static - Klassenmember

- existieren nur einmal für alle Objekte der Klasse, sind auch über die Notation
  klassenname::membername aufrufbar

- **static-Klassenmember** sind per Definition **eingebettete Klassen**, **typedef-Variablen**, **Enumerationen** und mit **static** vereinbarte **Datenmember** und **Methoden** (vgl. cla1.cpp):

```
class Foo { public: class Bar { /* ... */ };    // eingebettete Klasse
                    typedef int MyInt;          // typedef
                    enum { RED, BLUE, GREEN };
                    static int statMem;
                    static int statMeth(){ return (int)BLUE; }
                  //   weitere Member/Methoden
};

int Foo :: statMem = 5;  //static Datenmemer ausserhalb der Klasse deklarieren u. initialisieren

void main(){ Foo::MyInt myColor = (MyInt)Foo::GREEN; // .....  }
```

static - Klassenmember

- **static-Member** sind ein sicherer Ersatz für **globale Variablen** und **Funktionen** aus **C** , da public **static-Member** über den Klassennamen ebenfalls global aufrufbar, jedoch nur im Kontext mit der Klasse (eingeschränkt) nutzbar sind.

- **static-Member** **müssen** ausserhalb der Klasse noch einmal **deklariert** werden, eine **Initialisierung** ist **optional** möglich. Fehlt der Initialisierungswert, wird automatisch mit **0** initialisiert. Ein **Fehlen** der expliziten Deklaration verursacht einen **Fehler des Linkers**.

- **static-Methoden** dürfen nur auf **static-Member** zugreifen,

- **static-Member** berücksichtigen ebenfalls die Zugriffsspezifizierer **private**, **public**, **protected**

- **static-Methoden** dürfen **this** nicht nutzen ! (vgl. cla1e.cpp)

- **static-Member** werden u.a. zum Zählen der Anzahl der Objekte einer Klasse benutzt (vgl. cla6)

- **static-Komponenten** werden **nicht** von **sizof** oder **new** berücksichtigt !

- **Klassen** mit ausschliesslich **static-Membern** und **-Methoden** sind möglich (vgl. cla12, cla12a)