

Hlavičkové soubory a třída (o)

- hlavičkový soubor (.h), inline soubor (.inl, nebo .h), zdrojový soubor (.cpp)
- hlavička – deklarace třídy s definicí proměnných a metod, a přístupových práv ("těla" inline metod – lépe mimo) – předpis, netvoří kód
- inline soubor – "těla" inline metod – předpis. Jelikož jsou těla mimo definici třídy, musí obsahovat jméno třídy do které patří (a operátor přístupu). Mimo definici třídy je nutné k metodám uvádět, ke které třídě patří pomocí operátoru příslušnosti T::metoda
- zdrojový soubor – "těla" metod – "skutečný" kód. Jelikož jsou těla mimo definici třídy, musí obsahovat jméno třídy do které patří (a operátor přístupu)
- V souborech kde je třída používána musí být vložen hlavičkový soubor třídy.

hlavička: (soubor.h)

```
class T{  
data  
metody (bez "těla"); // těla v cpp  
inline metody (bez "těla"); // tělo je v h souboru  
metody (s „tělem) {} // inline metody  
};
```

těla metod přímo v hlavičce, nebo přidat
#include "soubor.inl"

soubor.inl obsahuje těla inline metod: **T::těla metod**
návratová hodnota **T::název(parametry) {...}**

zdrojový kód:

```
#include hlavička  
T::statické proměnné  
T::statické metody  
T::těla metod
```

soubory, kde je třída používána

```
#include "hlavička"
```

použití třídy

```
//===== konkrétní příklad ===== hlavičkový soubor
class POKUS {
int a;
public:
POKUS(void) { this->a = 0;} //má tělo -> inline, netvoří kód

inline POKUS(int xx); //označení -> inline, netvoří kód

POKUS(POKUS &cc); // nemá tělo, ani označení
// -> není inline = funkční volání = generuje se kód
};
// pokračování hlavičkového souboru
// nebo #include "xxx.inl" a v něm následující

// je inline, proto musí být v hlavičce protože je mimo tělo
// třídy musí být v názvu i označení třídy (prostoru)
POKUS::POKUS(int xx) { this->a = xx; }
// konec hlavičky (resp. inline)

// zdrojový soubor
#include "hlavička.h"
POKUS::POKUS (POKUS&cc) { this->a = cc.a; }
```