

Einführung in die Informatik I

Kapitel I.6: Schleifen

Prof. Dr.-Ing. Marcin Grzegorzek
Juniorprofessur für Mustererkennung im Institut für Bildinformatik
Department Elektrotechnik und Informatik
Fakultät IV der Universität Siegen

21.11.2012

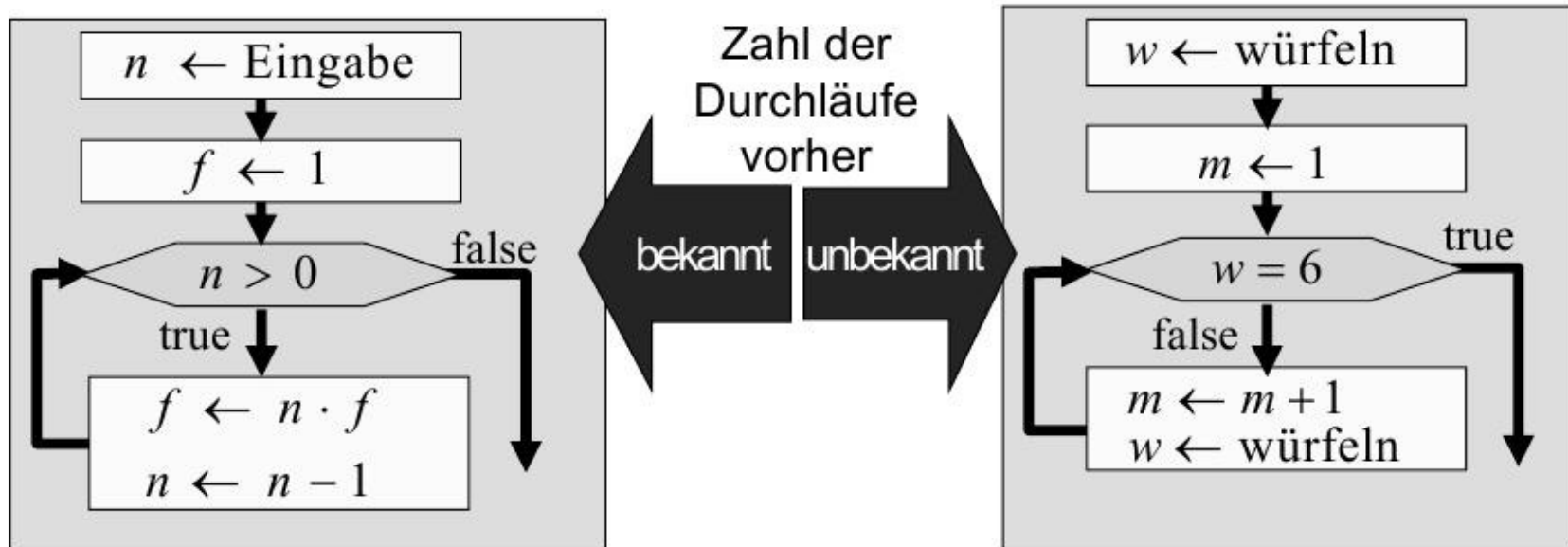
Inhaltsverzeichnis

- I. MATLAB-Einführung
 - 1. Voraussetzungen und Konventionen
 - 2. Variablen und arithmetische Ausdrücke
 - 3. Automatisierung 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
- III. MATLAB-Fortsetzung
- IV. Wissenschaftliche Werkzeuge

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.
- Manchmal die Anzahl der Durchläufe ist auch unbekannt. (s. Bsp. unten.)
- Beispiele: n -Fakultät und würfeln 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 durchgelaufen, andernfalls wird sie abgebrochen.

While-Schleifen

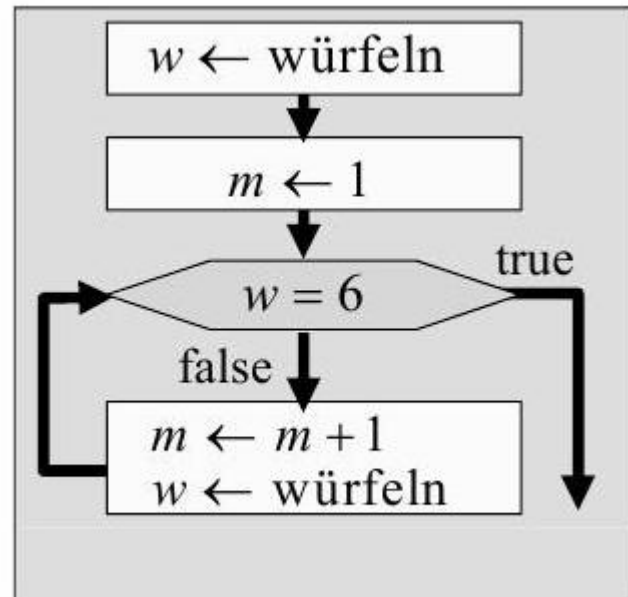
- Im Schleifenkörper muss der Wert der Schleifenbedingung beeinflusst werden. Anderfalls wird die Schleife nicht beendet und in diesem Fall handelt sich um eine unendliche Schleife und wird nie beendet, Programm hängt.

