Prozedurale und objektorientierte Programmierung

Algorithmen

Dozent: Tankred Grosch

Was ist ein Algorithmus?

Das Wort Algorithmus ist eine Abwandlung
des Namens **von Muhammed al-Chwarizmi** (s. Bild auf Briefmarke)
(etwa 783–850), dessen arabisches Lehrbuch
über das Rechnen mit indischen Ziffern (um 825) in der
mittelalterlichen lateinischen Übersetzung mit den Worten
Dixit Algorismi (Algorismi hat gesagt) beginnt.

- **Algorithmus** = genau definierte Handlungsvorschrift zur Lösung eines Problems oder einer bestimmten Art von Problemen in endlich vielen Schritten.

Der „ganze Alltag“ besteht aus Algorithmen:

- Gebrauchsanweisung, Bedienungsanleitung
- Rezepte zur Zubereitung von ...
- Anleitungen zum „Basteln“ eines/einer …
- Vorschriften nach denen man sich richtet (oder auch nicht)
- Prozessabläufe (Waschmaschine)
- Wegbeschreibung

Der Algorithmus legt das WIE der Problemlösung fest. Die zugehörige Aufgabenstellung ist die Problemspezifikation.

Algorithmen lösen i.A. nicht ein Problem sondern eine Problemklasse. Die konkrete Aufgabe wird durch Eingabeparameter bestimmt. Das zur Eingabe gelieferte Ergebnis heißt Ausgabe des Algorithmus.

Algorithmus = Verfahren

mit einer präzisen, endlichen Beschreibung
unter Verwendung effektiver Verarbeitungsschritte
effektiv = in endlicher Zeit ausführbar.

Eigenschaften von Algorithmen

- **endliche Länge**
- **Termination**
  - terminierender Algorithmus kommt in endlich vielen Schritten zum Ende
- **Determinismus**
  - deterministischer Algorithmus besitzt eindeutig vorgeschriebenen Verlauf
- **Determiniertheit**
  - determinierter Algorithmus führt zu eindeutigem Ergebnis
  - deterministischer Algorithmus ist immer determiniert
  - nicht-deterministischer Algorithmus ist bisweilen determiniert
- **Sequentialität / Parallelität**

Definition deterministischer Algorithmus

- Ein Algorithmus oder Programm wird durch Maschinen schrittweise abgearbeitet. Der Algorithmus heißt deterministisch, wenn es zu jeder Programmsituation eine nachfolgende Situation geben kann, wenn also zu jedem Zeitpunkt der Folgeschritt eindeutig bestimmt ist.

Determiniertheit

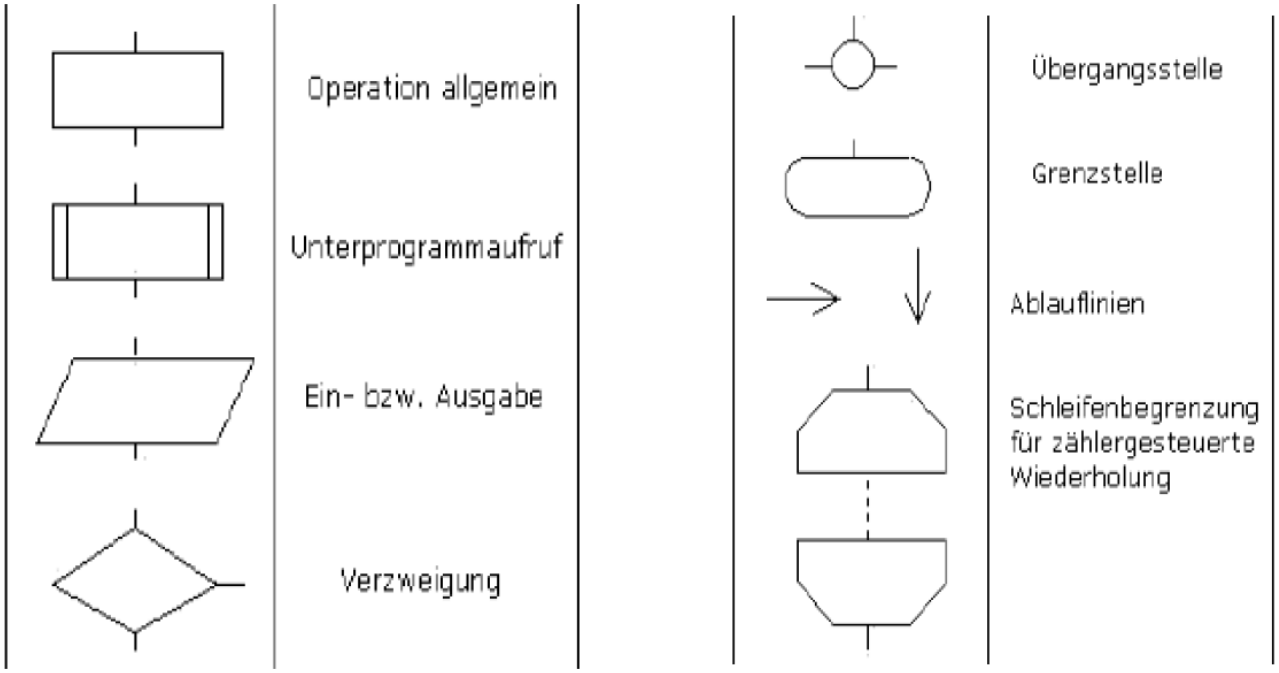
- bezieht sich auf das Ergebnis eines Algorithmus
- nicht jeder determinierte Algorithmus muss deterministisch sein, ein Algorithmus kann auch unterschiedliche Schritte abarbeiten und doch kommt er jeweils zum gleichen Ergebnis.

