

Definition und Deklaration von Variablen

Wenn eine Variable in einer Klasse oder Methode definiert wird, wird der Compiler dazu aufgefordert die Variable mit der selbst gewählten Bezeichnung und dem angegebenen Datentypen anzulegen.

```
int zahl = 0; // Definition
```

Eine Variable kann in ihrem Gültigkeitsbereich nur einmal existieren. Das folgende Beispiel wäre somit nicht möglich und der Compiler würde einen Fehler werfen.

```
int zahl = 0;  
  
int zahl = 1;
```

Dies ist die klassische **Variablendefinition**. Nun kann es Fälle geben, bei denen man eine Variable verwenden möchte, die an andere Stelle – bspw. in einem anderen Quelldatei – definiert worden ist. In diesen Fällen wird dann eine **Deklaration** genutzt.

Bei einer Deklaration wird dem Compiler mitgeteilt, dass es eine Variable mit dem vorgesehenen Namen und dem angegebenen Datentyp existiert. Durch die Deklaration hat der Compiler zusammen mit dem Datentyp und dem Bezeichner die wichtigsten Informationen vorliegen.

Um eine Variable nur zu deklarieren wird das Schlüsselwort **extern** davor gesetzt.

```
extern int zahl; // Deklaration
```

i

Der Compiler reserviert bei einer Deklaration keinen Speicherplatz für die Variable (im Beispiel *zahl*). Er kennt den Bezeichner und den Datentyp der Variablen und verlässt sich darauf, dass die Variable an einer anderen Stelle - wie bspw. in einer anderen Quelltextdatei - definiert wurde.

Eine Deklaration darf im Gegensatz zu einer Definition mehrmals im Code vorkommen. Da eine Definition wie eine Deklaration Bezeichner und Datentyp angibt, ist sie auch immer eine Definition. Andersherum gilt dies nicht.

[←Datentypen in C++](#)

[Ausgabe mit std::cout→](#)