

## Programmschleifen (Wiederholungen)

Programmschleifen sind vordefinierte Ablaufstrukturen, welche die kontrollierte Wiederholung eines definierten Programmabschnitts, des Schleifenkörpers, ermöglichen. Die Kontrolle erfolgt in Form eines einfachen oder zusammengesetzten **Bedingungs-ausdrucks**, der auf seinen Wahrheitswert geprüft wird und daher einen logischen Wahrheitswert (`true` / `false`) ergeben muß. In Java darf kein Zahlenwert substituiert werden, wie es in C/C++ üblich ist.

### while-Schleife

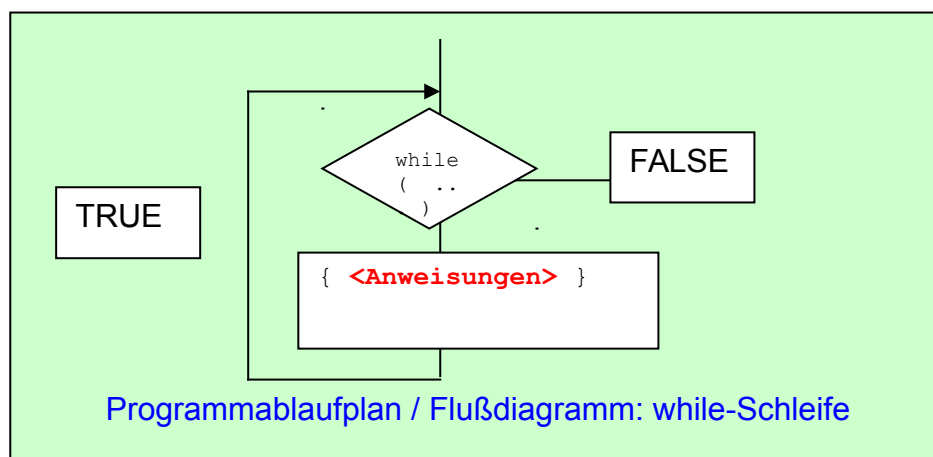
Die einfachste Form der kontrollierten Wiederholungen ist die while-Schleife. Sie ist eine kopfgesteuerte Schleife – das heißt, der Bedingungs-ausdruck wird vor Betreten des Schleifenkörpers geprüft. Das bedeutet: Mindestanzahl der Durchläufe  $\Rightarrow$  0.

```
while( <Ausdruck> ) {
    <Schleifenkörper>
}
```

Der Schleifenkörper wird so lange wiederholt durchlaufen, wie der Bedingungs-Ausdruck in den runden Klammern den Wahrheitswert TRUE ergibt. Die while-Schleife wird abgebrochen, wenn der Bedingungs-ausdruck den Wahrheitswert FALSE ergibt.

Als Bedingungs-Ausdruck ist jeder einfache oder zusammengesetzte Wert erlaubt, der in Java als Wahrheitswert interpretiert wird. Häufig werden im Bedingungs-ausdruck relationale und logische Operatoren eingesetzt, die einen Wahrheitswert zum Ergebnis haben.

```
while (s == 1 && t < 9 || 17 != 15 && s==0) {
    System.out.println( ""+ (s++));
    ...
}
```



Eine kontrollierte Endlosschleife wird mit Hilfe der Anweisung `while ( true )` in Kombination mit einer `break`-Anweisung innerhalb des Schleifenkörpers programmiert

