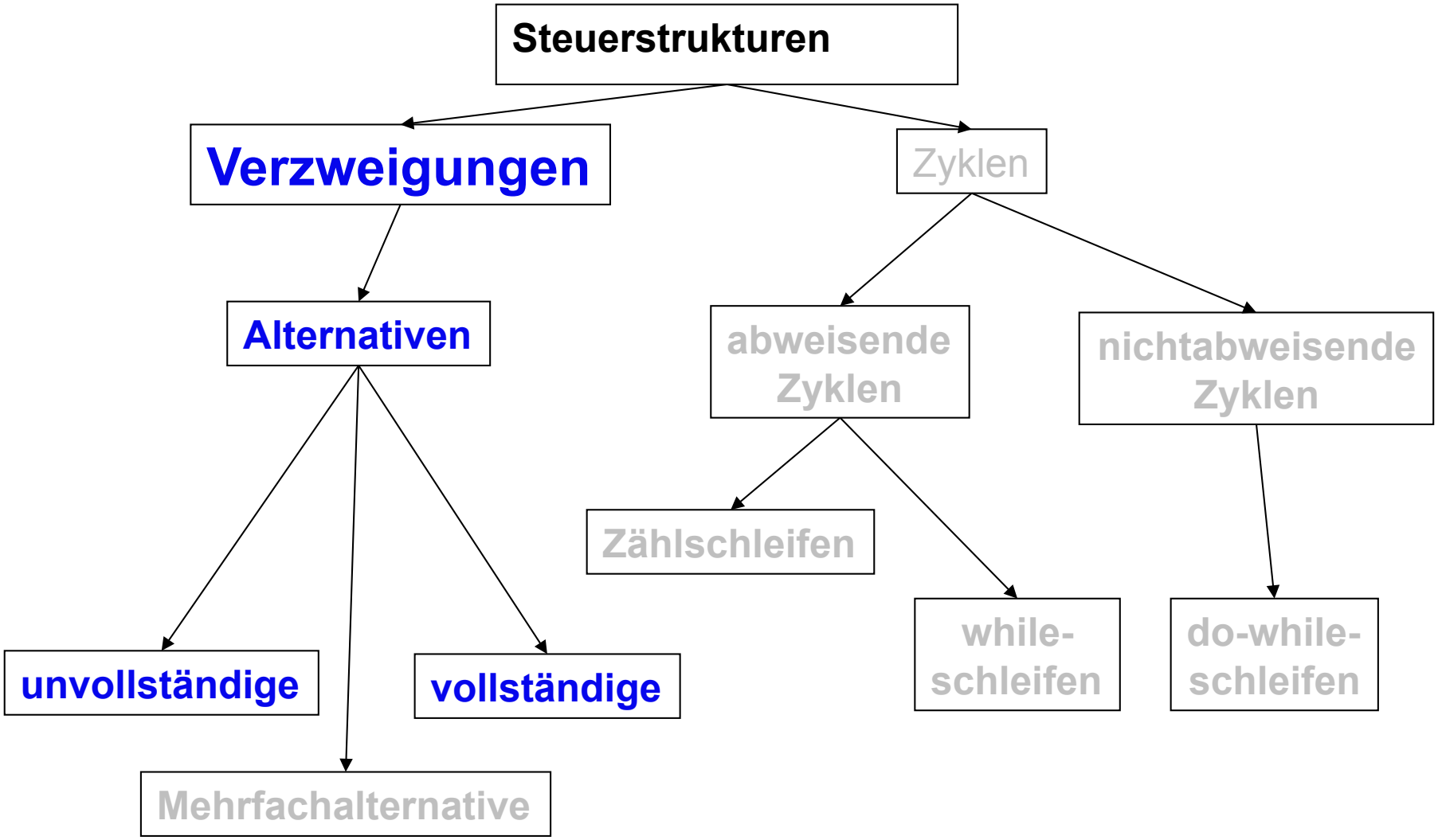
INF FAKULTÄT FÜR INFORMATIK

Java-Programmierung mit NetBeans

Steuerstrukturen

Dr. Henry Herper – Otto-von-Guericke-Universität Magdeburg - WS 2012/13

Steuerstrukturen



Alternative – if - Anweisung

Bei der unvollständigen Alternative wird die Anweisung im Verarbeitungsteil nur ausgeführt, wenn der Ausdruck für die Bedingung den Wert wahr (true) ergibt. Ist die Bedingung nicht erfüllt, so wird die Abarbeitung bei der Folgeanweisung fortgesetzt.

