2.3 Methoden: Aufträge und Anfragen

Jetzt bekommt das Konto Verhalten. Die Unterscheidung aus Abschnitt 1.8 kehrt hier unverändert wieder.

Merke

Eine Methode **ohne** Rückgabewert ist ein **Auftrag** an das Objekt — es soll etwas tun. Eine Methode **mit** Rückgabewert ist eine **Anfrage** an das Objekt — es soll etwas mitteilen.

Python

```python
class Konto:
    def __init__(self, inhaber):
        self.__inhaber = inhaber
        self.__stand = 0.0

    def einzahlen(self, betrag):            # Auftrag
        if betrag > 0:
            self.__stand = self.__stand + betrag

    def abheben(self, betrag):              # Anfrage: hat es geklappt?
        if betrag <= 0 or betrag > self.__stand:
            return False
        self.__stand = self.__stand - betrag
        return True

    def stand(self):                        # Anfrage
        return self.__stand

    def inhaber(self):                      # Anfrage
        return self.__inhaber
```

Object Pascal

```
type
  TKonto = class
  private
    FInhaber: String;
    FStand: Real;
  public
    constructor Create(inhaber: String);
    procedure Einzahlen(betrag: Real);
    function Abheben(betrag: Real): Boolean; virtual;
    function Stand: Real;
    function Inhaber: String;
  end;

procedure TKonto.Einzahlen(betrag: Real);
begin
  if betrag > 0 then
    FStand := FStand + betrag;
end;

function TKonto.Abheben(betrag: Real): Boolean;
begin
  if (betrag <= 0) or (betrag > FStand) then
```

```
23      Result := False
24    else
25    begin
26      FStand := FStand - betrag;
27      Result := True;
28    end;
29  end;
30
31  function TKonto.Stand: Real;
32  begin
33    Result := FStand;
34  end;
```

`abheben` ist der interessante Fall: Es *tut* etwas und *meldet* zugleich, ob es geklappt hat. Diese Mischform ist erlaubt und üblich — der Rückgabewert sagt nichts über den Zustand aus, sondern nur über den Erfolg des Auftrags.

Beispiel: Die Prüfung sitzt jetzt an genau einer Stelle

```
1 konto = Konto("A. Lovelace")
2 konto.einzahlen(120.0)
3
4 if konto.abheben(200.0):
5     print("ausgezahlt")
6 else:
7     print("Deckung reicht nicht:", konto.stand(), "€")
```

Ausgabe:

```
  Deckung reicht nicht: 120.0 €
```

Egal, von wie vielen Stellen im Programm abgehoben wird — geprüft wird immer hier, in der Methode `abheben`. Es gibt keine zweite Möglichkeit, an `__stand` heranzukommen.