

6.7.F Extras/Typdefinitionen mit typedef – Ergänzungen und Bilder

6.7.F.1 Speicherklassen, Type-Qualifier und Type-Specifier

Damit bei Variablen-, Typ- und Funktionsdeklarationen (=Prototypen) nur gültige Kombinationen entstehen, hat man die passenden Schlüsselworte in drei Gruppen geteilt.

- a) Speicherklassen
- b) Type-Qualifier
- c) Type-Specifier

Eine Variablen-, Typ- oder Funktionsdeklaration kann immer nur *eine* Speicherklasse und maximal *einen* Type-Qualifier haben.

Folgende Speicherklassen sind möglich:

- auto — Adresse wird zur Laufzeit zugeordnet (wird kaum benutzt; in C++ bedeutet es, dass der Typ passend zur Initialisierung gewählt wird)
- register — Variable soll im CPU-Register gespeichert werden, Adressoperator daher nicht anwendbar
- static —
 - in Funktion: Lebensdauer gleich Programmdauer;
 - außerhalb: Sichtbarkeit auf Modul (Compilationseinheit) beschränkt
- extern — Speicherplatz an anderer Stelle
- typedef — (!) keine Variable oder Prototyp, sondern Typ
- (ohne) —
 - in Funktion: wie auto
 - außerhalb einer Funktion:
 - * bei Variable: static
 - * bei Prototyp: extern

Folgende Type-Qualifier sind möglich:

- const
- volatile

Und diese Type-Specifier gibt es bekannterweise:

- char
- int
- float
- double
- short
- long
- signed
- unsigned
- void