`while` Schleife

Syntax:

```
while (Bedingung)          // die runden Klammern sind Pflicht
     Eine-Anweisung;       // Einrücken erhöht die Übersichtlichkeit
```

oder:

```
while (Bedingung)
{    Eine-Anweisung;
     Noch-eine-Anweisung;
     ...
}
```

Die von `while` abhängige Anweisung (bzw. der ganze `{}`-Block) wird solange wiederholt, wie die *`Bedingung`* wahr (als numerischer Ausdruck ungleich 0 in C/C++ oder als logischer Ausdruck `true` in C++/Java) ist. Ist die *`Bedingung`* schon am Beginn falsch (= 0 bzw `false`), so wird die Schleife **NIE** durchlaufen.

```
while (x > 0)
     x++;          // Warte auf den Überlauf

while (1)          // die EWIGE Schleife in C/C++, in Java/C++: while (true)
     x++;

while (0)          // die NIEMALS durchlaufene Schleife in C/C++, Fehler in Java
     x++;
```

Eine sehr nützliche Eigenschaft der Streameingabe `>>` von C++ ist, dass sie bei der Verwendung als Bedingung (in einer Schleife oder in einem `if`) den Erfolg oder Misserfolg der Einleseoperation zurückgibt!

```
while (cin >> x)      // lies Werte ein, solange welche da sind
     cout << "Habe erfolgreich " << x << " eingelesen!\n";
```

Fehler:

```
while (x > 0);        // Der Strichpunkt zählt als leere Anweisung
     x++;             // ist NICHT MEHR in der Schleife

while (x > 0)
     cout << "x: " << x << '\n';
     x++;          // nicht mehr in der Schleife
```