Algorithmen und Datenstrukturen 1

Prof. Dr. Carsten Lecon

Algorithmen und Datenstrukturen

- … lassen sich auch in drei Minuten erklären ;-)

Algorithmen & Datenstrukturen in 3 Minuten.avi

Quelle: https://www.youtube.com/watch?v=wVT62RtA248

Inhalt I

- Grundbegriffe
  - Der Algorithmenbegriff
  - Beispiele
  - Die Rolle von Algorithmen in der Informatik
- Notation von Algorithmen
  - Einleitung
  - Programmablaufplan
  - Struktogramm
  - Pseudocode

Inhalt I

- Grundbegriffe
  - Der Algorithmenbegriff
  - Beispiele
  - Die Rolle von Algorithmen in der Informatik
- Notation von Algorithmen
  - Einleitung
  - Programmablaufplan
  - Struktogramm
  - Pseudocode

Lernziele (Algorithmen)

- Ziele
  - Verständnis der Notwendigkeit von Algorithmen
  - Kenntnis der Eigenschaft von Algorithmen
  - Kennen von Beispielen von Algorithmen
  - Kenntnis der Notation von Algorithmen

Was ist ein Algorithmus?

Zutaten für 3 **Portionen** umrechnen

250 g Butter
250 g Zucker
1 Pck. Vanillezucker
5 **Ei(er)**
375 g Mehl
½ Pck. Zitrone(n) - Schale, agerieben oder Citroback
5 EL Milch
Puderzucker, zum Bestäuben
2 ½ TL Backpulver

Zubereitung

Butter schaumig rühren, Zucker und Vanillinzucker dazu, Eier einzeln dazu rühren, Citroback od. Zitronenschale, gesiebtes Mehl, Backpulver und die Milch unterrühren und den Teig auf drei gut gefettete und mit Bröseln ausgestreute Formen (z.B. Lamm, Hase, Henne od. Hahn) verteilen.
Im vorgeheizten Backofen bei 200 Grad (Umluft 180 Grad) ca. 40 Minuten backen. Die Figuren nach dem Backen vorsichtig aus der Form lösen und auf einem Kuchengitter auskühlen lassen. Dick mit Puderzucker bestäuben.

Arbeitszeit:	ca. 20 Min.
Schwierigkeitsgrad:	simpel
Brennwert p. P.:	keine Angabe
Freischaltung:	23.04.03
Rezept-Statistiken:	306.433 (5.045)* gelesen
	2.797 (82)* gespeichert
	25.002 (501)* gedruckt
	419 (8)* verschickt
	* nur in diesem Monat

