

Kapitel 5: Algorithmen und Programmierung

Gliederung von Kapitel 5

1. Grundlegendes zu Algorithmen, einfache Beispiele
2. Nassi-Shneiderman-Diagramme (Struktogramme; formale grafische Darstellung von Algorithmen)
3. Einführung in das Programmieren mit VBA (Visual Basic for Applications) in Excel



Algorithmen & Programmierung

- **Algorithmus:** präzise Folge von Anweisungen
 - Schritt für Schritt („erst dies, dann das ...“)
 - z. B.: Kochrezept, Reparaturanleitung
 - Fallunterscheidungen („wenn die Kontrolllampe leuchtet, dann ... – anderenfalls ...“)
 - Wiederholungen („Klicken Sie solange auf Abbrechen, bis das Fenster verschwindet“)
- **Programm:** Beschreibung eines Algorithmus in einer speziellen, Computer-verständlichen Sprache

Einführung in Algorithmen (1/8)

- Algorithmen sind Schritt-für-Schritt-Anleitungen, nicht zwingend für einen Computer
- Beispiele
 - mathematische Problemlösungen, etwa:
 - Lösung für quadratische Gleichung finden
 - $n \times n$ -Matrix in Dreiecksform bringen
 - Kochrezept: Zutatenliste, nötige Handlungen (mischen, verrühren, erhitzen, abkühlen etc.)
 - Anleitung zum Reifenwechsel



Einführung in Algorithmen (2/8)

- Wesentliche Eigenschaft: vollständig und allgemein anwendbar
 - Wenn eine Aufgabe anliegt, für die der Algorithmus „zuständig“ ist, dann wird er „funktionieren“
 - Beschreibungen sind meist für eine ganze Klasse von Problemen gültig, nicht nur für ein Einzelproblem (nicht: „so addieren Sie 1001_b und 111_b “, sondern: „so addieren Sie Dualzahlen“)
 - Fallunterscheidungen erlauben große Unterschiedlichkeit in den Einzelproblemen („wenn ..., dann ...“)

Einführung in Algorithmen (4/8)

- Beispiel „1 Ei kochen“ kein gutes Beispiel: nicht besonders allgemeingültig
- besser: „So kochen Sie n Eier“ mit beliebigem n
 - Konsequenzen: Berücksichtigen, wie viele Eier im Eierkocher Platz haben; ggf. Kochvorgang mehrfach durchführen
- noch besser wäre: „So kochen Sie n Objekte vom Typ x “ - mit Fallunterscheidung für verschiedene x (nicht praktikabel)



Einführung in Algorithmen (3/8)

- Beispiel: „Kochrezept“ Ei kochen
 - **Sie benötigen: Eierkocher, Stromanschluss, Wasser, 1 Ei**
 - **Schritt 1: Leeren Sie den Wasservorrat im Eierkocher**
 - **Schritt 2: Füllen Sie den Eierkocher bis zur Markierung „1 Ei“ mit Wasser**
 - **Schritt 3: Geben Sie ein Ei in den Wasserkocher**
 - **Schritt 4: Stellen Sie den Eieranzahlregler auf „1“ und schalten Sie den Wasserkocher ein**
 - **Schritt 5: Warten Sie, bis ein Signal ertönt**
 - **Schritt 6: Schalten Sie den Wasserkocher aus**
 - **Schritt 7: Entnehmen Sie das Ei**

Einführung in Algorithmen (5/8)

- Eigenschaften eines „guten“ Algorithmus
 - jede Anweisung klar verständlich
 - Anweisungen sind eindeutig - sie lassen keine Interpretation zu
 - Anweisungen auch unabhängig davon, wer sie ausführen wird - Ergebnis immer gleich
 - Voraussetzungen, wann der Algorithmus anwendbar ist, sind klar genannt



Einführung in Algorithmen (6/8)

- Weg, um diese klare Struktur zu erreichen:
starke Formalisierung
 - z. B. durch grafische Darstellung mit optischen Elementen, die für folgende Konzepte stehen
 - Fallunterscheidung
 - wiederholte Ausführung einer Anweisung (so genannte Schleifen)
- Nassi-Shneiderman-Diagramme (Struktogramme)

Einführung in Algorithmen (8/8)

- Ein paar Programmiersprachen
 - Basic:

```
PRINT "Hallo Welt"
```
 - C:

```
main () {  
    printf ("Hallo Welt\n");  
}
```
 - Pascal:

```
BEGIN  
    writeln ("Hallo Welt");  
END
```
 - Windows-Shell:

```
echo "Hallo Welt"
```