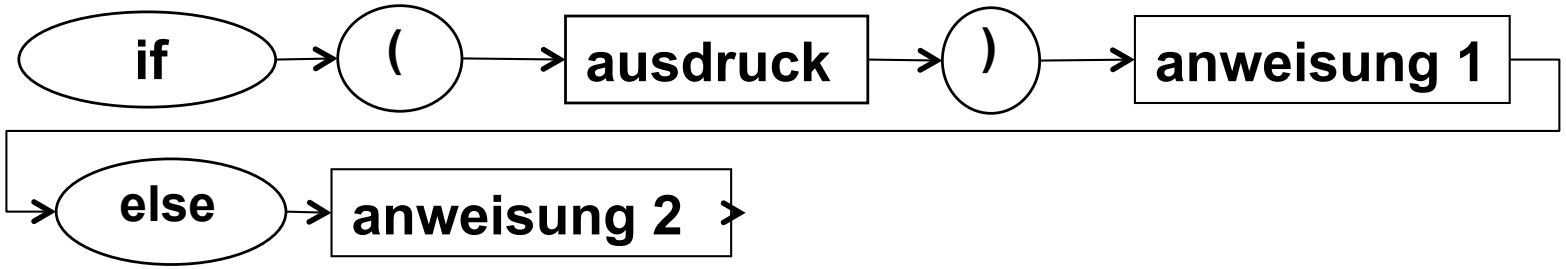
Beispiele:

```
if ( a < 0 )
  System.out.println("Wert: " + a + " ist kleiner als 0");

if ( a < b ) {
  System.out.println(a + " < " + b);
  a = 2 * a;
  }
```

Alternative – if - Anweisung

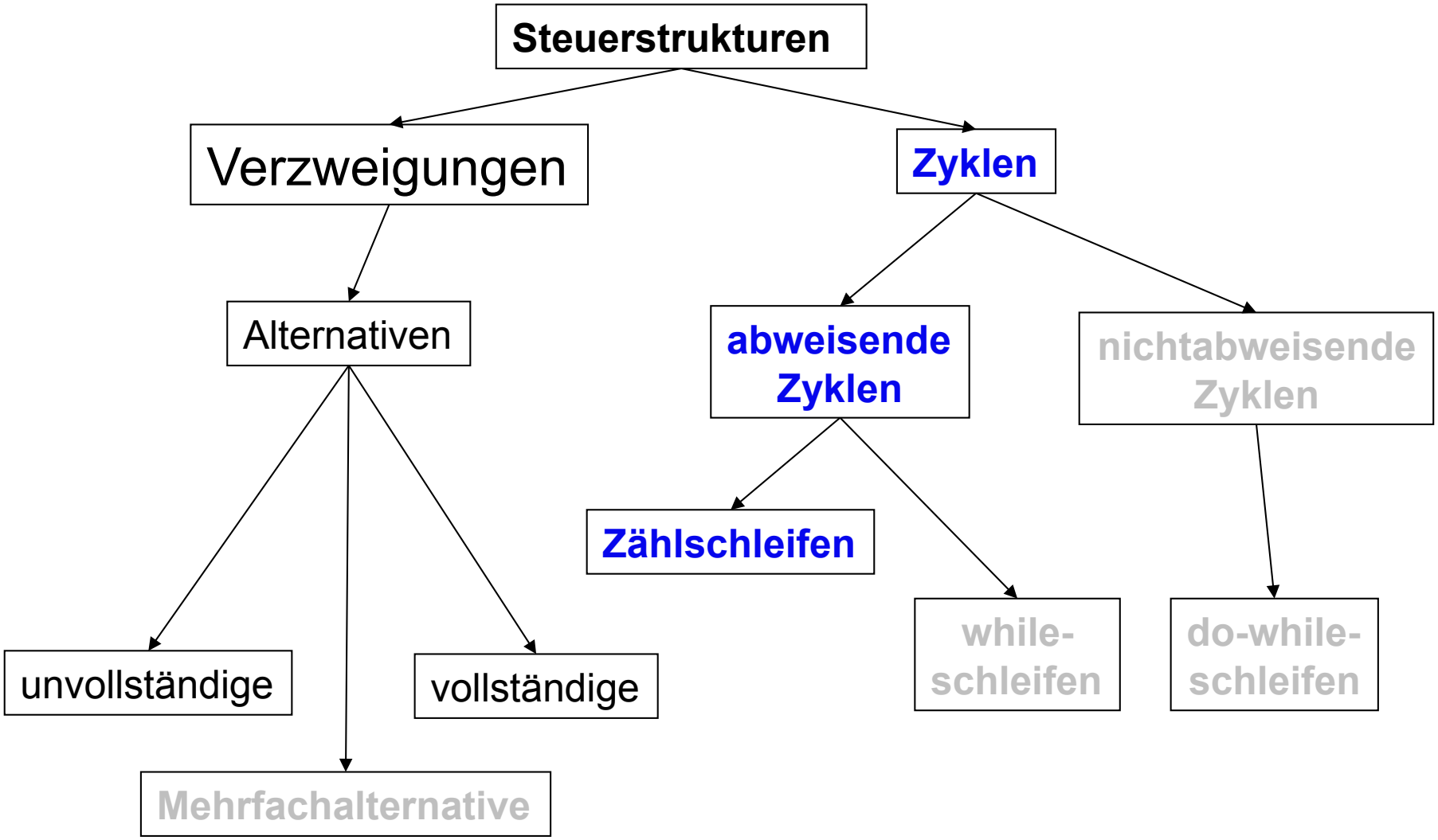
Bei der vollständigen Alternative wird die Anweisung 1 im Verarbeitungsteil ausgeführt, wenn der Ausdruck für die Bedingung den Wert wahr (true) ergibt. Ist die Bedingung nicht erfüllt, so wird die Anweisung 2 ausgeführt.