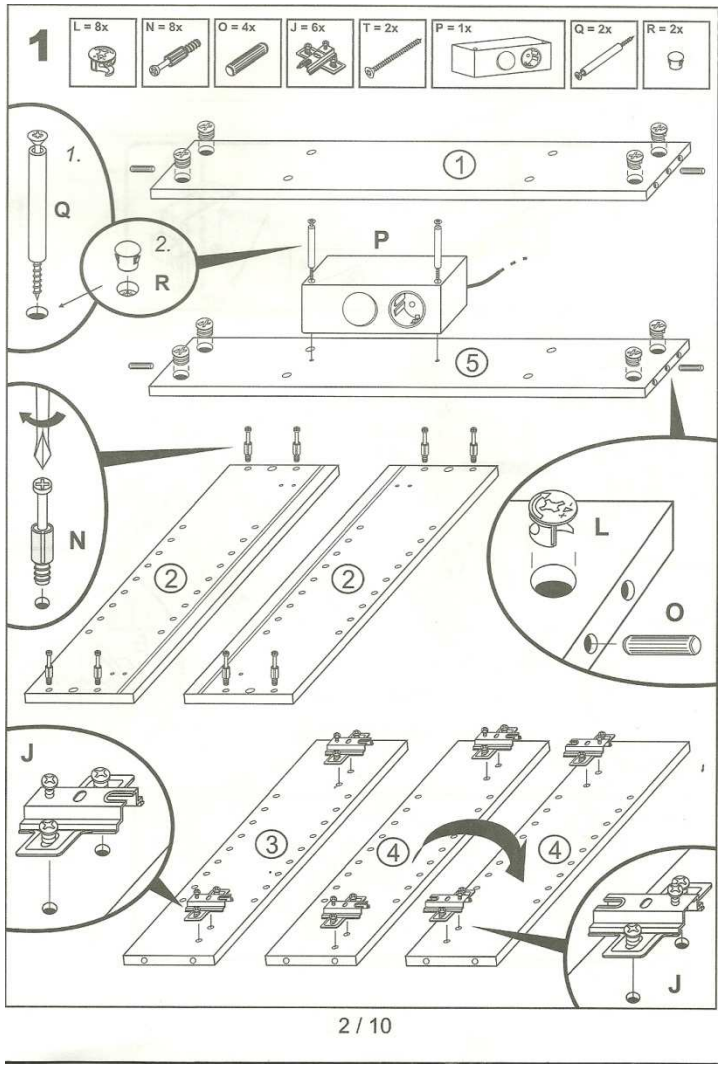
Daten

Algorithmus

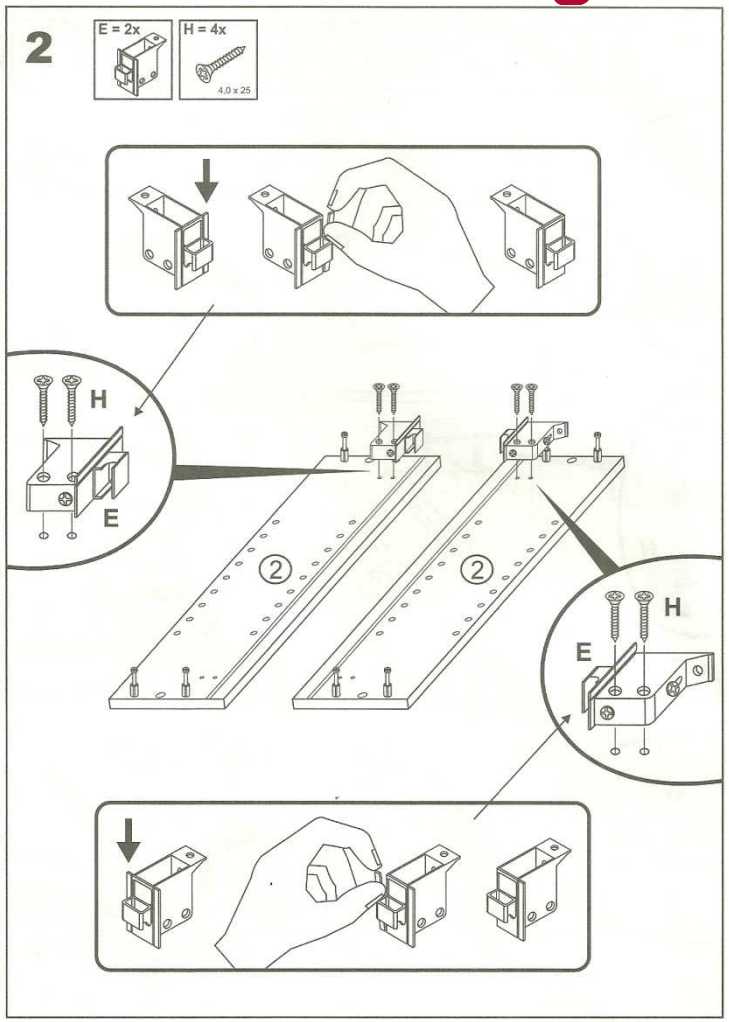
(Zeit-) Komplexität

Komplexität

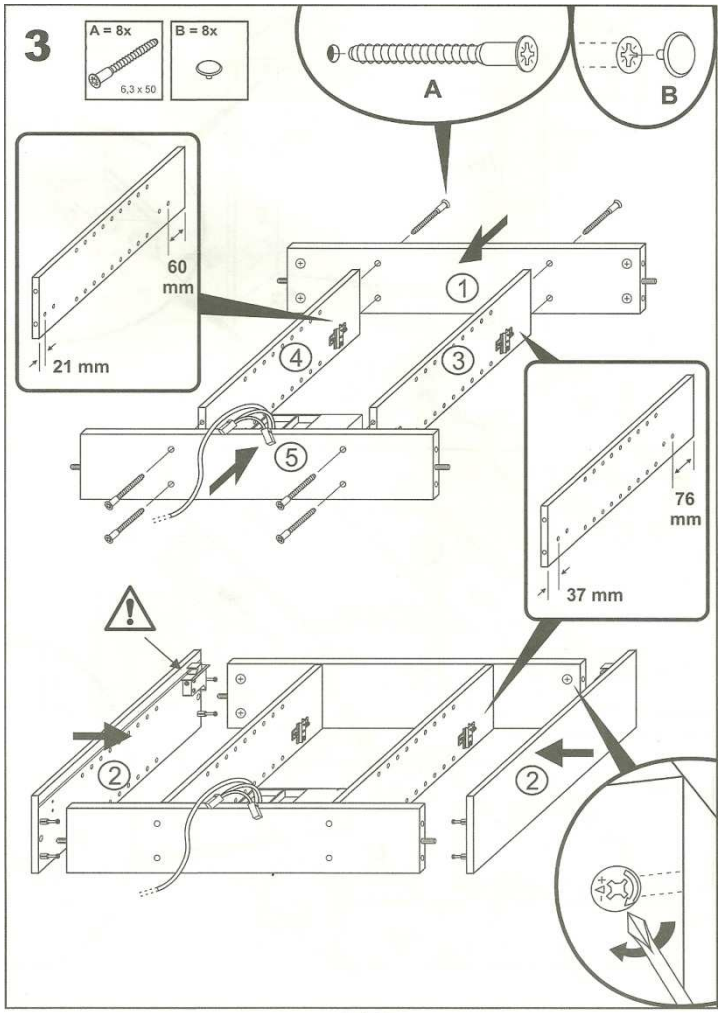
Was ist ein Algorithmus?



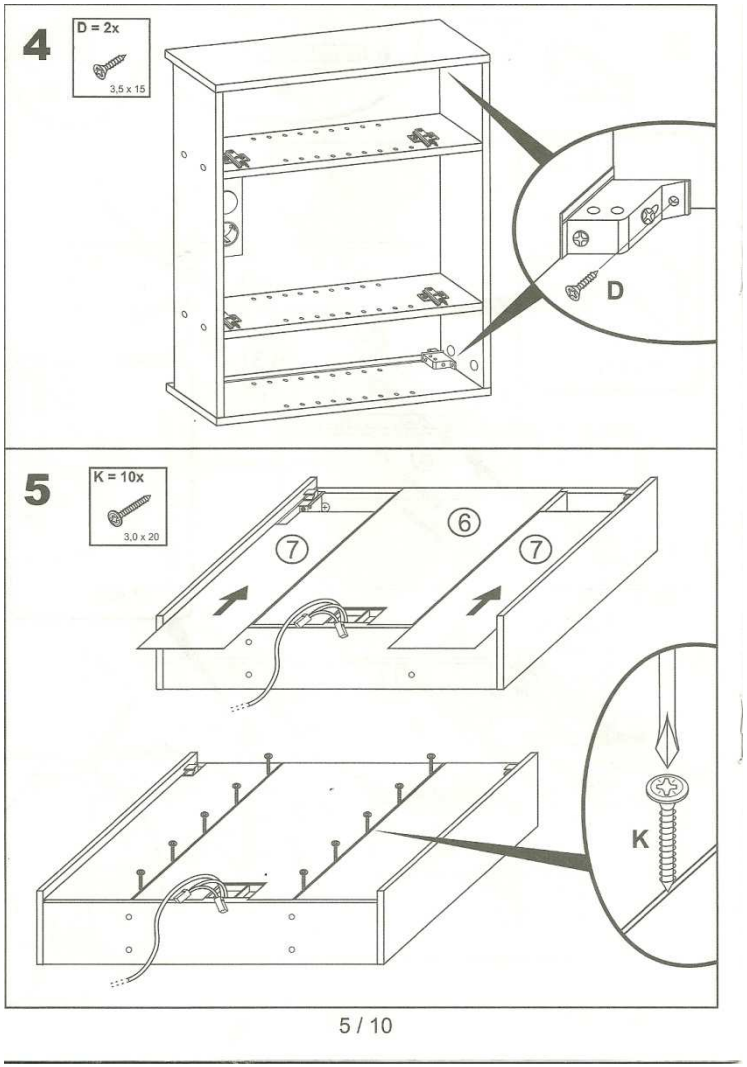
Was ist ein Algorithmus?



