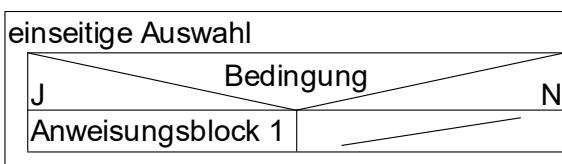


L1_4 Die Alternative

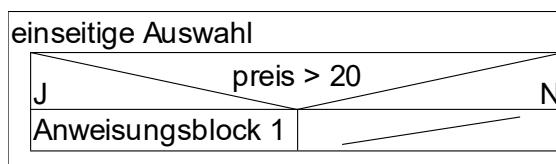
Häufig soll ein Programm etwas ausführen, das von einer aktuellen Situation abhängig ist. Beispielsweise könnte der Käfer *kara* prüfen, ob er auf einem Blatt steht und dieses aufheben, falls er tatsächlich auf einem Blatt steht, oder einen Schritt vorwärts gehen, wenn er nicht auf einem Blatt steht.

Dieses Vorgehen wird als „Fallunterscheidung“ bezeichnet: In dem einen „Fall“ soll das Programm etwas anderes machen als in einem anderen „Fall“. Bekannt ist die Situation vom Einsatz der Wenn-Funktion in der Tabellenkalkulation.

1.1 Die einseitige Auswahl (bedingte Verarbeitung)



(L1_4_1a_einseitige_Auswahl.stg)

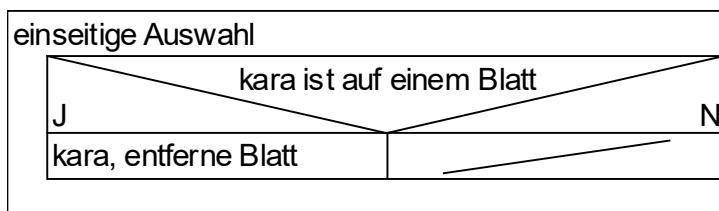


(L1_4_1b_einseitige_Auswahl.stg)

Nur wenn die Bedingung zutreffend (wahr) ist, wird der Anweisungsblock 1 durchlaufen. Ein Anweisungsblock kann aus einer oder mehreren Anweisungen bestehen. Trifft die Bedingung nicht zu (falsch), wird der Durchlauf ohne eine weitere Anweisung fortgeführt (Austritt unten).

1.2 Syntax einer einseitigen Auswahl

In der Bedingung wird unterstellt, dass nach der positiven Erfüllung gesucht wird. Auf die Angabe „ist wahr“ wird verzichtet.



(L1_4_1a_einseitige_Auswahl.stg)

```

10 if kara.onLeaf():
11   kara.removeLeaf()

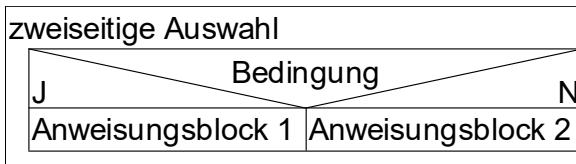
```

Im dargestellten Beispiel wird die Aktion *removeLeaf()* ausgeführt, wenn die Bedingung *onLeaf()* eintritt. Ansonsten passiert nichts.

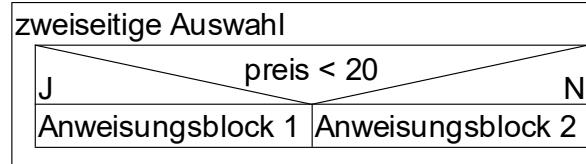
Das heißt, wenn der Käfer *kara* auf einem Blatt steht, liefert der Sensor *onLeaf()* den Wert *wahr* zurück. Tritt dieser Fall ein, bewirkt die Aktion *removeLeaf()*, dass er das Blatt entfernt.

In Python wird eine Fallunterscheidung mit der Anweisung *if - else* umgesetzt. Dabei wird die *if*-Bedingung mit einem Doppelpunkt abgeschlossen und in der/den Folgezeile/n die Anweisungen für den Fall, dass die Bedingung erfüllt ist, eingerückt notiert.