- **Anweisungen an den Bearbeiter (Mensch/Maschine)**

- **Struktur -> bestimmt den Ablauf**
  - + Sequentielle Ausführung der Anweisungen
  - + Bedingte Ausführung einer Anweisung
    (wenn … dann …)
  - + Alternative Ausführung zweier Anweisungen
    (wenn… dann … sonst …)
  - + Alternative Ausführung mehrerer Anweisungen
    (… falls … dann …, falls … dann …, falls … dann…)
  - + Wiederholte Ausführung von Anweisungen
    - **Feste Anzahl von Wiederholungen (führe … mal aus: …)**
    - **Unbestimmte Anzahl mit Abbruchbedingung (solange … führe … aus)**

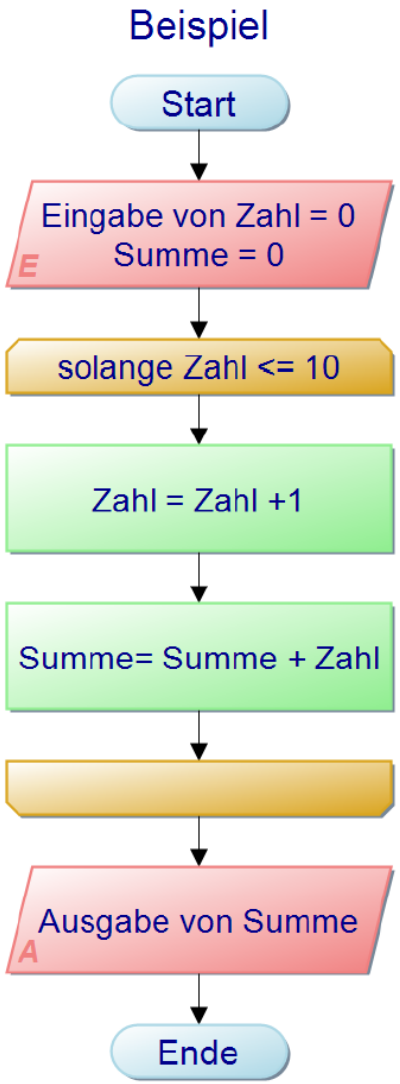
Programmablaufplan (PAP)

- **Ablaufdiagramm für einen Algorithmus**
  - auch Flussdiagramm oder Programmstrukturplan
  - graphische Darstellung zur Umsetzung eines Algorithmus
  - beschreibt die Folge von Operationen zur Lösung einer Aufgabe.



Programmablaufplan (PAP)

- **Beispiel**



Algorithmus von Euklid

- zur Bestimmung des größten gemeinsamen Teilers (ggT)

```
solange x ≠ y ist, wiederhole:
   wenn x > y, dann:
      ziehe y von x ab und weise das Ergebnis x zu.
   andernfalls:
      ziehe x von y ab und weise das Ergebnis y zu.
ENDE solange
x (bzw. y) ist der gesuchte ggT
```

Was enthält dieser Algorithmus alles?

- Variablen x und y
- Grundoperationen (vergleichen, abziehen, zuweisen)
- **sequentielle Folge** von Anweisungen, aber auch Auswahl (**Selektion**) und Wiederholung (**Iteration**)

Aussagen über Algorithmen

- **Voraussetzungen**
  - unter welchen Bedingungen arbeitet der Algorithmus?
    - Menge aller erlaubten Eingaben
    - Menge aller möglichen Ausgaben (bei erlaubten Eingaben)
  - Was geschieht bei falscher Eingabe?

- **Terminationsverhalten**
  - endet ein Algorithmus für **alle** möglichen Eingaben? Beweis!
  - Ist es möglich, den Algorithmus in einen nicht-endenden Zyklus zu bringen?

- **Korrektheit**
  - löst der Algorithmus das gestellte Problem?
  - liefert der Algorithmus bei Ausführung die gewünschte Ausgabe als Funktion der Eingabe?

Aussagen über Algorithmen (cont'd)

- **Aufwand/Effizienz**
  - Speicherplatzbedarf?
  - Ausführungszeit?
  - Abhängigkeit der Ausführungszeit von der Eingabe?