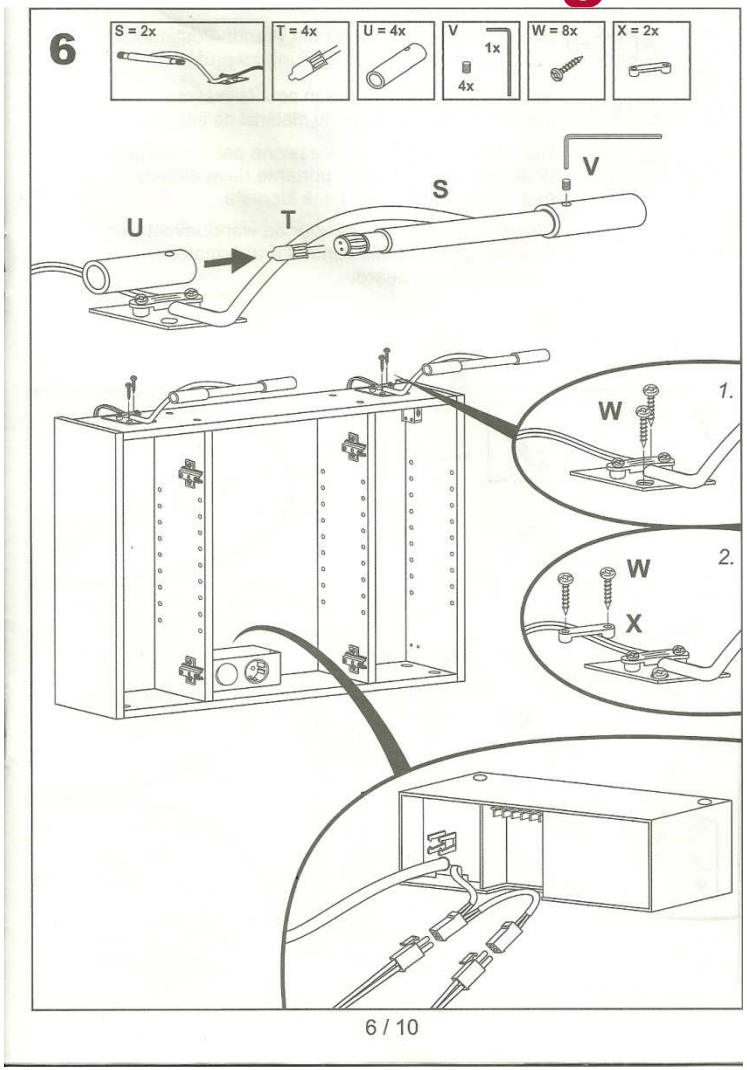
Was ist ein Algorithmus?



Was ist ein Algorithmus?



Was ist ein Algorithmus?



Was ist ein Algorithmus?

- Was sind die Eigenschaften dieses Rezeptes und dieser Bauanleitung?
- Wie lässt sich das auf die Informatik übertragen?

Was ist ein Algorithmus?

- Definition Nr. 1:
  - Ein Verfahren zur Lösung eines Problems, das für eine Realisierung in Form eines Programms geeignet ist.

    (Sedgewick: Algorithmen in C)

Was ist ein Algorithmus?

- Definition Nr. 2:
  - Eine Berechnungsvorschrift zur Lösung eines Problems heißt genau dann Algorithmus, wenn zu dieser Berechnungsvorschrift eine äquivalente Turingmaschine existiert, die für jede Eingabe, die eine Lösung besitzt, stoppt.

Was ist ein Algorithmus?

- Definition Nr. 3:
  - Folge von exakten Arbeitsanweisungen zum Lösen einer Rechenaufgabe in endlich vielen, eindeutig festgelegten Schritten. Jede mit einem Algorithmus lösbare Aufgabe kann prinzipiell auch von einem Rechenautomaten gelöst werden.

    (Meyers Taschenlexikon)

Was ist ein Algorithmus?

- Definition Nr. 4:
  - Ein Algorithmus ist eine eindeutige, endliche Beschreibung eines allgemeinen, endlichen Verfahrens zur schrittweisen Ermittlung gesuchter Größen aus gegebenen Größen.

Was ist ein Algorithmus?

- Definition Nr. 5:
  - Ein Algorithmus ist eine präzise, d.h. in einer festgelegten Sprache abgefasste, endliche Beschreibung eines schrittweisen Problemlösungsverfahrens zur Ermittlung gesuchter Größen aus gegebenen Größen, in dem jeder Schritt aus einer Anzahl ausführbarer Aktionen und einer Angabe über den nächsten Schritt besteht.

    (Rechenberg, Pomberger (eds.): Handbuch Informatik)

Was ist ein Algorithmus?

- Definition Nr. 6:
  - Unter einem Algorithmus versteht man eine Verarbeitungsvorschrift, die so präzise formuliert ist, dass sie von einem mechanisch oder elektronisch arbeitenden Gerät durchgeführt werden kann.

    (Schülerduden Informatik, 1986)

Was ist ein Algorithmus?

- Herkunft des Names:
  - Abu Abdallah Muhammead ibn Musa al-Chwarizmi (al Charazmi)
  - Persischer Mathematiker des Mittelalters (ca. 783 bis 850)
  - Rechnen mit indischen Ziffern
  - Die lateinische Übersetzung eines seiner Werke beginnt mit „Dixit Algorismi“
  - Die Briefmarke entstand 1983 anlässlich seine 1200. Geburtstages.

Bildquelle: http://de.wikipedia.org/wiki/Al-Chwarizmi#mediaviewer/File:1983_CPA_5426_(1).png

Eigenschaften von Algorithmen

- Determiniertheit:
  - Bei gleichen Startbedingungen und gleicher Eingabe wird immer die gleiche Ausgabe geliefert.
- Determinismus:
  - Nach jedem Schritt steht der nächste Schritt eindeutig fest.
- Statische Finitheit:
  - Algorithmus kann in endlicher Länge beschrieben werden.
- Dynamische Finitheit:
  - Ablauf des Algorithmus benötigt zu jedem Zeitpunkt der Ausführung nur begrenzt viel Speicherplatz.
- Vollständigkeit:
  - Vollständige Beschreibung eines Lösungsverfahrens

Eigenschaften von Algorithmen

- Frei nach Vornberger:
  - „Die drei K´s“
    - **K**omplexität
      - Zeiteffizienz
      - Speichereffizienz
    - **K**orrektheit
    - **K**ann das Programm anhalten?
      - Terminiertheit

Eigenschaften von Algorithmen

- Komplexität:
  - Zeiteffizienz:
    - Terminiert nach einer „günstigen“ Zeit
  - Speichereffizienz:
    - Benötigt „wenig“ Speicherplatz

Eigenschaften von Algorithmen

- Korrektheit:
  - Terminiert bei jeder erlaubten Eingabe mit der korrekten Ausgabe
- Terminiertheit:
  - Terminiert bei jeder erlaubten Eingabe nach endlich vielen Schritten

Ältester Algorithmus

- Bekanntester, nicht-trivialer Algorithmus:
  - Euklischer Algorithmus
    - Berechnung ggT

Inhalt I

- Grundbegriffe
  - Der Algorithmenbegriff
  - Beispiele
  - Die Rolle von Algorithmen in der Informatik
- Notation von Algorithmen
  - Einleitung
  - Programmablaufplan
  - Struktogramm
  - Pseudocode

Beispiele für Algorithmen

Kürzen eines Bruches p/q:

1. Primfaktorzerlegung des Zählers
2. Primfaktorzerlegung des Nenners
3. Suche in Zähler und Nenner die größte gemeinsame Sequenz und streiche sie jeweils

Beispiele für Algorithmen

Kürzen eines Bruches p/q:

1. Bestimme den größten gemeinsamen Teiler (ggT) g von p und q
2. p´=p/g
3. q´=q/g
4. Der gekürzte Bruch lautet p´/q´

ggT? → weiterer Algorithmus

Beispiele für Algorithmen

Größter gemeinsamer Teiler (ggT)

Gegeben: a, b > 0

Gesucht: ggT(a,b)

Solange a,b>0: Ersetze die größere Zahl durch den Rest der Division.

