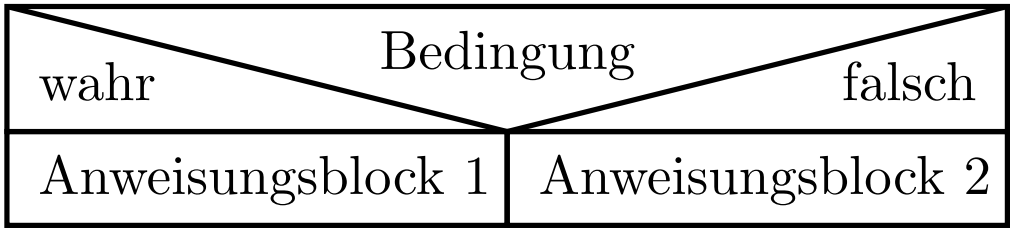
Informatik für Schüler, Foliensatz 4 Fallunterscheidungen und logische Ausdrücke

Prof. G. Kemnitz

Institut für Informatik, Technische Universität Clausthal

27. Oktober 2009

Binäre Fallunterscheidung

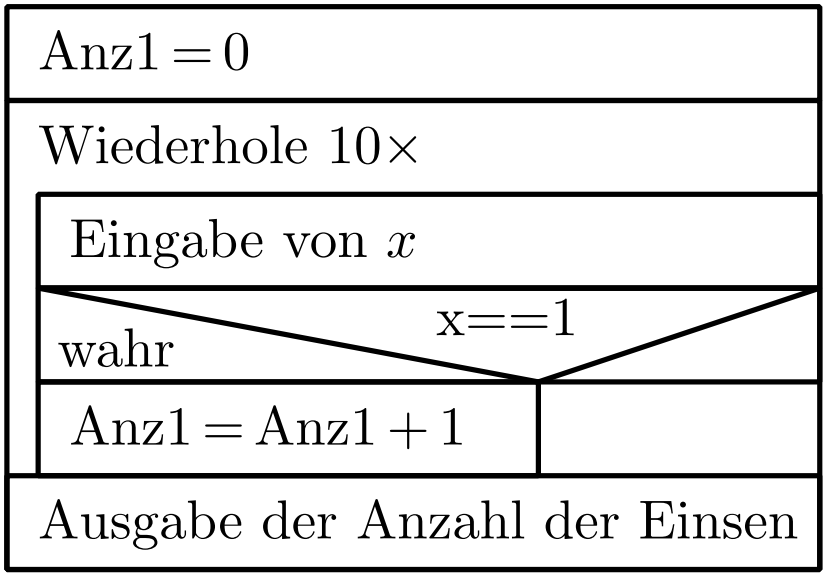


- in Python

```
if Bedingung:
    Anweisungsfolge_Block_1
[else:
    Anweisungsfolge_Block_1]
```

- Bedingung ist ein Ausdruck vom Typ bool, z.B. ein Vergleich
- Die Anweisungsblöcke sind Teilprogramme aus einer oder mehreren Anweisungen, der auch Fallunterscheidungen und Schleifen enthalten dürfen

Eingegebene »Einsen« zählen



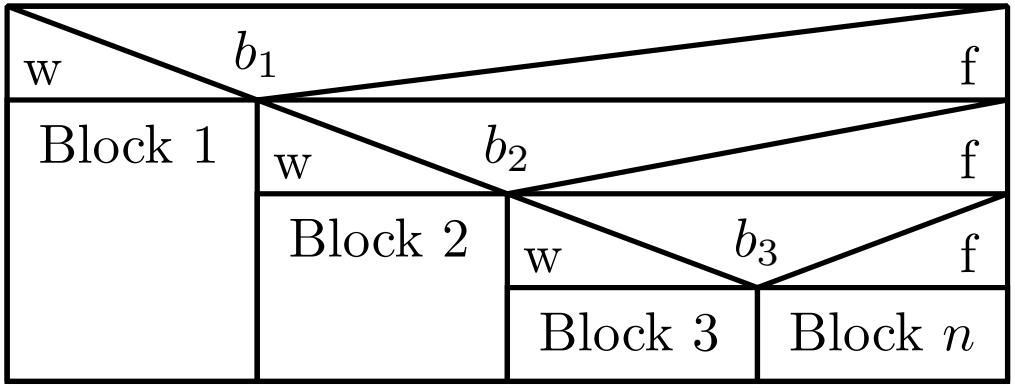

```
Anz1 = 0
for idx in range(1, 10):
  x = input('x =')
  if x==1:
    Anz1 = Anz1 + 1
print 'Die Anzahl der Einsen war:', Anz1
```