Beispiel: Würfeln bis eine 6 fällt

- Einen Würfel implementiert man in MATLAB wie folgt mit der Formel:
Zahl=ceil(6*rand)
- Beachte: Im Flussdiagramm kann die Schleife auch im false-Fall durchlaufen werden.

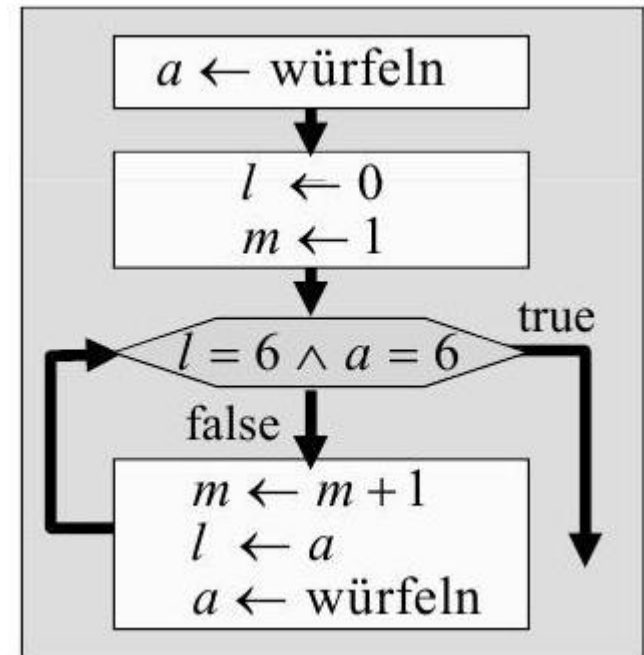
- MATLAB Code:

```
Zahl=ceil(6*rand);  
Anzahl_der_Durchläufe=1;  
While Zahl < 6  
    m=m+1;  
    Zahl=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 1 merken.



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 [A:x:B] bzw. erniedrigt [B:-x:A].
- 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 und ist bekannt.

Beispiele für Zähl-Schleifen

- Wertetabelle der Sinus-Funktion für $x = \{0, 0.1, 0.2, \dots, n\}$:

```
n=10;  
for x=0:0.1:n  
    disp(x)  
    disp(sin(x))  
end
```

- Summe der Zahlen von 1 bis 100:

```
Summe=0;  
for 1:100 %Schrittweite 1 kann weg gelassen werden  
    Summe = Summe + i;  
end
```

Beispiele für Zähl-Schleifen

- Summe der Zahlen von 1 bis 100 mit Schrittweite 2:

```
Summe=0;  
for i=1:2:100 % Letzte Zahl ist 99  
    Summe=Summe + i;  
end
```

- Fakultät von 10:

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

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.

Beispiele für Unzulässige Zähl-Schleifen

- Obergrenze unveränderbar:

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

Beispiele für Unzulässige Zähl-Schleifen

- Schrittweite unveränderbar:

```
step=1;  
for Zaehler =1:step:10  
    disp(Zaehler);  
    Step=step + 11;  % wirkungslos  
end
```

- Variable unveränderbar:

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

Informationen

Quelle :

- [1] Vorlesungsskript Einführung in die Informatik I, Prof. Reichhardt, Universität Siegen, 2009.
- [2] A Guide to MATLAB for Beginners and Experienced Users, Brian R. Hunt, Ronald L. Lipsman, Jonathan M. Rosenberg, Cambridge University Press, 2001.
- [3] An Introduction to Programming and Numerical Methods in MATLAB, S.R. Otto and J.P. Denier, Springer, 2005.

Personen:

- Prof. Dr. Marcin Grzegorzek
 - Address: Research Group for Pattern Recognition Department ETI, University of Siegen Hoelderlinstr. 3, H-F 016, D-57076 Siegen.
- Dr. -Ing. Andreas Hoffmann
 - Address: Research Group for Pattern Recognition Department ETI, University of Siegen, Hoelderlinstr. 3, H-B 8405, D-57076 Siegen.