Am Ende ist die größere Zahl von a und b der gesuchte ggT.

Beispiele für Algorithmen

Anekdote von Gauß

Algorithmus 1:

$$\sum_{i=1}^{n} i = 1 + 2 + 3 + \cdots + (n-1) + n$$

Algorithmus 2:

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

Inhalt I

- Grundbegriffe
  - Der Algorithmenbegriff
  - Beispiele
  - Die Rolle von Algorithmen in der Informatik
- Notation von Algorithmen
  - Einleitung
  - Programmablaufplan
  - Struktogramm
  - Pseudocode

Die Rolle der Algorithmen in der Informatik

- Warum lohnt es sich, sich mit Algorithmen zu befassen?
  - Typischerweise werden in einer Vorlesung eine Menge von Algorithmen vorgestellt, die später in eigenen Programmen verwendet werden können.
  - Dennoch: Man kommt in die Situation, dass ein vorgefertigter Algorithmus gerade nicht verfügbar ist.
  - Man muss also in der Lage sein, selber Algorithmen zu entwickeln.
  - Die Algorithmik bietet hierzu eine Reihe von Techniken für den Algorithmenentwurf an.

Die Rolle der Algorithmen in der Informatik

- Algorithmen
  - Sie sind das zentrale Thema der Informatik.
  - Sie sind eng mit Datenstrukturen verknüpft.
  - Entwurfstechniken sind interessant, zumindest aber wichtig.
  - Korrektheit muss gezeigt werden können.
  - Kenntnis über ihre Effizienz ist wesentlich.

Algorithmen als Technik

- Computer sind schnell, aber nicht unendlich schnell.
- Speicher ist nicht kostenlos.
- Algorithmen, die (deutlich) schneller sind oder die mit (deutlich) weniger Speicherplatz auskommen als andere, sind besser.
- Selbst wenn die Computer schneller werden oder die Speicher preiswerter, geschieht dies immer „nur“ mit konstanten Faktoren.
- Vor allem für „schwierige“ Probleme ist ein besserer Algorithmus mehr wert als ein schneller Rechner.

Inhalt I

- Grundbegriffe
  - Der Algorithmenbegriff
  - Beispiele
  - Die Rolle von Algorithmen in der Informatik
- Notation von Algorithmen
  - Einleitung
  - Programmablaufplan
  - Struktogramm
  - Pseudocode

Bausteine für Algorithmen

- Elementare Operationen
- Sequentielle Ausführung
  - Hintereinanderausführen von Schritten
- Parallele Ausführung
  - Gleichzeitiges Ausführen von Schritten
- Bedingte Ausführung
  - Ausführung nur unter einer bestimmten Bedingung
- Schleife
  - Wiederholung einer Tätigkeit bis Eintreten einer Endbedingung
- Unterprogramm
- Rekursion

Notation von Algorithmen

- Möglichkeiten:
  - Natürliche Sprache
  - Programmiersprache
  - Mathematische Formel
  - Programmablaufplan, Ablaufdiagramm
  - Struktogramm
  - Pseudocode

- Vor-/Nachteile?

Notation von Algorithmen

Mathematische Formeln

1. $\sum_{i=1}^{n} i = 1 + 2 + 3 + \cdots + (n-1) + n$
2. $|x| = \begin{cases} x & \text{falls } x >= 0 \\ -x & \text{falls } x < 0 \end{cases}$

Inhalt I

- Grundbegriffe
  - Der Algorithmenbegriff
  - Beispiele
  - Die Rolle von Algorithmen in der Informatik
- Notation von Algorithmen
  - Einleitung
  - Programmablaufplan
  - Struktogramm
  - Pseudocode

Notation von Algorithmen

- Programmablaufplan, Flussdiagramm
  - Spezifikation nach DIN 66001
  - Darstellung eines Ablaufs durch Symbole
  - Verknüpfung der Symbole durch Pfeile
  - Elemente: Grenzpunkte, Operationen, Unterprogramme, Alternative, Ein-/Ausgabe, Wiederholung

- Problem:
  - Unstrukturierte Abläufe darstellbar

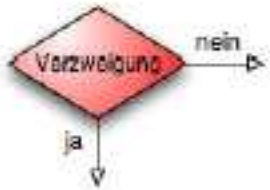
Notation von Algorithmen

Elemente von Programmablaufplänen

Grenzpunkte

Operationen

Verzweigungen



Notation von Algorithmen

Elemente von Programmablaufplänen

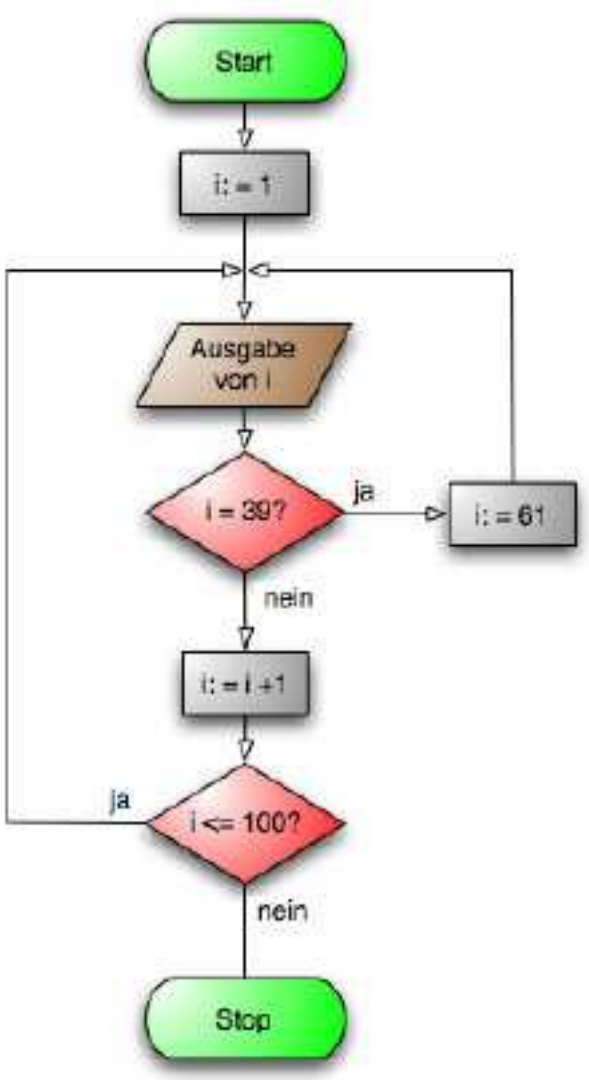
Unterprogramm

Ein-/Ausgabe

u.v.m.

