

# Einführung in die Informatik I

## Kapitel I.6: Schleifen

Prof. Dr. Marcin Grzegorzek<sup>1</sup>

Research Group for Pattern Recognition  
[www.pr.informatik.uni-siegen.de](http://www.pr.informatik.uni-siegen.de)

Institute for Vision and Graphics  
University of Siegen, Germany



---

<sup>1</sup>Die im Rahmen dieser Lehrveranstaltung verwendeten Lernmaterialien wurden uns zum Großteil von Herrn Prof. Dr. Wolfgang Wiechert und Herrn Prof. Dr. Roland Reichardt zur Verfügung gestellt.

# Inhaltsverzeichnis

## **I. MATLAB-Einführung**

1. Voraussetzungen und Konventionen
2. Variablen und arithmetische Ausdrücke
3. Automatisierungen von Berechnungen
4. Logische Ausdrücke
5. Verzweigungen
- ▶ 6. Schleifen
7. Fehlersuche in Programmen
8. Funktionen
9. Arbeitsweise von Funktionen
10. Vektoren
11. Matrizen

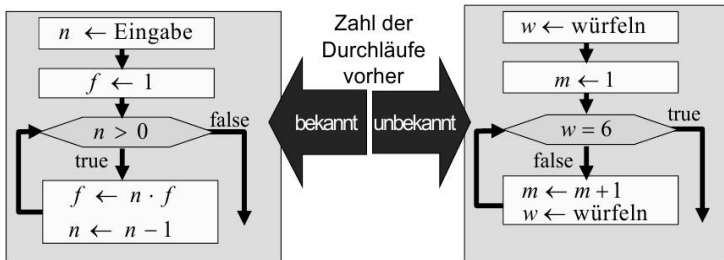
## **II. Algorithmen**

1. Suchen
2. Spezielle Suchalgorithmen
3. Sortieren
4. Rekursion und Quicksort

# Schleifen

- Programmschleifen ermöglichen die wiederholte Ausführung einer Anweisungssequenz
- Die Zahl der Durchläufe wird durch eine Bedingung gesteuert
- Bei jedem Durchlauf können sich einige Variablenwerte ändern
- Beispiel 1: Fakultät von  $n$  berechnen
- Beispiel 2: Zahl  $m$  der Würfe bis eine 6 fällt.

$$n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$$

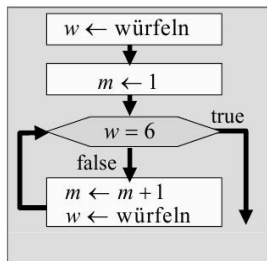


# while-Schleifen

- Eine while-Schleife hat in MATLAB die Form
  - **while** *Schleifenbedingung*  
*Schleifensequenz*  
**end**
- Bei einer while-Schleife muss nicht von Anfang an feststehen, wie oft eine Schleife durchlaufen wird
- Vor **jedem** Durchlauf wird eine logische Schleifenbedingung abgeprüft.
- Ist die Schleifenbedingung wahr, so wird die Schleife ein weiteres mal durchlaufen, anderenfalls wird sie abgebrochen.
- Im Schleifenkörper muss der Wert der Schleifenbedingung beeinflusst werden. Andernfalls wird die Schleife nicht beendet (Endlosschleife, Programm „hängt“).

## Beispiel: Würfeln bis eine 6 fällt

- Einen Würfel implementiert man in MATLAB wie folgt:  
`w=ceil(6*rand);`
- Beachte: Im Flussdiagramm kann die Schleife auch im false-Fall durchlaufen werden



- Pseudocode:

```
w ← ceil(6 · rand);  
m ← 1;  
while w < 6  
    m ← m + 1;  
    w ← ceil(6 · rand);  
endwhile;
```

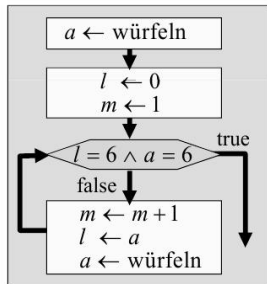
MATLAB-Code:

```
w=ceil(6*rand);  
m=1;  
while w<6  
    m=m+1;  
    w=ceil(6*rand);  
end
```

# Beispiel: Würfeln bis 2 mal hintereinander 6 fällt

- Das Programm soll so modifiziert werden, dass gezählt wird bis zwei Sechsen hintereinander fallen
- Dazu muss man sich die jeweils letzte Zahl  $l$  merken:

```
 $a \leftarrow \text{ceil}(6 \cdot \text{rand}) ;$  -- aktuelle Zahl  
 $l \leftarrow 0;$  -- letzte Zahl  
 $m \leftarrow 1 ;$  -- Zähler  
while  $l < 6 \vee a < 6$   
     $m \leftarrow m + 1 ;$   
     $l \leftarrow a ;$   
     $a \leftarrow \text{ceil}(6 \cdot \text{rand}) ;$   
endwhile ;
```



