# Kontrollstrukturen mit Alternativen / Verzweigung

Bisher haben wir Methoden entwickelt, die ausschließlich aus einzelnen Anweisungen bestehen, die nacheinander ausgeführt werden. Das Programm läuft sozusagen gerade durch, ohne Fallunterscheidungen (Alternativen/Verzweigungen) zu berücksichtigen. Mit einer bedingten Anweisung lassen sich Alternativen berücksichtigen. Dies wird in Python mit einer if-Anweisung realisiert.

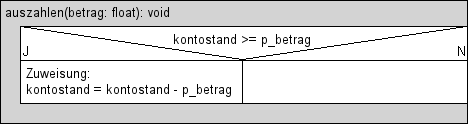
## Einseitige Verzweigung

Eine einseitige Verzweigung führt einen Programmcode nur dann aus, wenn eine Bedingung wahr (True) ist. Das folgende Beispiel zeigt eine einseitige Verzweigung.

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | class Konto:  def \_\_init\_\_(self, p\_kontonummer, p\_inhaber, p\_kontostand):  self.kontonummer = int(p\_kontonummer)  self.inhaber = str(p\_inhaber)  self.kontostand = float(p\_kontostand)  def auszahlen(self, p\_betrag):  if self.kontostand >= p\_betrag:  self.kontostand = self.kontostand - p\_betrag  #Hauptprogramm  k1 = Konto(3456789, "Simon Amsel", 0.0)  k1.einzahlen(100)  k1.auszahlen(50)  print(k1.kontostand)  k1.auszahlen(100)  print(k1.kontostand) | Ausgabe  50  50 |

Die einseitige Verzweigung beginnt mit dem Schlüsselwort if. Darauf folgt eine Bedingung (self.kontostand >= betrag). Trifft diese Bedingung zu, wird der Anweisungsblock ausgeführt. Dieser Block wird in Python gebildet, indem nach der Bedingung ein Doppelpunkt kommt. Alle zum if-Block gehörigen Anweisungen werden darunter eingerückt geschrieben (vgl. Zeile 5). Ist der Block beendet, wird die Einrückung der Anweisungen beendet/zurückgeführt. Die einseitige Verzweigung verhindert in diesem Beispiel, dass ein Kontostand durch eine Auszahlung negativ werden kann (vgl. Zeile 16, 17).

**Struktogramm:**



Wichtig: Beachten Sie die Liste der möglichen Vergleichsoperatoren auf der folgenden Seite.

**Merke:**

|  |  |
| --- | --- |
| Einseitige Verzweigung | |
| if Bedingung:  Anweisung |  |

**Vergleichsoperatoren**

Vergleichsoperatoren werden immer zum Vergleich zweier Werte verwendet und finden ihre Anwendung zumeist bei Wiederholungen oder Alternativen. Als Ergebnis des Vergleichs wird ein boolescher Wert (True / False) zurückgegeben. Meist wird eine Variable mit einem festen Wert oder eine Variable mit einer anderen Variablen verglichen.

(Hinweis: In Python werden auch Textvergleiche mit dem Operator „==“ durchgeführt.)

|  |  |  |
| --- | --- | --- |
| **Operator** | **Beispiel** | **Beschreibung** |
| < | a < b | a ist kleiner als b |
| <= | a <= b | a ist kleiner oder gleich b |
| > | a > b | a ist größer als b |
| >= | a >= b | a ist größer oder gleich b |
| == | a == b | a ist gleich b |
| != | a != b | a ist ungleich b |