Notation von Algorithmen

Beispiel Programmablaufplan



Inhalt I

- Grundbegriffe
  - Der Algorithmenbegriff
  - Beispiele
  - Die Rolle von Algorithmen in der Informatik
- Notation von Algorithmen
  - Einleitung
  - Programmablaufplan
  - Struktogramm
  - Pseudocode

Notation von Algorithmen

- Struktogramm/Nassi-Shneiderman-Diagramm
  - Spezifikation in DIN 66261
  - Ähnlich wie Programmablaufplan, aber Zwang zur Strukturierung
  - Elemente: Sequenz, Alternative, Wiederholung, Unterprogrammaufruf

Notation von Algorithmen

- Wozu Struktogramme?
  - Strukturiertes Vorgehen:
    - Erst die Struktur des Programms entwickeln, dann Programm schreiben
  - Vermeidung von Fehlern
    - Strukturiertes Vorgehen statt *trial and error*
  - Vor Schreiben des Programms macht man sich Gedanken zum Aufbau und Ablauf des Programms
  - Übersichtliche Darstellung unabhängig von einer Programmiersprache
  - Programme, die entsprechend eines Struktogramms implementiert werden, sind in ihrem Aufbau und ihrer Logik leichter überschaubar.

Quelle: http://www.math.tu-berlin.de/ppm/skripte/struktogramme2009.pdf

Notation von Algorithmen

- Struktogramme: Vorgehen:
  - Ein Struktogramm wird von oben nach unten gelesen.
  - Die Teilalgorithmen bzw. Anweisungen werden in Form von Rechtecken geschrieben.
  - Jedes Struktogramm hat jeweils einen Eingang (obere Kante) und einen Ausgang (untere Kante)
  - Der Ausgang eines Segements ist der Eingang des nachfolgenden Segments.
  - Man beginnt mit einem Grobentwurf und verfeinert diesen schrittweise, d.h. eine Anweisung wird durch einen Teilalgorithmus ersetzt.
  - Struktogramme können verschachtelt werden.
    Überall dort, wo eine Anweisung stehen darf, kann ein Teilalgorithmus oder ein ganzes Struktogramm eingefügt werden.

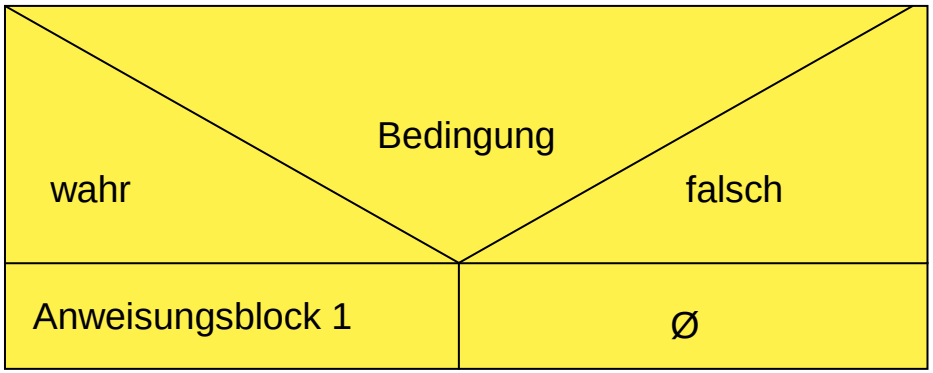
Quelle: http://www.math.tu-berlin.de/ppm/skripte/struktogramme2009.pdf

