

Notizen AIK ProgTech 3: Die Initialisierungs-Liste

Klaus Kusche

Wo & Wie?

- Im Konstruktor
- In der Kopfzeile nach der `)` der Parameter, vor der `{` mit dem Code:
... `) : member1(wert1), member2(wert2), ... { ...`
Bei Mitgliedern, die Objekte sind: ... `memberx(Konstruktor-Argumente) ...`
Als Erstes bei abgeleiteten Klassen: ... `) : Vaterklasse(Konstruktor-Argumente), ...`
- In der Reihenfolge der Member!!!
- **Achtung:** Man kann nur Member der eigenen Klasse (hier deklariert) initialisieren!
Member, die man von der Vaterklasse geerbt hat, müssen im Konstruktor der Vaterklasse initialisiert werden!

Funktion

- Die Member werden gleich beim Anlegen des Objektes mit den angegebenen Werten initialisiert.
- Wenn ein Member selbst wieder ein Objekt ist:
Es wird durch den Aufruf seines Konstruktors mit den angegebenen Argumentwerten initialisiert.

Warum?

- Schöner als normaler Zuweisung im Code des Konstruktors.
- Bei Mitgliedern, die selbst wieder Objekte sind:
Effizienter als das Anlegen mittels Standard-Konstruktor und nachträgliches Ändern auf die richtigen Werte mittels Set-Methoden:
Gleich mit dem richtigen Konstruktor und den richtigen Werten anlegen!
- Sicherer (man vergisst weniger leicht, ein Member zu initialisieren).

Wann notwendig?

- Bei Mitgliedern, die selbst wieder Objekte sind und keinen Standard-Konstruktor haben:
Zum Aufruf eines Konstruktors (geht nur in der Init-Liste, nachher nicht mehr!).
- Bei abgeleiteten Klassen, wenn explizit ein Konstruktor der Vaterklasse aufgerufen werden soll (geht nur in Init-Liste, nachher nicht mehr!).
- Bei Mitgliedern, die Referenzen oder **const** sind:
Solche Member können nur durch Initialisierung (in der Init-Liste) gesetzt werden, nicht nachträglich im Code des Konstruktors durch Zuweisung.