Anweisungsblöcke und Einrücktiefe

- Die Zusammenfassung von Anweisungsfolgen zu Blöcken wird in Python durch die Einrücktiefe gekennzeichnet.
- Schleifenkörper und die Blöcke der einzelnen Abarbeitungsalternativen haben eine höhere Einrückungstiefe.
- Wenn der übergeordnete Block weitergeht, ist die Einrücktiefe wieder auf den Wert dieses Blocks zurückzustellen.

```
Anz1 = 0                        # Block 1
for idx in range(1, 10):        # Block 1
  x = input('x =')              #  Block 2
  if x==1:                      #  Block 2
    Anz1 = Anz1 + 1             #    Block 3
print '...', Anz1               # Block 1
```

Unterscheidung mehrerer Fälle

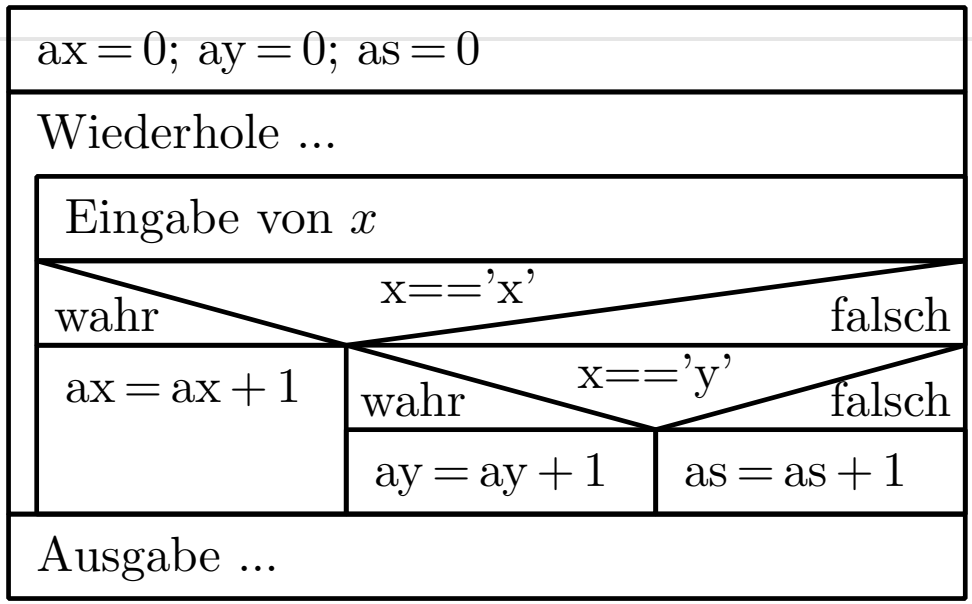


b_i	Bedingung i
w	wahr (true)
f	falsch (false)

- in Python

```
if Bedingung_1:
    Anweisungsfolge_Block_1
{elif Bedingung_i:
    Anweisungsfolge_Block_i}
[else:
    Anweisungsfolge_Block_n]
```

Es soll gezählt werden, wie oft der Buchstabe »x«, wie oft der Buchstabe »y« und wie oft etwas andes eingegeben wird.




```
ax = 0; ay = 0;
as = 0;
for idx in range(1, 10):
  x = input('x =');
  if x=='x':
    ax = ax + 1
  elif x=='y':
    ay = ay +1
  else:
    as = as + 1
print 'ax =', ax, 'ay =', ay, 'as =', as
```

Bildung logischer Ausdrück

- Logische Bedingungen werden mit Vergleichs- und Ordnungsrelationen gebildet:

a<b	a<=b	a>b	a>=b	==	!=
kleiner	kleiner gleich	größer	größer gleich	gleich	ungleich