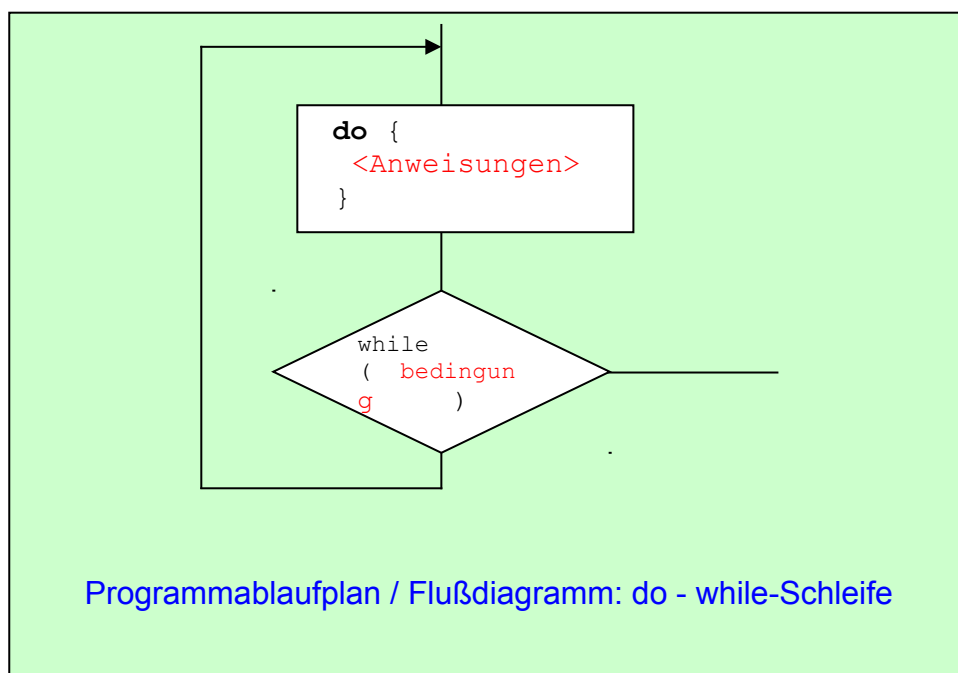
```
while( true ) {
    ...
    if (s > 1) break;
}
```

### Do – while-Schleife

Die `do-while`-Schleife führt den Anweisungsblock sofort aus, wie die Bedingungsprüfung den Wert `TRUE` ergibt. Die Bedingungsprüfung erfolgt am Ende der Schleife, so daß der Schleifenkörper auf jeden Fall mindestens einmal durchlaufen wird - deshalb wird sie auch fußgesteuerte Schleife genannt.

```
do {
    <Schleifenkörper>
}
while( <ausdruck> ) ;
```

Die `do-while` muss mit einem **Semikolon** abgeschlossen werden, damit der Java-Compiler sie von einer unvollständigen `while`-Schleife unterscheiden kann.



## FOR-Schleife

Ein aufgrund seiner Flexibilität häufig eingesetzter Wiederholungstyp ist die for-Schleife. Sie erlaubt die genaueste Kontrolle über die Bedingungen der Wiederholung. Die for-Schleife wird auch als zählende Schleife bezeichnet, weil der zusammengesetzte Bedingungsausdruck eine Zählvariable einschließt, deren Wertebereich geprüft wird. Der gesamte Bedingungsausdruck besteht aus drei Teil-Ausdrücken, die durch Semikola getrennt sind:

- **Initialisierung:** Die Zählvariable wird auf einen frei definierbaren Anfangswert gesetzt.
- **Bereichsprüfung der Zählvariablen**
- **Setzen der Schrittweite der Veränderung der Zählvariablen**

```
for ( Anfangswert; Bereichsprüfung; Schrittweite) {  
    <Schleifenkörper>  
}
```

```
for (int i=10; i > 0; i--) {  
    System.out.println(""+i);  
}
```

Im Unterschied zu C muß die Zählvariable nicht außerhalb (vor) der Schleife deklariert werden, sondern sie kann innerhalb des Bedingungsausdrucks definiert werden. Dies ist von Vorteil, wenn der Zählwert nur innerhalb der Schleife benutzt wird.

Es können nötigenfalls auch **mehrere Zählvariablen** verwendet werden, die durch Kommata abgetrennt werden.

```
for (int i=10, j=5; i<20 && j>5; i--,j++) {  
    System.out.println("i: "+i+" j: "+j);  
}
```

