

statický datový člen třídy (o)

- vytváří se pouze jeden na třídu, společný všem objektům třídy
- např. počítání aktivních objektů třídy, společné nastavení pro všechny prvky třídy, zabránění vícenásobné inicializaci, zabránění vícenásobnému výskytu objektu ...
- v deklaraci (*.h) třídy označen jako static

```
class string {  
    static int pocet;  
    // deklarace statické proměnné nerezervuje paměť  
};
```

- existuje i v okamžiku, kdy neexistuje žádný objekt třídy
- není součástí objektu, vytvoří se jako globální proměnná (nutno definovat a inicializovat ve zdrojovém souboru *.cpp)

```
int string::pocet = 0;
```

statické metody (o)

- pouze jedna na třídu
- je možné ji volat aniž by existoval (jediný) objekt dané třídy
- nemá this (= „normální“ funkce s přístupem k privátním datům)
- ve třídě označená static
- vlastní tělo ve zdrojové části
- nesmí být virtuální
- může k private členům (obdobně jako friend funkce),
- externí volání se jménem třídy bez objektu **Třída::fnc()**

statické členy – použití v programu

// v *.h popis

```
class Třída { public:  
    static int Pocet; //staticka data  
    static int Kolik (void); //staticka metoda  
    // příklady použití/přístupu v rámci třídy  
    int Prvku1(void){return Pocet;}  
    int Prvku2(void){return Kolik();}  
}
```

// v souboru *.cpp kód – pouze jednou, plná cesta

```
int T::Pocet = 0; // místo v paměti a inicializace  
int T::Kolik() {return Pocet;} // kód funkce
```

// způsob použití v programu – celá cesta

```
T::Pocet = 1; // pro blokování, chceme-li jen jeden prvek  
int p1 = T::Pocet; // je-li public  
int p2 = T::Kolik(); // je-li public  
T a;  
int p3 = a.Prvku1(); // pro metodu musí existovat prvek T
```