Beispiel:

```
if ( a < b )
  System.out.println(a + " < " + b);
else
  System.out.println(a + " >= " + b);
```

Steuerstrukturen

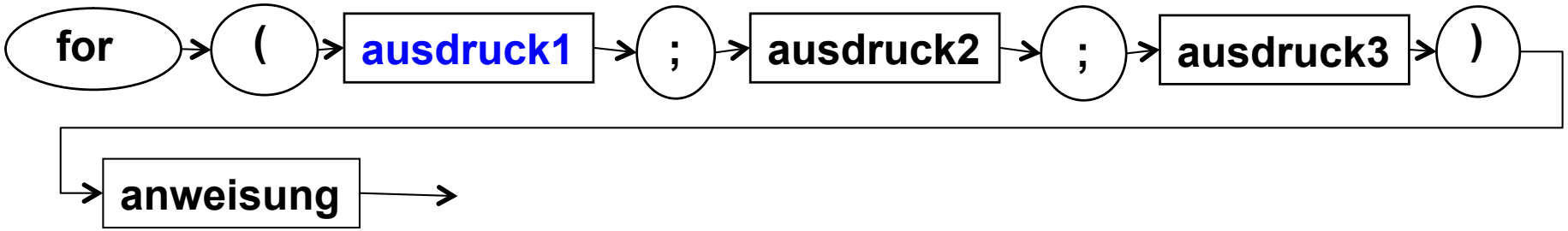


Zyklen - Iterationen

Das Ziel von Zyklen besteht darin, bestimmte **Programmabschnitte mehrfach zu durchlaufen**. Die Anzahl der Durchläufe ist von dem Zustand der Abbruchbedingung abhängig.

Schleifenanweisungen ermöglichen die Implementierung **iterativer Zyklen**.

Abweisende Zyklen - Zählschleifen

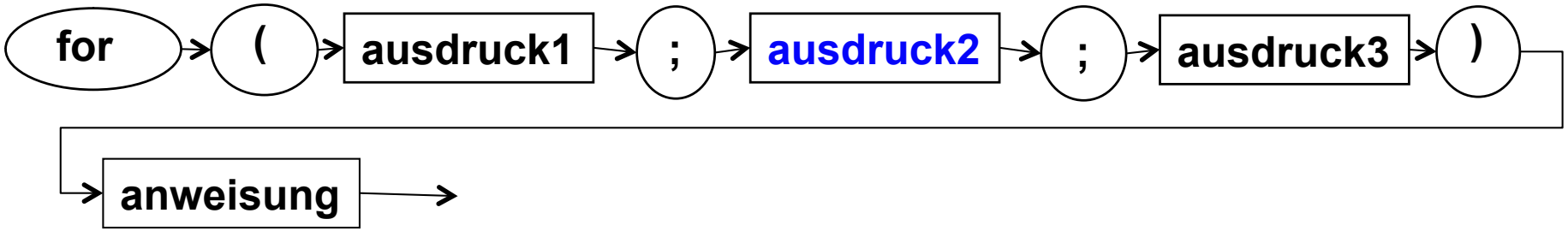


Die for-Schleife hat folgenden Syntax:

```
for ( ausdruck1; ausdruck2; ausdruck3)
      anweisung
```