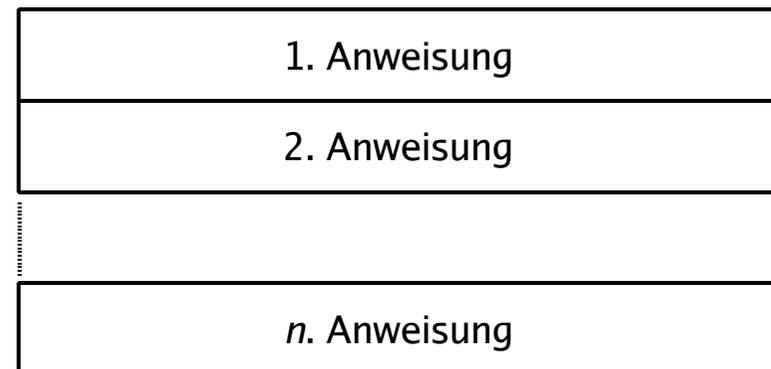


Einführung in Algorithmen (7/8)

- zusätzliche Regeln für Anweisungen an einen Computer:
 - Computer versteht verschiedene Programmiersprachen, aber keine natürlichen Sprachen
 - Syntax (also grammatikalischer Aufbau, Regelwerk) der Sprache muss man präzise einhalten, sonst versteht der Rechner nichts („syntax error“)
 - Neben rein syntaktischen gibt es auch logische Regeln (z. B. Reihenfolge von Befehlen)

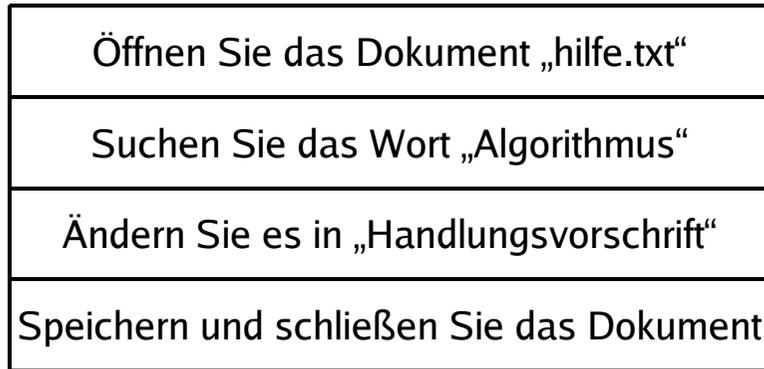
Nassi-Shneiderman (1/9)

- **Sequenz**
(Anweisungsfolge, Hintereinander-Ausführung)



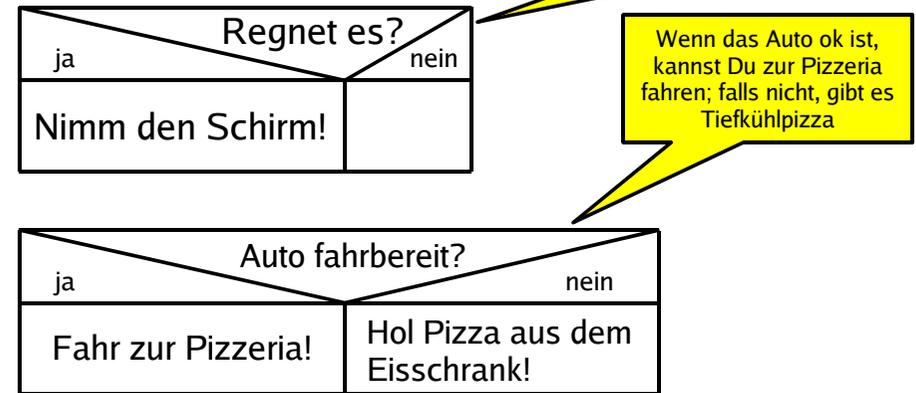
Nassi-Shneiderman (2/9)

- Sequenz: Beispiel



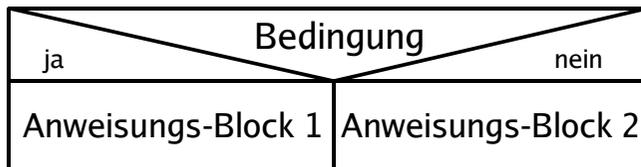
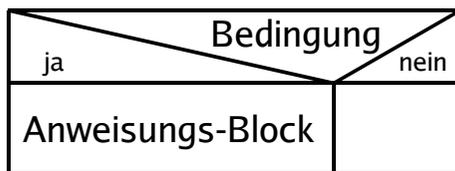
Nassi-Shneiderman (4/9)