Notation von Algorithmen

- Struktogramm:
  - Sequenzieller, linearer Ablauf

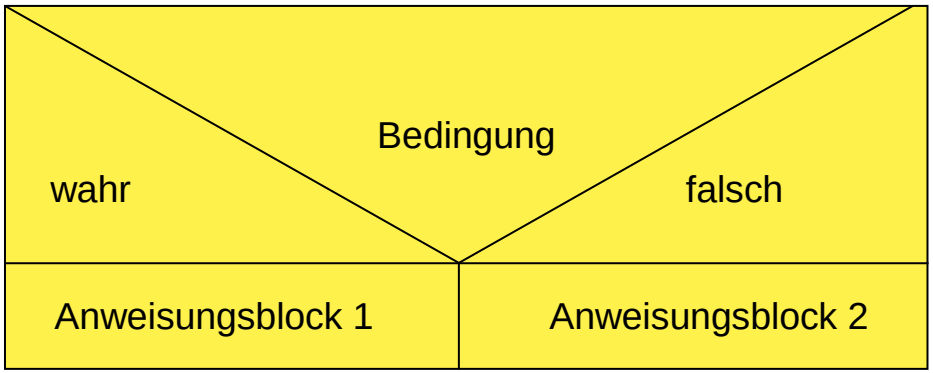
Notation von Algorithmen

- Struktogramm:
  - Einfache Alternative



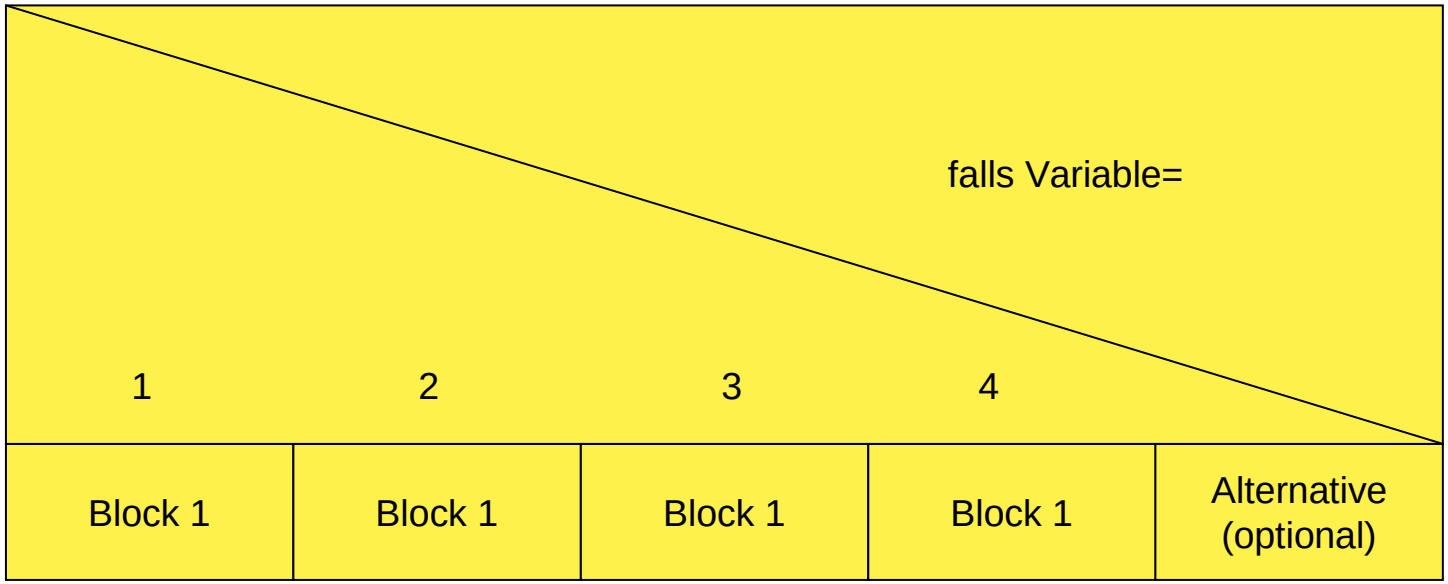
Notation von Algorithmen

- Struktogramm:
  - Zweifache Alternative



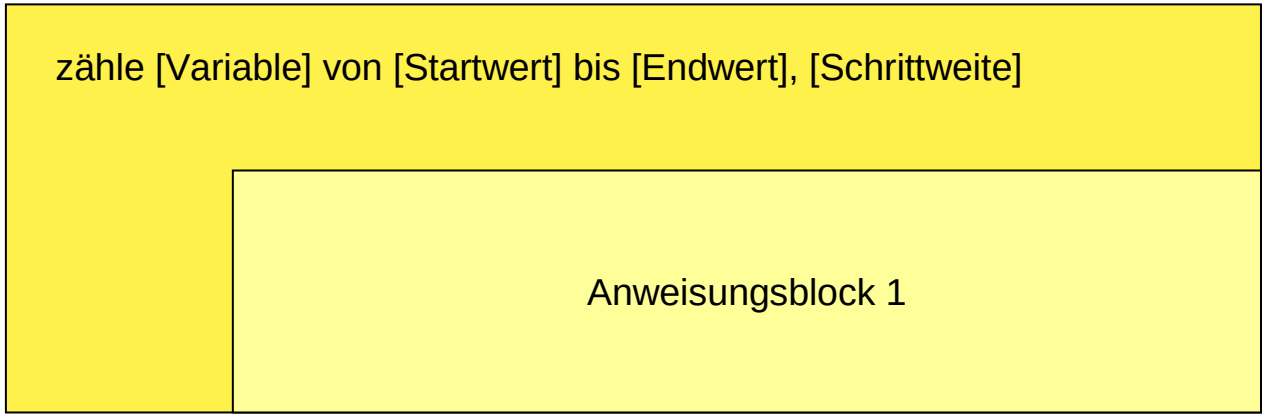
Notation von Algorithmen

- Struktogramm:
  - Fallauswahl



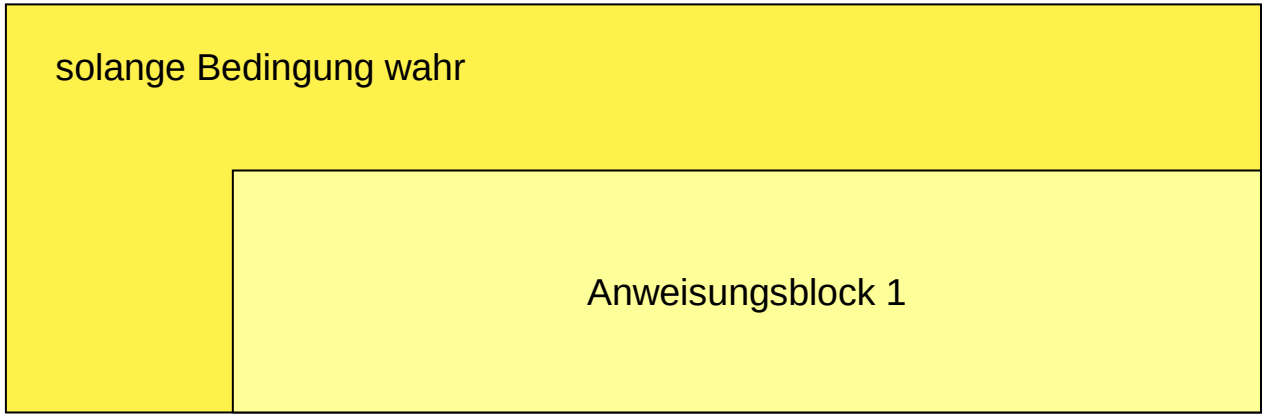
Notation von Algorithmen

- Struktogramm:
  - Iteration (zählergesteuert)