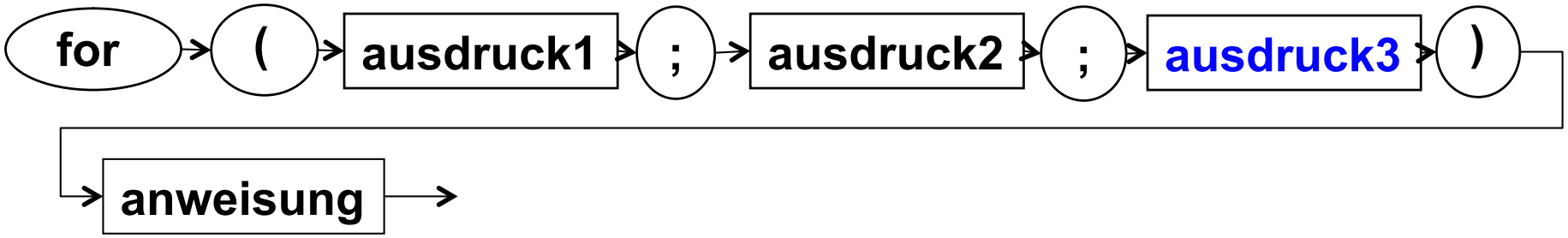
Bei der Abarbeitung der for-Schleife wird zuerst der ausdruck1 ausgeführt, was einer Initialisierung entspricht, da dieser Ausdruck nur einmal ausgeführt wird. Damit der Anfangswert der Schleifenvariablen gesetzt.

Abweisende Zyklen - Zählschleifen



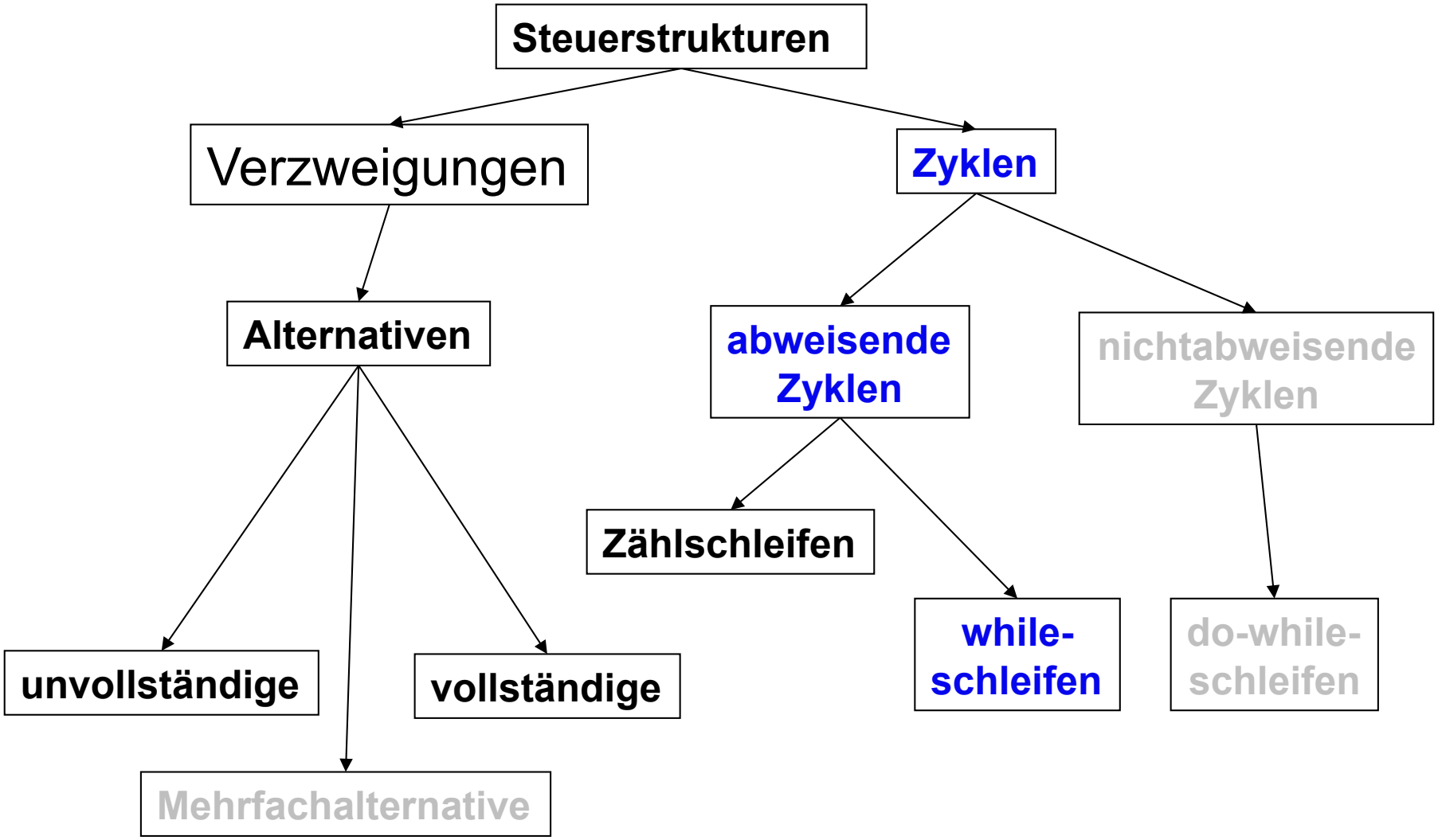
Mit dem ausdruck 2 wird die Schleifenbedingung festgelegt. Der Zykluskörper wir solange abgearbeitet, solange Ausdruck 2 den Wahrheitswert wahr liefert. Hat Ausdruck 2 nach der Initialisierung den Wahrheitswert falsch, so wird keine Anweisung des Zykluskörpers abgearbeitet ➔ abweisender Zyklus.