- Verzweigungen: Beispiel



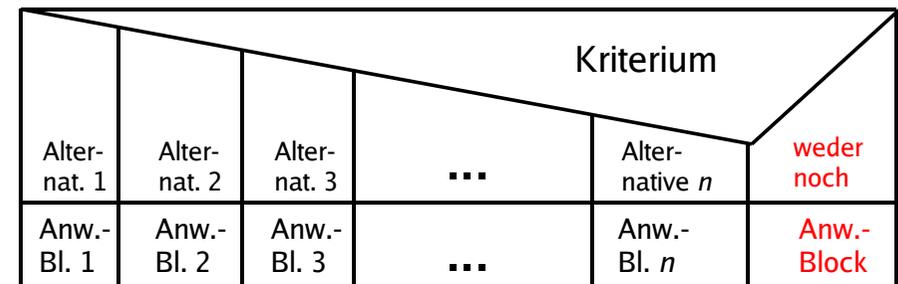
Nassi-Shneiderman (3/9)

- Verzweigung / Alternative
(Ausführung ist abhängig von Bedingung)



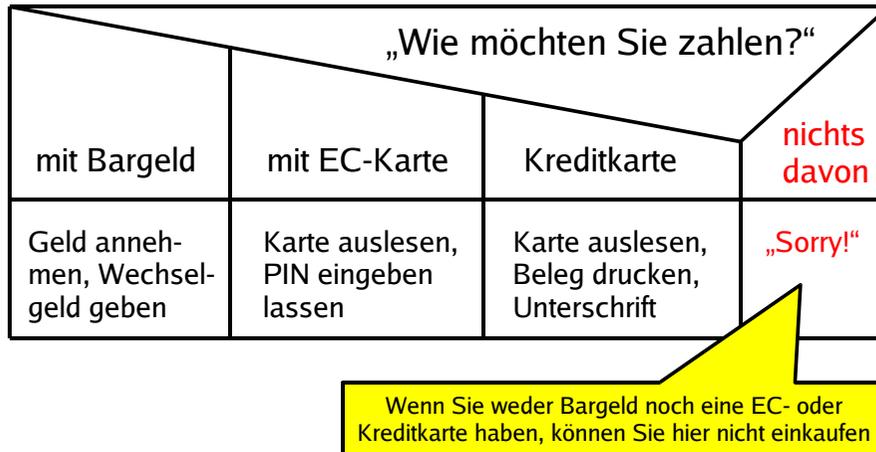
Nassi-Shneiderman (5/9)

- Mehrfachverzweigung
(mehr als zwei Alternativen)



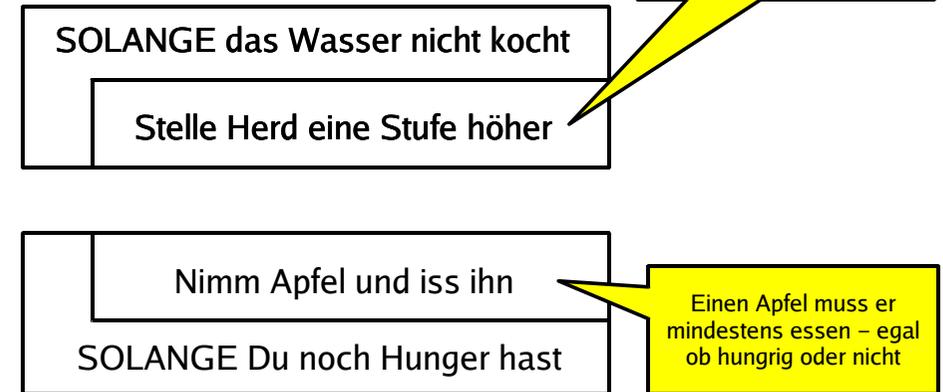
Nassi-Shneiderman (6/9)

- Mehrfachverzweigung: Beispiel



Nassi-Shneiderman (8/9)

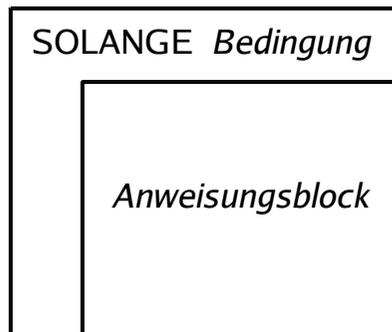
- Beispiel: Schleife



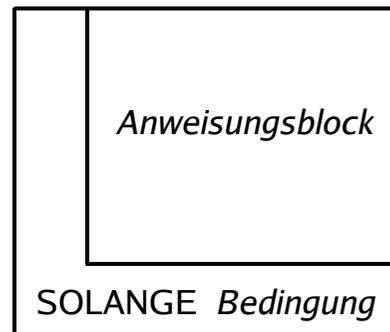
Nassi-Shneiderman (7/9)

- Schleife (Iteration, Wiederholung)

Variante 1: vorher prüfen



Variante 2: nachher prüfen



Nassi-Shneiderman (9/9)

- alle zusammen:

