

# Grundformen von Algorithmen

---

## 1. Elementare, einfache Anweisung

z.B. Addition, Multiplikation ...

## 2. Sequenzielle Anweisungsfolge (linearer A.)

zeitliche Abfolge von Schritten

Beispiel: zuerst Socken, dann Hose, dann Schuhe anziehen usw.

## 3. Bedingte Ausführung

Auswahl eines Schrittes aufgrund einer zu testenden Bedingung

### Variante 1 (einseitige Auswahl)

Form: Falls Bedingung  $x$ , dann Schritt  $y$

Beispiel: Kaffee kochen (wenn Wasser heiß, dann...)

### Variante 2 (zweiseitige Auswahl)

Form: Falls Bedingung, dann Schritt  $a$ , sonst Schritte  $b$  oder  $c$

Beispiel: Verkehrsampel (wenn Ampel ausgefallen, dann fahre vorsichtig weiter -  
sonst: falls Ampel rot oder gelb, halte an - sonst: fahre weiter)

### Variante 3 (mehrseitige Auswahl)

Form: Wenn Fall 1, dann Reaktion 1; wenn Fall 2, dann Reaktion 2; wenn Fall 3, dann Reaktion 3

Beispiel: Min-/Max-Tempo = 70/120 km/h (wenn  $< 70$ : + Gas, wenn  $\geq 70$ : +- Gas, wenn  $> 120$ : - Gas)

## 4. Schleife

Wiederholung eines Schrittes aufgrund einer zu testenden Bedingung

### Variante 1

Form: Wiederhole Schritt solange, bis Abbruchkriterium erfüllt

Beispiel a: Primzahl (prüfe  $x \in \mathbb{N}$  solange, bis ermittelt ist, ob  $x$  durch 1 und  $x$  selbst teilbar)

Beispiel b: nächste Primzahl  $< 100$  (prüfe Teilbarkeit von  $x + 1 \in \mathbb{N}$  solange  $x < 100$ )

### Variante 2

Form: Solange Bedingung, führe aus alle nötigen Schritte

Beispiel: Bestimmung von  $\max x$  einer vorgegebenen Liste  $L \in \mathbb{N}$  (prüfe  $x$  solange, bis  $x > x-1$ )

Ausführung: Setze erste Zahl als bislang größte Zahl, solange Liste nicht abgearbeitet ist; lies nächste Zahl ein; falls diese Zahl  $>$  bislang größte Zahl, dann setze diese Zahl als bislang größte Zahl; gib bislang größte Zahl aus [usw.]

### Variante 3

Form: Wiederhole für nominellen Bereich (*Zählschleife: für  $i = 1$  bis  $n$* )

Beispiel: Primzahlbestimmung  $< 100$  (prüfe  $x \in \mathbb{N} < 100$  solange, bis ermittelt ist, ob  $x$  durch  $x$  teilbar)

Ausführung: Wiederhole für 1 bis 99: Test auf Primzahleigenschaft; falls  $x$  Primzahl, gib  $x$  aus.

## 5. Weitere Bausteine

- Parallele Ausführung (parallele Operationen, z.B. Addition zweier Vektorarten in einem Schritt)
- Unterprogramm (z.B. Klasse.Methode:  $x = \text{IO.readInt}$ )
- Rekursion („Türme von Hanoi“: 8 verschieden große Scheiben von Stapel A nach C verlagern)