

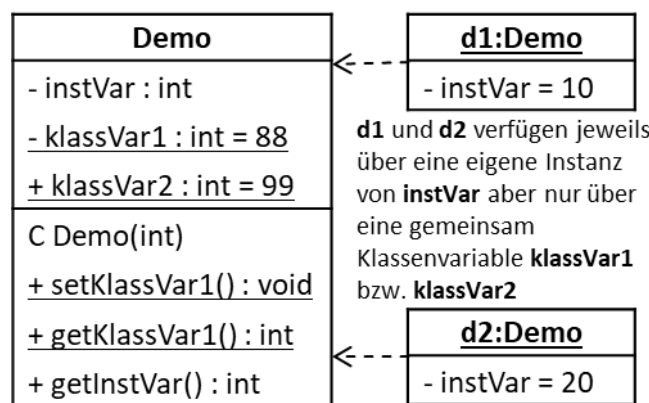


Klassenspezifische Daten sind Daten, die *nur einmal* für *alle* Objekte einer Klasse existieren. Sie sind also nicht an einzelne Objekte, sondern an alle Objekte einer Klasse gleichermaßen gebunden. ... Diese von mehreren Objekten einer Klasse *gemeinsam* benutzbaren Daten müssen nicht global sein. Sie können genauso gut gekapselt werden wie andere Daten eines Objekts. Sie sind dann *innerhalb einer Klasse* und *nur für Objekte dieser Klasse* gleichermaßen zugreifbar. Diese Daten sind **static** – eine weitere Bedeutung dieses Schlüsselworts (die bisher bekannten sind »Variable für Funktionen mit Gedächtnis« und »nur in dieser Datei gültige Deklaration«). Bei der Definition und Initialisierung wird für eine beliebige Anzahl von Objekten einer Klasse nur *ein* Speicherplatz pro **static**-Element angelegt, auf den von *allen* Objekten dieser Klasse zugegriffen werden kann. Klassenspezifische Funktionen führen Aufgaben aus, die *an eine Klasse, nicht aber an ein Objekt gebunden sind*. Sie können zum Beispiel mit den **static**-Daten arbeiten. ...

Quelle: Ulrich Breymann (2017) „Der C++ Programmierer“ / 5.Auflage

```
class Demo {
private:
    // Instanzvariable
    int instVar;
    // Klassenvariable
    static int klassVar1;
public:
    static int klassVar2;

public:
    int getInstVar() {
        return instVar;
    }
    static int getKlassVar1() {
        return klassVar1;
    }
    static void setKlassVar1(int x) {
        klassVar1 = x;
    }
    Demo(int x) {
        this->instVar = x;
    };
};
// Initialisierung
int Demo::klassVar1 = 88;
int Demo::klassVar2 = 99;
```



d1 und d2 verfügen jeweils über eine eigene Instanz von **instVar** aber nur über eine gemeinsam Klassenvariable **klassVar1** bzw. **klassVar2**

Der Zugriff auf Klassen-Member erfolgt über:
Demo::klassVar2 bzw. **Demo::getKlassVar1()**

Da d1 und d2 einen gemeinsamen Zugriff besitzen, ist für die öffentliche Klassenvariable auch der Aufruf **d1.klassVar2** bzw. **d2.klassVar2** möglich. Beide Varianten greifen aber auf die gleiche Variable zu!

Ein häufiger und typischer Anwendungsfall für Klassenvariablen ist das Zählen der Objektinstanzen einer Klasse. Dazu wird im Konstruktor eine entsprechende Variable inkrementiert. Zur korrekten Funktionsweise ist zu beachten, dass dementsprechend im Destruktor dekrementiert werden muss.

