Kapitel 7

Schleifen

Schleifen

- Schleifen werden zur wiederholten Ausführung von Anweisungen verwendet.
- Es werden zwei Arten von Schleifen unterschieden:
  - *for* -Schleife
  - *while* -Schleife

for -Schleife

- Ist die Anzahl der Schleifendurchläufe im Voraus bekannt, wird die *for* -Schleife verwendet.
- Die Elemente zur Kontrolle der Schleife werden im Schleifenkopf zusammengefasst.
- Da die Elemente zur Kontrolle der Schleife am Anfang stehen, nennt man diese Schleife: **kopfgesteuerte Schleife**.

```
for variable = Ausdruck
        Anweisungen
end
```

for -Schleife

- Für den Ausdruck gibt es 3 unterschiedliche Möglichkeiten.
- Der Ausdruck kann sein:
    1. Initialisierung:Endwert
    2. Initialisierung:Schrittweite:Endwert
    3. Vektor

Im Prinzip ist also jeder Ausdruck, den ich bei der for-Schleife verwende ein Vektor:

1:10 ist der Vektor [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
1:2:10 ist der Vektor [1, 3, 5, 7, 9]
[2, 4, 6, 8, 10] ist der Vektor [2, 4, 6, 8, 10]

for -Schleife

1. Initialisierung:Endwert

```
for i = 1:5
    disp(i);
end
```

```
1
2
3
4
5
```

for -Schleife

- Bei dieser for-Schleife ist i die Zählvariable.
- Die Schleife wird beginnend mit dem Wert der Initialisierung bis zum Endwert durchlaufen.
- Bei jedem Schleifendurchlauf wird i um 1 erhöht.

- Der Ausdruck (1:5) erstellt also einen Vektor.
  Er lautet in diesem Beispiel [1, 2, 3, 4, 5] und wird in der for-Schleife Element für Element durchlaufen.

for -Schleife

2. Initialisierung:Schrittweite:Endwert

```
for i = 1:2:10
    disp(i);
end
```

```
1
3
5
7
9
```

for -Schleife

- Bei dieser for-Schleife ist i die Zählvariable.
- Die Schleife wird beginnend mit dem Wert der Initialisierung bis zum Endwert durchlaufen.
- Die Schrittweite ist angegeben mit 2, daher wird bei jedem Schleifendurchlauf i um 2 erhöht.

- Der Ausdruck (1:2:10) erstellt also einen Vektor.
  Er lautet in diesem Beispiel [1, 3, 5, 7, 9] und wird in der for-Schleife Element für Element durchlaufen.

for -Schleife

3. Vektor

```
for i = [2, 4, 6, 8, 10]
    disp(i);
end
```

```
2
4
6
8
10
```

for -Schleife

- Bei dieser for-Schleife ist i die Zählvariable.
- Die Schleife wird beginnend mit dem ersten Wert des Vektors bis zum letzten Wert des Vektors durchlaufen.

- Der Ausdruck ([2, 4, 6, 8, 10]) ist ein Vektor und wird in der for-Schleife Element für Element durchlaufen.

for -Schleife

- Die Werte des Vektors müssen **nicht** in aufsteigender Reihenfolge stehen.

```
for i = [2, 4, -1]
    disp(i * 2);
end
```

```
4
8
-2
```

for -Schleife

- Wenn gewünscht, kann man auch rückwärts zählen lassen.

```
for i = -1:-5:-15
    disp(i * 2);
end
```

```
-2
-12
-22
```

for -Schleife

- Ist der Endwert **kleiner** als die Initialisierung, die Schrittweite aber positiv, so wird die Schleife **nicht** durchlaufen.

```
for i = 1:-10
    disp(i * 2);
end
```

Keine Ausgabe !

while -Schleife

- Ist die Anzahl der Schleifendurchläufe nicht bekannt, wird die *while* -Schleife verwendet.
- Die Schleife wird nur ausgeführt, wenn der Ausdruck wahr ist.
- Die Überprüfung, ob der Ausdruck wahr ist, geschieht bei jedem Schleifendurchlauf.
- Da der zu prüfende Ausdruck am Anfang steht nennt man diese Schleife: **kopfgesteuerte Schleife**.

```
while Ausdruck
        Anweisungen
end
```

while -Schleife

- Der Ausdruck muss in Matlab **nicht** in Klammern stehen.
- Aus Erfahrung zwecks besserer Übersicht, werden wir ihn **trotzdem bitte** in Klammern schreiben:

```
while (Ausdruck)
        Anweisungen
end
```

while -Schleife

- Der Ausdruck kann ganzzahlig oder vom Typ *logical* sein.
- Ist er ganzzahlig, entspricht 0 *false* und ungleich 0 *true*.

```
i = 0;
while (i<5)
    i = i+1;
    fprintf("%i ", i);
end
```

```
1 2 3 4 5
```

while -Schleife

- Ist der Ausdruck schon vor dem Eintritt in die Schleife *false*, wird der Schleifenkörper nicht durchlaufen.

```
i = 10;
while (i<5)
    i = i+1;
    fprintf("%i ", i);
end
```

Keine Ausgabe !

while -Schleife

- Ändert sich der Ausdruck nicht, kann man versehentlich eine Endlosschleife produzieren:

```
i = 0;
while (i<5)
                                                    % i = i+1;
    fprintf("%i ", i);
end
```

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0...
```

while -Schleife

- Endlosschleifen sind meist nicht gewollt.
- Benötigt man aber doch eine Endlosschleife, so wird diese meist einfach durch while(1) herbeigeführt.

