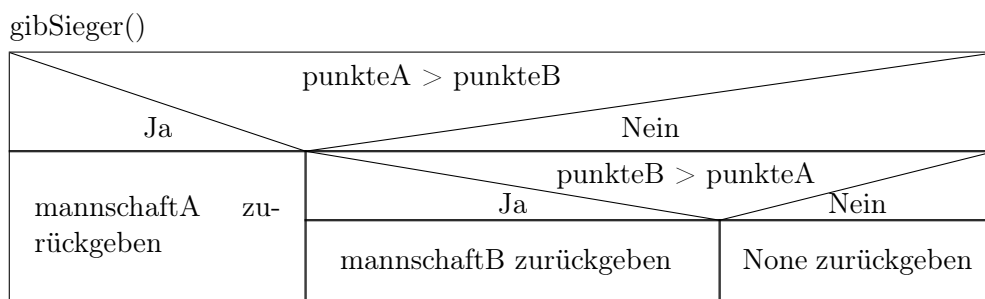


## Verzweigungen

Algorithmen, wie die Methoden der Klasse `Spiel`, müssen nicht immer nur lineare Abfolgen von Anweisungen beinhalten, sondern können auch Wahlmöglichkeiten enthalten. Man spricht hier üblicherweise von **Verzweigungen**. Eine Verzweigung gibt abhängig von einer Bedingung an, welcher von zwei Algorithmenschnitten im Folgenden durchlaufen werden soll. Durch eine Verschachtelung von Verzweigungen können auch mehr als zwei Möglichkeiten berücksichtigt werden.

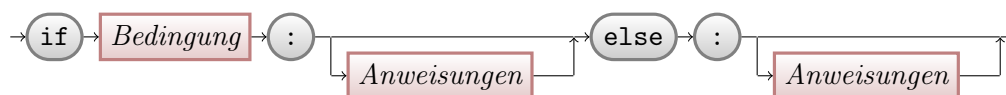
### Darstellung in Struktogrammen

Eine verschachtelte Verzweigung findet sich auch bei der Methode `gibSieger()` und könnte wie folgt aussehen. Ist die Bedingung `punkteA > punkteB` erfüllt, so wird der linke Abschnitt abgearbeitet, ansonsten der rechte. Im rechten Abschnitt wird dann erneut zwischen zwei Wahlmöglichkeiten unterschieden.



### Syntax in Python

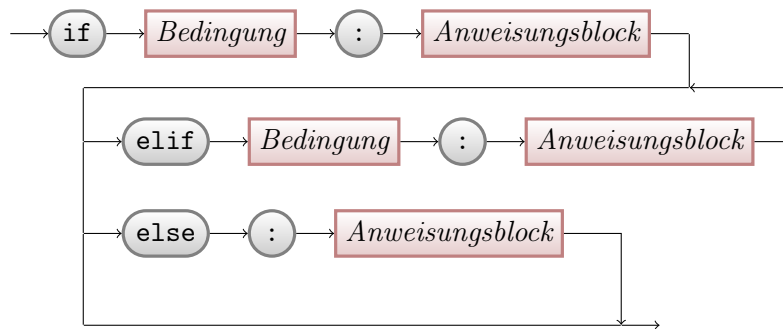
In Python dienen die Schlüsselwörter **if** (wenn), und **else** (sonst) zur Beschreibung einer einfachen Verzweigung. Die Reihenfolge kann man im folgenden Syntaxdiagramm sehen:



Sollen mehr als zwei Möglichkeiten berücksichtigt werden, muss man eine verschachtelte Verzweigung verwenden. Dafür gibt es das Schlüsselwort **elif** (sonst, wenn). Im folgenden Syntaxdiagramm wird von drei Möglichkeiten ausgegangen. Durch weitere Verschachtelungen können



allerdings auch noch mehr Möglichkeiten berücksichtigt werden.



Das oben angegebene Struktogramm der Methode `gibSieger()` könnte in Python wie folgt implementiert werden:

```

1 def gibSieger(self):
2     if self.punkteA > self.punkteB:
3         return self.mannschaftA
4     elif self.punkteB > self.punkteA:
5         return self.mannschaftB
6     else:
7         return None

```

## Arbeitsaufträge

- Integrieren Sie die oben beschriebene Methode `gibSieger()` in ihre Klasse `Spiel`.  
**WICHTIG:** Die Attributbezeichner Ihrer Klasse `Spiel` können sich von den oben verwendeten (`punkteA`, `punkteB`, `mannschaftA`, `mannschaftB`) unterscheiden! Passen Sie die Methode bei der Implementierung entsprechend an.
- Testen Sie die Methode, indem Sie sich von einer ausgewählten Spielbegegnung mit einer `print`-Anweisung den Sieger ausgeben lassen.
- Implementieren Sie analog zur Methode `gibSieger()` die Methode `gibVerlierer()`, die ebenfalls Teil der Klasse `Spiel` ist.
- Implementieren Sie die Methoden `gibPunkteSieger()` und `gibPunkteVerlierer()` innerhalb Klasse `Spiel`. Im Gegensatz zu den beiden Methoden aus Aufgabe 1 und Aufgabe 3 sollen nicht die Mannschaften zurückgegeben werden, sondern die erzielte Punktzahl des Siegers bzw. des Verlierers.
- Testen Sie die nun alle Methoden auf einmal, indem Sie sich von einer ausgewählten Spielbegegnung mit `print`-Anweisungen den Sieger mit seinen erzielten Punkten und den Verlierer mit seinen erzielten Punkten ausgeben lassen.

