) vgl. Skript und Programm: Animat2p
- b) vgl. Skript; Alltagsalgorithmen sind nicht vollständig präzise; sie bauen oft auf das Verständnis und die Erfahrung des Adressaten; sie benutzen nicht ein fest definiertes Sprachniveau.
- c) d.h. dass er geeignete Hard- **und** Software benutzt.

- 2) Es gibt nur eine Chance, wenn der HSV das letzte Spiel gewinnt. Dies wird die erste Frage (Entscheidung) sein. (P.S.: nicht gewinnen heißt: Unentschieden oder verlieren!)

Danach (also nur falls der HSV gewonnen hat!) geht es darum, ob einer im anderen Spiel gewinnt. Wenn es dort ein Unentschieden gibt, wird der HSV 2. Gewinnt Turin, ist der HSV draußen; gewinnt Athen, so kommt es auf die Tordifferenz an. So weit reichte es hier ... Die Schüler konnten den Rest offen lassen. Er zählte nur als Zusatz. Die Abfrage TD bedeutet: Hat HSV eine bessere Tordifferenz als Athen? Im Nein-Fall muss bei Gleichheit weiter nach der Anzahl der geschossenen Tore untersucht werden.



- 3a) 10 Mal tritt die Summe 12 auf.

3 2 3 2 6 1 7 8 1 3 4 5 6 4 8 2 7 7 5 4 2 4 6 2 1 3 4 4 9 1 2 4 4 5  
 3 2 3 2 6 1 7 8 1 3 4 5 6 4 8 2 7 7 5 4 2 4 6 2 1 3 4 4 9 1 2 4 4 5

(Ein mehrfaches, farblich unterschiedenes Unterstreichen zur Markierung gelingt mir per Hand besser als per Textverarbeitung. Daher hier diese Notlösung.)

- b) Ich beginne bei der ersten Zahl links vorne. Ich laufe solange von ihr aus nach rechts wie die Summe kleiner oder gleich 12 bleibt oder ich das Ende der Zahlenfolge treffe. Erreiche ich genau 12, so markiere ich diese Zahlen. Sonst gibt es mit dieser Anfangszahl keine Summe 12. Es kann nie mehr weniger werden.

Nun führe ich gleiches mit jeder Zahl als Startzahl der Summe durch; bis zur vorletzten als Startzahl. (Die letzte Zahl ist auch kleiner als 10 und damit kleiner als 12).

Kurz: Nacheinander wird jede Zahl zum Anfang einer Summe, die solange nach rechts läuft, bis 12 erreicht oder überschritten ist. Wird genau 12 erreicht, markiert man die Zahlen.

- c) Es gibt mehrere (ich denke: 5) Möglichkeiten 2 der 10 aufgefundenen 12er-Summen zu vernichten. Durch Entfernen der 6 in 2 4 6 2 1 3, die sogar in drei 12-Summen vorkommt, werden auch nur zwei Möglichkeiten vernichtet, weil drei verschwinden, aber eine neu entsteht. 2 4 2 1 3 !!

3 2 3 2 6 1 7 8 1 3 4 5 6 4 8 2 7 7 5 4 2 4 6 2 1 3 4 4 9 1 2 4 4 5

Ebenso entsteht beim Entfernen der 4 in 4 6 2 1 3 eine neue 12-Summe: 4 2 \_ 6 .

## GK Informatik 12

## Nr.1

## Lösungen S.2

- 4a) vgl. Unterricht / Folien  
 Delphi trennt die  
 Gestaltung im Formular - textliche Erstellung im Editor  
 WIE soll's aussehen - WAS soll passieren  
 Delphi nutzt getrennte Dateien, um diese beiden Teile abzuspeichern.  
 Beides wird durch einen Rahmen, die sog. Projektdatei, zusammengehalten.
- b) Button1Click: Die Turtle zeichnet ein Rechteck (80x50Pixel groß)
- c) BFastClick: Die Turtle zeichnet mehrere gleichseitige, grüne Dreiecke. Die Seitenlänge der Dreiecke ist anfangs 100 Pixel und wird immer um 2 Pixel größer solange sie unter 140 bleibt. Die Startpunkte der Dreiecke werden jeweils um 10 Pixel nach oben verschoben.  
 Wird Z.32 auskommentiert, also benutzt, so dreht sich die Schildkröte vorm ersten Zeichnen und beginnt nicht in der Mitte sondern ein Stück links davon. Die Dreiecke liegen damit nebeneinander statt übereinander versetzt. Wird in Zeile 34 noch RT(2); auskommentiert, so werden die Dreiecke leicht gegeneinander verdreht (jeweils 2°) gezeichnet. Es bildet sich eine "Kurve".
- d) Es wird ständig die Farbe gewechselt, aber immer nur grün zum Zeichnen benutzt. Daher sollte man vor der Schleife die Farbe auf Grün setzen und nach der Schleife wieder zurück auf schwarz. In der Schleife ist dies unnötig und damit Ressourcenverschwendung.

-----  
 Reserve-Bild zur ChampionsLeague-Tabelle:

Champions League,					
Gruppe E					
Juventus Turin - Hamburger SV					1:0
Deportivo La Coruña - Panathinaikos Athen					1:0
1 Deportivo La Coruña	3	2	3	0	5:3
2 Juventus Turin	5	1	0	1	6:0
3 Hamburger SV	1	1	2	2	4:3
4 Panathinaikos Athen	1	1	0	2	3:4
Letzter Spieltag: Hamburger SV - Deportivo La Coruña. Panath. Athen - Juventus Turin (M: 8.11., 20:45 Uhr)					