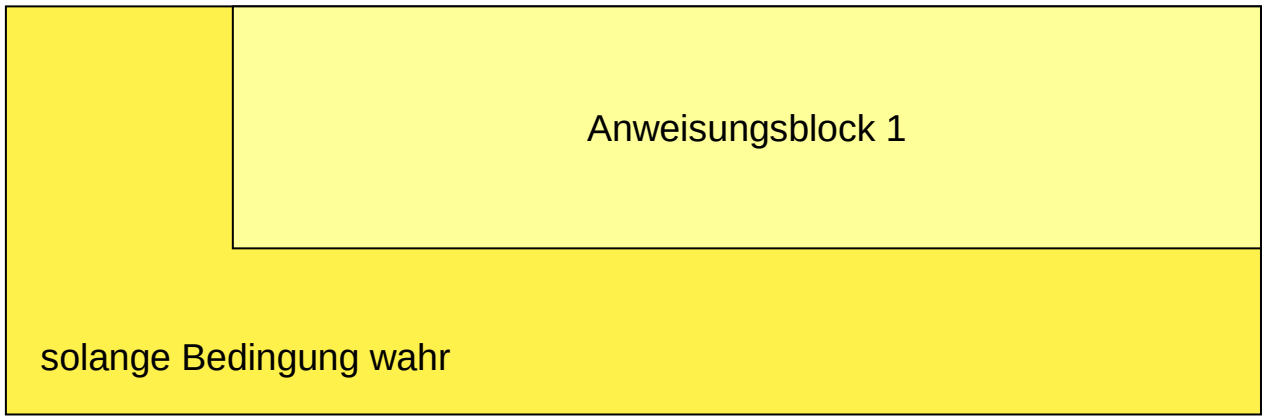
Notation von Algorithmen

- Struktogramm:
  - Abweisende (kopfgesteuerte) Schleife



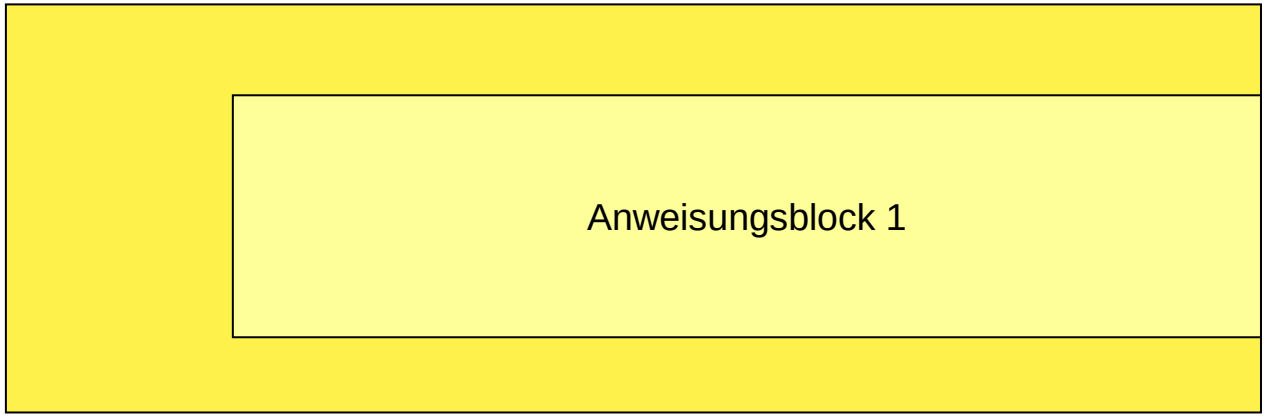
Notation von Algorithmen

- Struktogramm:
  - Iteration: nicht abweisende (fußgesteuerte) Schleife



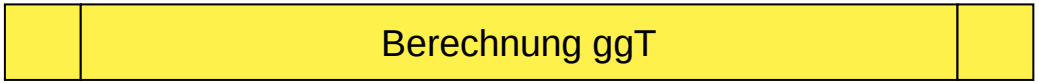
Notation von Algorithmen

- Struktogramm:
  - Iteration: Endlosschleife



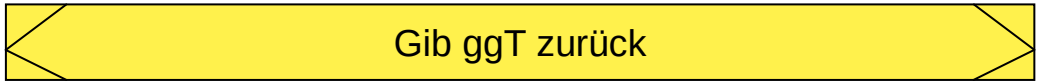
Notation von Algorithmen

- Struktogramm:
  - Unterprogrammaufruf



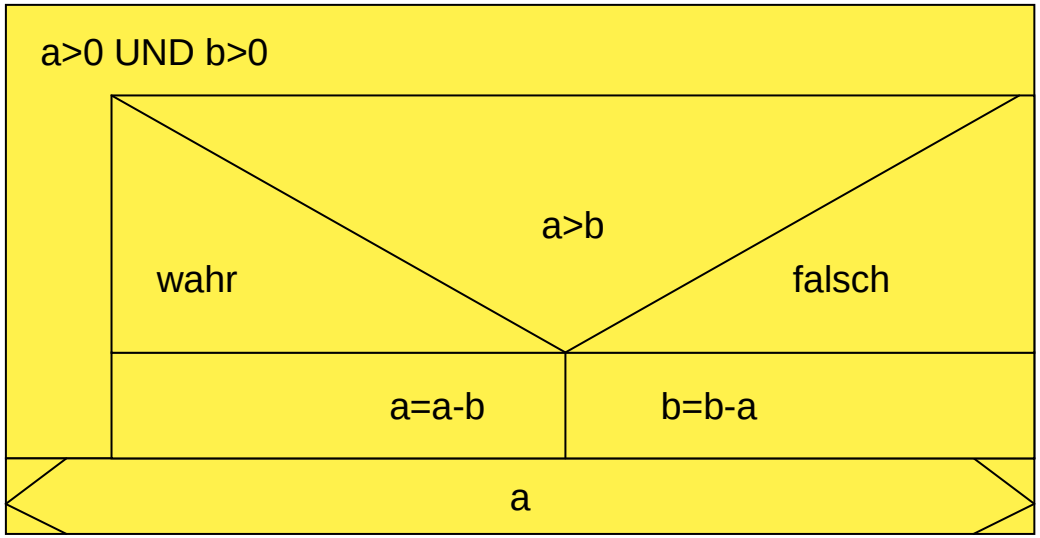
Notation von Algorithmen

- Struktogramm:
  - Rücksprung (aus Unterprogramm)