- Die Ordnungsrelationen sind in Python auch auch auf Zeichenketten und andere Operanden anwendbar.
- Stellen Sie Vermutungen über die Werte folgender Ausdrücke auf und überprüfen Sie diese im Anschluss:

```
75 > 0.12
'Haus' < 'Maus'
15 == 'Fünfzehn'
```

Verknüpfung logischer Ausdrücke

- Logische Ausdrücke können negiert und mit den logischen Operatoren UND und ODER verknüpft werden:

a	b	a and b	a or b	not b
False	False	False	False	True
False	True	False	True	False
True	False	False	True	
True	True	True	True	

- Welche Werte haben die folgenden Variablen und Ausdrücke?

```
(3 < 5) or ('x' > 'y')
a = (5==7); b = (8 >= 3)
(a and b) or (not (a and b)
```

Kommentare

- Programme sind ohne zusätzliche Erklärungen, aussagekräftige Namen für die einzelnen Objekte und eine klare Struktur selbst für den Autor nach wenigen Wochen nicht mehr zu verstehen
- Kommentare beginnen mit dem Zeichen »#« und reichen bis Zeilenende
- Am Anfang eines Programms sollte immer ein kurzer Hilfetext mit den wichtigsten Benutzerhinweisen, Autor, Erstellungsdatum etc. stehen, z.B.:

```
# Programm, dass nacheinander 10 Zahlen einliest,
# die positiven Zahlen zählt, die negativen Zahlen
# zählt und abschließend beide Werte ausgibt.
# Autor:              Sven Maier
# letzte Änderung: 21.03.2006
```

Klausuraufgaben

- Aufgabe 1: Eingabeaufforderung; Erzeugung mehrerer Ausgabezeilen in einer Schleife
- Aufgabe 2: Erzeugung einer Wertetabelle für eine in einer Liste vorgegebener Eingaben und eine vorgegeben Berechnungsvorschrift z.B.:

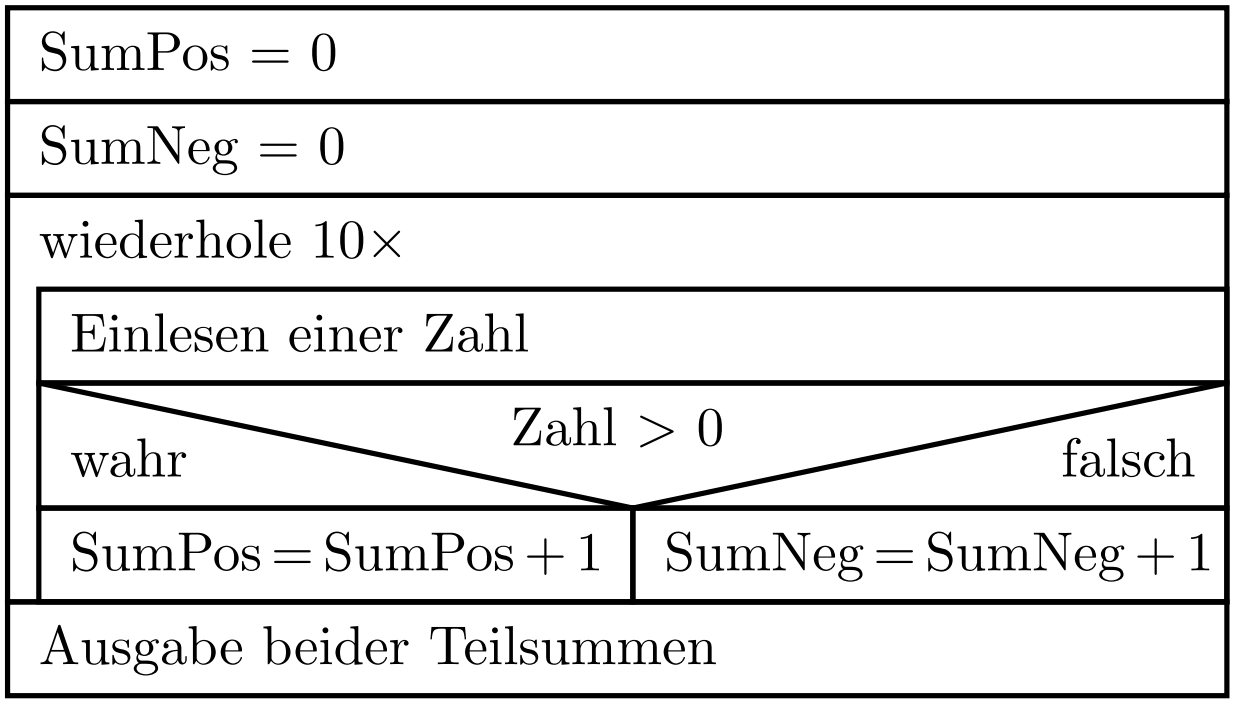
$$y = \begin{cases} 2 \cdot x & \text{für } x < -3 \\ x^2 & \text{für } -3 \leq x \leq 3 \\ -2 \cdot x & \text{sonst} \end{cases}$$