## for-Schleife

- Eine for-Schleife (Zählschleife) hat in MATLAB die Form:
  - ***for Schleifenvariable = Start:Schritt:Ende  
Schleifensequenz  
end***
- Bei for-Schleifen wird die *Schleifensequenz* für eine vorgegebene Wertesequenz der *Schleifenvariablen* durchlaufen.
- Zu Beginn wird die *Schleifenvariable* auf den *Startwert* gesetzt
- Bei jedem Schleifendurchlauf wird die *Schleifenvariable* um die *Schrittweite* erhöht bzw. erniedrigt.
- Die Schleife wird beendet, wenn der *Endwert* überschritten ist.
- Wie oft und für welche Werte die Schleife durchlaufen wird, steht somit schon vor Schleifenbeginn fest.
- Flussdiagramme kennen keine spezielle Notation für for-Schleifen.



## Beispiele für Zähl-Schleifen

- Wertetabelle der Sinus-Funktion für  $x=0.0, 0.1, 0.2, \dots, \text{Top}$ 

```
Top=10.0;  
for x=0.0:0.1:Top  
    disp(x);  
    disp(sin(x));  
end
```

 Variablenwerte zulässig
- Summe der Zahlen von 1 bis 100

```
Summe=0;  
for i=1:100  
    Summe=Summe+i;  
end
```

 Schrittweite 1 kann weggelassen werden
- Summe der Zahlen von 1 bis 100 mit Schrittweite 2

```
Summe=0;  
for i=1:2:100  
    Summe=Summe+i;  
end
```

 letzte Zahl ist 99
- Fakultät von 10

```
Fak=1;  
for i=10:-1:1  
    Fak=Fak*i;  
end
```

 herunterzählen

# Unzulässige Zähl-Schleifen

- *Schrittweite* und *Endwert* wird bei der `for`-Schleife **vor** dem ersten Schleifendurchlauf bestimmt. Sie können während des Schleifendurchlaufs nicht mehr verändert werden:
  - Obergrenze unveränderbar:

```
n=1;
for Zaehler=1:n
    if ceil(6*rand)~= 6
        n=n+1; % wirkungslos
    end;
end;
```
  - Schrittweite unveränderbar:

```
Step=1;
for Zaehler=1:Step:10
    disp (Zaehler);
    Step=Step+1; % wirkungslos
end;
```
  - Variable unveränderbar:

```
for Z=1:10
    disp (Z);
    Z=Z+10; % wirkt nur im Block
    disp(Z);
end;
```
-

## Beispiel: Zinsrechnung

- Bei der Abzahlung eines Kredits wird monatlich ein fester Betrag  $MB$  gezahlt
- Dieser Betrag wird zunächst für die Bezahlung der monatlichen Schuldzinsen verwendet
- Die Schuldzinsen sind ein fester Prozentsatz  $PS$  vom noch verbleibenden Kredit  $K$
- Der übrig bleibende Betrag  $T$  wird dann zur Tilgung des Kredits eingesetzt
- Monatlich ist also folgendes zu berechnen:
$$T \leftarrow MB - PS \cdot K;$$
$$K \leftarrow K - T;$$
- Frage:
  - Wie lange läuft der Kredit ?
  - Wie viel kostet der Kredit insgesamt ?
  - Wie hoch ist die letzte Rate ?

# Pseudocode

## Eingabe:

$K$  -- Kredit

$PS$  -- Prozentsatz,

$MB$  -- Monatlicher Betrag

## Ausgabe:

$LZ$  -- Laufzeit

$GK$  -- Gesamtkosten

$LR$  -- letzte Rate

$LZ \leftarrow 0$  ;

$GK \leftarrow 0$  ;

**while**  $K > 0$

$LZ \leftarrow LZ + 1$  ;

$T \leftarrow MB - PS \cdot K$  ;      -- Tilgung

**if**  $T \geq K$       -- Kreditende erreicht

$T \leftarrow K$  ;

$LR \leftarrow PS \cdot K + T$

$GK \leftarrow GK + LR$  ;

$K \leftarrow 0$  ;

**else** -- Kreditende noch nicht erreicht

$K \leftarrow K - T$  ;

$GK \leftarrow GK + MB$  ;

**endif** ;

**endwhile** ;

# Matlab Code

```
Kredit          = input('Kredit ');
Prozente        = input('Prozentsatz pro Monat ');
Monatl_Betrag   = input('Monatlicher Betrag ');

Laufzeit        = 0;
Gesamtkosten   = 0;

while Kredit > 0
    Laufzeit = Laufzeit + 1;
    Tilgung  = Monatl_Betrag - Prozente * Kredit % Tilgung
    if Tilgung >= Kredit % Kreditende erreicht
        Tilgung      = Kredit;
        Letzte_Rate  = Prozente * Kredit + Tilgung;
        Gesamtkosten = Gesamtkosten + Letzte_Rate;
        Kredit = 0;
    else % Kreditende noch nicht erreicht
        Kredit      = Kredit - Tilgung
        Gesamtkosten = Gesamtkosten + Monatl_Betrag
    end
end
% Ausgabe
Laufzeit
```