2.1 Die zweiseitige Auswahl (alternative Verarbeitung)



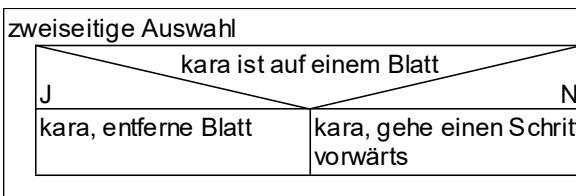
(L1_4_2a_einseitige_Auswahl.stg)



(L1_4_2b_einseitige_Auswahl.stg)

Wenn die Bedingung zutreffend (wahr) ist, wird der Anweisungsblock 1 durchlaufen. Trifft die Bedingung nicht zu (falsch), wird der Anweisungsblock 2 durchlaufen. Ein Anweisungsblock kann aus einer oder mehreren Anweisungen bestehen. Der Verzweigungsblock wird nach der Ausführung des jeweiligen Anweisungsblocks verlassen

2.2 Syntax einer zweiseitigen Auswahl



(L1_4_2c_einseitige_Auswahl.stg)

```

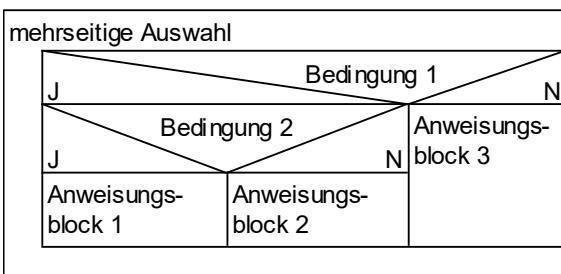
10 if kara.onLeaf():
11     kara.removeLeaf()
12 else:
13     kara.move()
  
```

Im dargestellten Beispiel wird die Aktion *removeLeaf()* ausgeführt, wenn die Bedingung eintritt. Ansonsten geht der Käfer *kara* einen Schritt - *move()* - vorwärts.

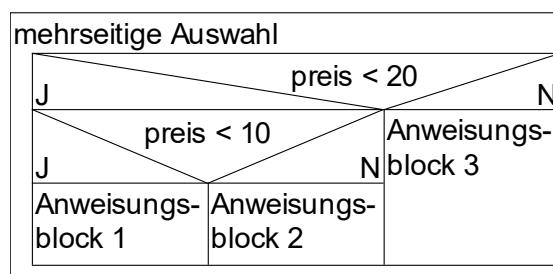
Das heißt, wenn der Käfer *kara* auf einem Blatt steht, liefert der Sensor *onLeaf()* den Wert *wahr* zurück. Tritt dieser Fall ein, bewirkt die Aktion *removeLeaf()*, dass er das Blatt entfernt. Wenn der Käfer *kara* auf keinem Blatt steht, geht er einen Schritt vorwärts.

In Python werden die Anweisungen für den Nein-Fall der if-Bedingung mit dem Schlüsselwort *else:* eingeleitet. Die Anweisung für den Fall, dass die Bedingung nicht erfüllt ist, werden in der/den Folgezeile/n eingerückt notiert.

3.1 Die mehrseitige Auswahl



(L1_4_3a_einseitige_Auswahl.stg)

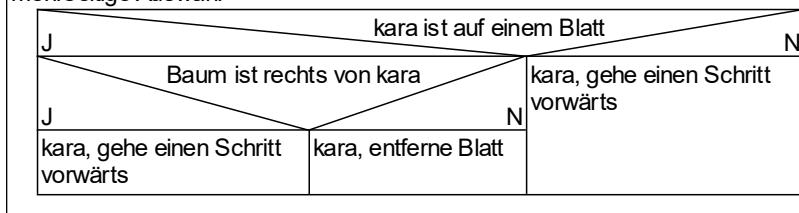


(L1_4_3b_einseitige_Auswahl.stg)

Die mehrseitige Auswahl wird auch „verschachtelte“ Auswahl genannt, da eine weitere Bedingung folgt. Die Verschachtelung ist ebenso im Nein-Fall möglich.

3.2 Syntax einer mehrseitigen Auswahl

mehrseitige Auswahl



(L1_4_3c_einseitige_Auswahl.stg)

```

10 if kara.onLeaf():
11     if kara.treeRight():
12         kara.move()
13     else:
14         kara.removeLeaf()
15     else:
16         kara.move()
  
```

Im dargestellten Beispiel wird die Aktion *move()* ausgeführt, wenn beide Bedingungen eingetreten sind. Ist nur die erste, nicht aber die zweite Bedingung erfüllt, wird die Aktion *removeLeaf()* ausgeführt. Wenn die erste Bedingung nicht erfüllt ist, wird die Aktion *move()* ausgeführt.

Das heißt, wenn der Käfer *kara* auf einem Blatt steht, liefert der Sensor *onLeaf()* den Wert *wahr* zurück. Ist dies der Fall, wird zusätzlich geprüft, ob ein Baum rechts von ihm steht. Trifft auch das zu, liefert der Sensor *treeRight()* den Wert *wahr* zurück. Die Aktion *move()* bewirkt, dass er einen Schritt geradeaus geht.

Liefert der Sensor *treeRight()* den Wert *false*, bewirkt die Aktion *removeLeaf()*, dass der Käfer *kara* das Blatt entfernt.

Wenn der Sensor *onLeaf()* den Wert *false* zurückgibt, führt die Aktion *move()* dazu, dass der Käfer *kara* einen Schritt vorwärts geht.