- Aufgabe 3: Suche und Beseitigung der Fehler in einem vorgegebenen Programm.

Die Lösungen sind in vorgegeben Verzeichnissen unter vorgegebenen Dateinamen zu speichern. Erlaubte Hilfsmittel: Folien auf der Webseite, eigene Lösungen und Mitschriften

Aufgabe 4.1

Schreiben Sie ein Programm »Sum10.py«, das nacheinander 10 Zahlen einliest, die positiven Zahlen zählt, die negativen Zahlen zählt und abschließend beide Anzahlen ausgibt. Algorithmus als Struktogramm:



Hilfestellungen

- Das Programm und alle weiteren zu entwickelnden Programme sollen im Unterverzeichnis »Uebung4« stehen
- Schleifenkörper und die Anweisungsblöcke der einzelenen Fälle sind einzurücken. Alle Anweisungen eines Blockes müssen dieselbe Einrücktiefe besitzen.
- Kommentare nicht vergessen!
- Eingabeanweisung:

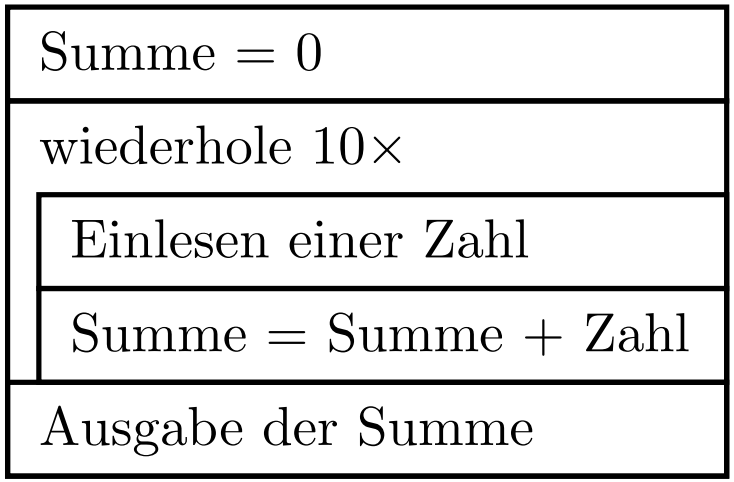
```
x = input('x =')
```

- Linux-Kommandos:

```
man [Befehl]
ls [-l][-a] [Pfad][Datei]
mkdir Verzeichnisname
cd Pfad
python Sum10.py
```

Aufgabe 4.2

Im nachfolgenden Struktogramm werden 10 Eingabewerte addiert:



- Erweitern Sie das Struktogramm so, dass innerhalb des Schleifenkörpers nur Eingabewerte vom Typ »int« oder »float« zur Summe addiert werden. Wenn der Eingabewert keine Zahl ist, soll zusätzlich der Text »Ungültige Eingabe!« und wenn die Eingabe eine Zahl ist, der Text »Zwischenergebnis:« gefolgt vom Wert ausgegeben werden.
- Programmieren und testen Sie den Algorithmus.

Aufgabe 4.3

- Entwickeln Sie ein Struktogramm und ein Programm, dass aus einer Folge von zehn Eingebewerten den größten und den kleinsten Wert bestimmt und am Ende ausgibt.