Abweisende Zyklen - Zählschleifen



Ist der Zykluskörper abgearbeitet, so erfolgt eine Reinitialisierung durch ausdruck3. Bei der Verwendung als Zählschleife wird der ausdruck3 zur Inkrementierung bzw. zur Dekrementierung genutzt.
Bei der Reinitialisierung können mehrere Anweisungen, durch Komma (Kommaoperator) getrennt angegeben werden.

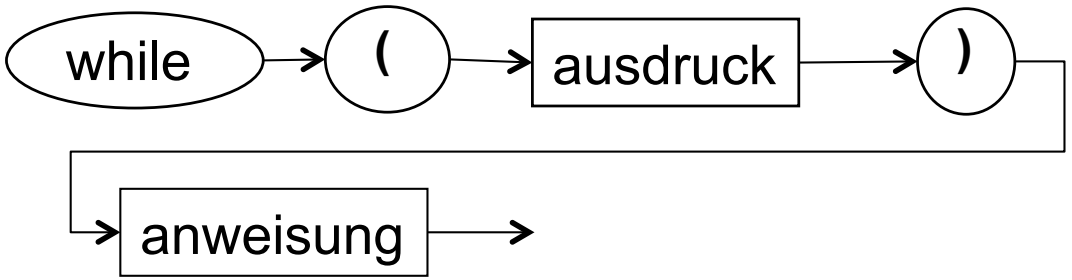
Steuerstrukturen



Abweisende Zyklen – while-Schleifen

```
while(schleifenbedingung)
  schleifenrumpf
```

Die while - Anweisung wird auch als "Solange Bedingung erfüllt, wiederhole Anweisung" übersetzt. Sie ist eine Zyklusart, bei der die Wiederholzahl vor Ausführung nicht bekannt ist. Die while-Schleife hat folgende Syntax:



Abweisende Zyklen – while-Schleifen

```
while ( ausdruck )
      anweisung
```

Falls bzw. solange der Ausdruck den Wert wahr (true) liefert, wird die Anweisung wiederholt abgearbeitet. Sollen mehrere Anweisungen abgearbeitet werden, so müssen sie zu einer Blockanweisung zusammengefasst werden.

Eine while-Schleife kann immer eine for-Schleife ersetzen.

Terminiertheit von Schleifen

Es ist zu beachten, dass die Schleife terminiert (beendbar) ist. Dazu sind folgende Überprüfungen notwendig:

- Ist in der Anweisung überhaupt eine Variable vorhanden, die der Bedingung den für den Abbruch notwendigen Wert geben kann?

- Wird diese Bedingung jemals erreicht?

Aufgabe: Kürzen von Brüchen

Das Prinzip des euklidischen Algorithmus wird auch *gegenseitige Wechselwegnahme* genannt. Eingangsgrößen sind zwei natürliche Zahlen a und b. Bei der Berechnung verfährt man nach Euklid (Subtraktionsverfahren) wie folgt:

1. setze m = a; n = b
2. ist m < n, so vertausche m und n
3. berechne r = m - n
4. setze m = n, n = r
5. ist r ≠ 0 fahre fort mit Schritt 2

Nach Ablauf des Verfahrens hat man mit m den ggT von a und b gefunden.