```
i = 0;
while (1)
      i = i+1;
      fprintf("%i ", i);
end
```

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19...
```

while -Schleife

- Es können auch mehrere Ausdrücke geprüft werden.

```
i = 1; j = 1;
while ((i > 0) && (j < 5))
    i = i+1;
    j = j+1;
    fprintf("%i ", i);
end
```

```
2 3 4 5
```

break -Anweisung

- Die *break* –Anweisung wird verwendet, um eine Schleife unmittelbar zu verlassen.
- Bei verschachtelten Schleifen wird nur die aktuelle Schleife verlassen.

break -Anweisung - Beispiel

```
for i = 1:10
    if (i == 5)
        break;
    end
    fprintf("%i  ", i);
end
```

```
1  2  3  4
```

continue -Anweisung

- Die *continue* –Anweisung wird verwendet, um in einer Schleife unmittelbar die nächste Wiederholung zu beginnen.
- Bei einer *for* -Schleife wird direkt zum nächsten Element gesprungen.
- Bei einer *while* -Schleife wird zur Schleifenbedingung gesprungen.

continue -Anweisung - Beispiel

```
for i = 1:10
    if (i == 5)
        continue;
    end
    fprintf("%i  ", i);
end
```

```
1  2  3  4  6  7  8  9  10
```

return -Anweisung

- Durch den Befehl *return* kann ein Programm oder eine Funktion sofort beendet werden.

```
for i = 1:10
    if (i == 5)
        break;
    end
    fprintf("%i  ", i);
end
disp("Ende");
```

```
1  2  3  4  Ende
```

```
for i = 1:10
    if (i == 5)
        return;
    end
    fprintf("%i  ", i);
end
disp("Ende");
```

```
1  2  3  4
```

Kopfgesteuerte Schleifen - Beispiel

- In folgendem Beispiel werden alle Primzahlen zwischen 1 und 1000 ermittelt und ausgegeben.
- Die Ermittlung erfolgt nach einem vorgegebenen Algorithmus unter Zuhilfenahme der Modulo-Berechnung.
- Siehe Beispiel 4 (auf der Webseite zu finden).

do/while -Schleife

- In den meisten Programmiersprachen gibt es zusätzlich eine Schleife, die fußgesteuert ist.
- Der zu prüfende Ausdruck steht in diesem Fall am Ende der Schleife. Daher ist es eine: **fußgesteuerte Schleife**.
- Das Besondere ist, dass diese Schleife **mindestens einmal** durchlaufen wird.
- In Matlab existiert für diese Art der Schleife leider kein eigener Befehl.

do/while -Schleife

- Man muss sie sich aus anderen Befehlen konstruieren, um den gleichen Effekt zu erhalten.

```
while (true)
        Anweisungen
        if (Ausdruck)
            break;
        end
end
```

do/while -Schleife

- Beispiel:

```
while (true)
        i = input("Eine positive Zahl eingeben: ");
        if (i > 0)
            break;
        end
end
```

do/while -Schleife

- Man könnte so auch prüfen, ob überhaupt ein gültiger Wert eingegeben wurde. Diese Schleife wird nur verlassen, wenn eine positive Zahl eingegeben wurde.

```
while (true)
        i = input("Eine positive Zahl eingeben: ");
        if (i > 0)
            break;
        end
end
```

do/while -Schleife

- Möchte man nur Zahlen als Eingabe haben, kann man leere Eingaben und Eingaben, die keine Zahlen sind, abfangen.

- Der Befehl *isempty(i)*
  prüft, ob die Variable i leer ist.

- Der Befehl *isnan(i)*
  prüft, ob die Variable i ein Zeichen und keine Zahl ist. (isnan => is not-a-number)

do/while -Schleife

- Das würde dann so aussehen:

```
while (true)
        i = input("Bitte eine Zahl eingeben: ", "s");
        if ( ~isempty(i) && ~isnan(str2double(i)) )
            i = str2double(i);
            break;
        end
end
```

Fußgesteuerte Schleife - Beispiel

- In folgendem Beispiel wird ein Auswahlmenü programmiert.
- Der Benutzer kann die Zahlen 1, 2 oder 3 eingeben.
- Bei 1 oder 2 wird das entsprechende Programm ausgeführt.
- Bei 3 wird das Auswahlmenü beendet.
- Siehe Beispiel 5 (auf der Webseite zu finden).

Verschachtelte Schleifen

- Schleifen dürfen ineinander verschachtelt werden.
- Beispiel 1:

```
for i = 1:2
    for j = 1:2
        fprintf("%i  %i\n", i, j);
    end
end
```

Ausgabe:
1 1
1 2
2 1
2 2

Verschachtelte Schleifen

- Beispiel 2:

  Entscheidend ist, wo die Variable j auf 1 gesetzt wird !

```
for i = 1:2
    j = 1;
    while (j <= 2)
        fprintf("%i  %i\n", i, j);
        j = j + 1;
    end
end
```

Ausgabe:

```
1  1
1  2
2  1
2  2
```

Verschachtelte Schleifen

- Beispiel 3:

```
j = 1;
for i = 1:2
    while (j <= 2)
        fprintf("%i  %i\n", i, j);
        j = j + 1;
    end
end
```

Ausgabe:
1 1
1 2

Verschachtelte Schleifen - Beispiel

- In folgendem Beispiel wird eine Tabelle erstellt mit den Werten für Radius, Höhe und Volumen.
- Dabei wird der Radius und die Höhe in 0.5er Schritten hochgezählt und dann das entsprechende Volumen ausgerechnet.
- Siehe Beispiel 6 (auf der Webseite zu finden).