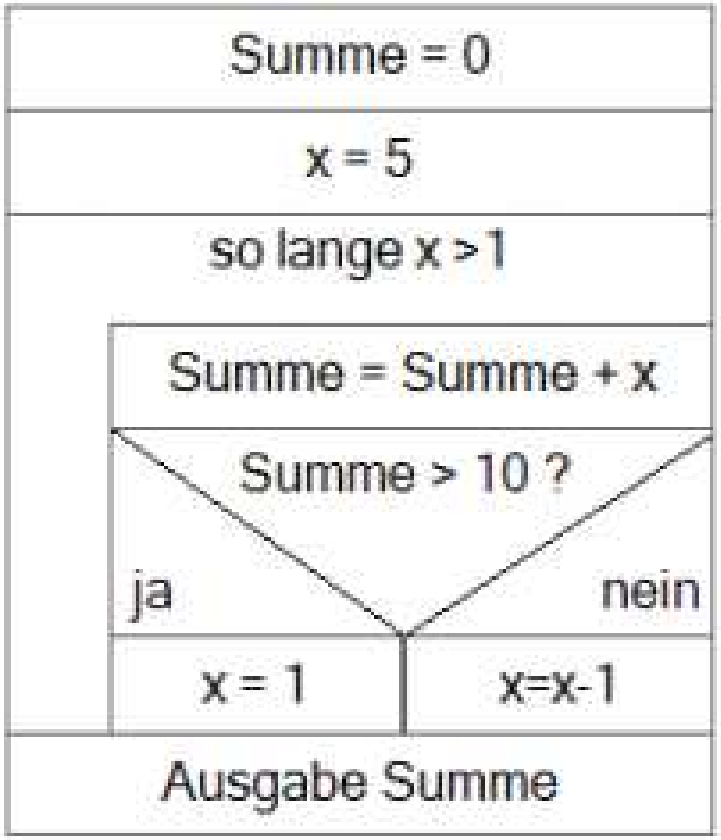


Notation von Algorithmen

- Struktogramm:
  - Beispiel: Unterprogramm zur Berechnung des ggT



Übung Struktogramm

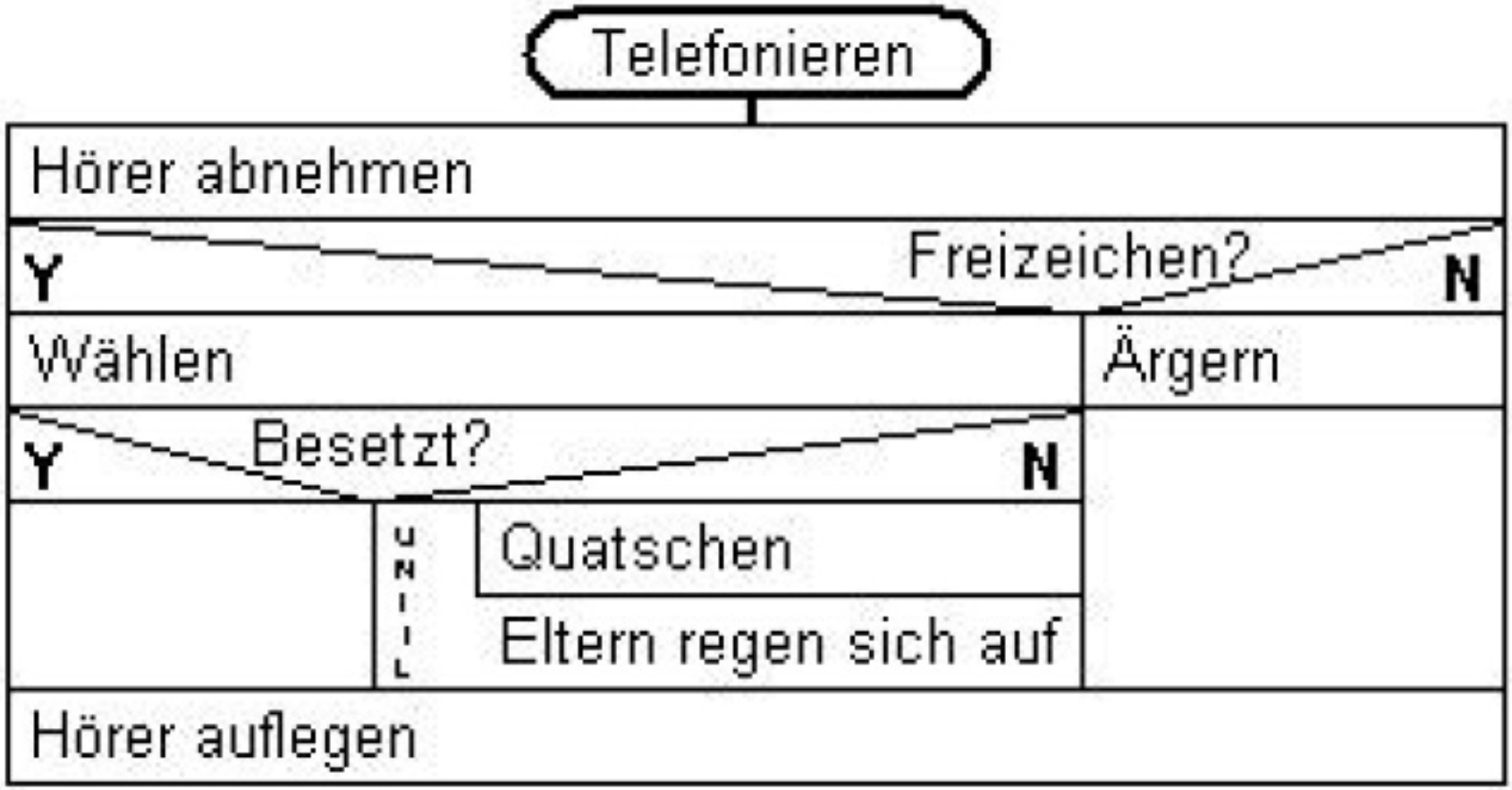


Quelle: http://www.pellatz.de/downloads/Struktogramme.pdf

Hörsaalübung

- Erstellen Sie einen allgemeingültigen Algorithmus für „Telefonieren“.
- Erstellen Sie das dazugehörige Struktogramm.

Hörsaalübung



Inhalt I

- Grundbegriffe
  - Der Algorithmenbegriff
  - Beispiele
  - Die Rolle von Algorithmen in der Informatik
- Notation von Algorithmen
  - Einleitung
  - Programmablaufplan
  - Struktogramm
  - Pseudocode

Notation von Algorithmen

- Pseudocode
  - Mischung als natürlicher Sprache und mathematischer Notation; manchmal auch Programmiersprache
  - Typischerweise leicht zu verstehen (für Menschen)
  - Strukturell einer Programmiersprache sehr ähnlich
  - Enthält keine Implementierungsdetails
  - Es gibt hierfür keine (Industrie-) Norm

Notation von Algorithmen

- Pseudocode
  - Sequenz, linearer Ablauf

```
1: Anweisung 1
2: Anweisung 2
3: Anweisung 3
4: Anweisung 4
```

Notation von Algorithmen

- Pseudocode:
  - Einfache Alternative

```
1: if Bedingung then
2:     Anweisungsblock
3: end if
```

Notation von Algorithmen

- Pseudocode:
  - Zweifache Alternative

```
1: if Bedingung then
2:     Anweisungsblock 1
3: else
4:     Anweisungsblock 2
5: end if
```

Notation von Algorithmen

- Pseudocode:
  - Iteration (zählergesteuert)

```
1: for Variable=Startwert
       bis Endwert
       mit Schrittweite do
2:         Anweisungsblock 1
3: end for
```

Notation von Algorithmen

- Pseudocode:
  - Iteration: Abweisende (kopfgesteuerte) Schleife

```
1: while Bedingung ist wahr do
2:         Anweisungsblock 1
3: end while
```

Notation von Algorithmen

- Pseudocode:
  - Iteration: Nicht abweisende (fußgesteuerte) Schleife

```
1: repeat
2:         Anweisungsblock 1
3: until Bedingung ist wahr
```

Notation von Algorithmen

- Pseudocode:
  - Endlosschleife

```
1: while wahr do
2:     Anweisungsblock 1
3: end while
```

Notation von Algorithmen

- Pseudocode:
  - Unterprogrammaufruf

```
1: z=ggT(a,b)
```

Notation von Algorithmen

- Pseudocode:
  - Beispiel Unterprogramm ggT

```
 1: function ggT(a,b)
 2:   while a>0 AND b>0 do
 3:     if a>b then
 4:       a=a-b
 5:     else
 6:       b=b-a
 7:     end if
 8:   end while
 9:   return a
10: end function
```

Zusammenfassung

- Algorithmen sind wesentlicher Bestandteil bei der Spezifikation von Software
- Algorithmen sind eng mit Datenstrukturen verknüpft
- Es gibt unterschiedliche Notationen für Algorithmen
- Wichtige Eigenschaften von Algorithmen sind: Komplexität, Korrektheit und Terminierung