

inline metody (o)

- obdoba inline funkcí pro třídu
 - slouží pro zápis krátkých metod
 - rozbalené do kódu, předpis, netvoří kod
-
- automaticky jsou to metody s „tělem” v deklaraci třídy (hlavičkový soubor)
 - inline jsou i metody v deklaraci třídy s klíčovým slovem inline, tělo mimo (v hlavičce)
 - pouze hlavička metody v deklaraci třídy a tělo mimo (ve zdroji) – není inline
-
- obsahuje-li složitý kód (cykly) může být inline potlačeno (překladačem)
 - může být bráno překladačem pouze jako doporučení

.h soubor	.cpp soubor	pozn.
metoda() {}	-	inline funkce, je definováno tělo v hlavičce
metoda();	metoda:: metoda() {}	není inline. Tělo je definováno mimo hlavičku a není uvedeno inline
inline metoda();	inline metoda:: metoda(){}	je inline "z donucení" pomocí klíčového slova inline. Definice by ale měla být též v hlavičce (v cpp chyba)
metoda () {}	- neinline	nelze programátorskými prostředky zajistit aby nebyla inline, může ji však překladač přeložit jako neinline (na inline příliš složitá)
metoda ();	inline metoda:: metoda(){}	špatně (mělo by dát chybu) – mohlo by vést až ke vzniku dvou interpretací – někde inline a někde funkční volání
inline metoda()	metoda:: metoda() {}	špatně – mohlo by vést až ke vzniku dvou interpretací – někde inline a někde funkční volání; i když je vlastní kód metody pod hlavičkou takže se o inline ví, nedochází k chybě

inline metody (o) – postup při psaní programu

- podle „délky“ metody rozhodneme, zda je vhodné, aby byla inline (pokud funkce či metoda obsahuje cykly, nebo volání jiných metod, nemá smysl, aby byla inline)
- definice třídy by měla být přehledná – měla by obsahovat pokud možno jen deklarace

inline metody (o) – postup při psaní programu

- těla extrémně krátkých inline metod můžeme psát přímo do „těla“ definice třídy -> metoda je implicitně inline (bez označení)
- delší těla inline metod je výhodné kvůli přehlednosti psát mimo „tělo“ definice třídy. Před prototypem se uvede klíčové slovo inline, vlastní tělo metody se napíše za definici třídy do hlavičkového souboru
- metody s funkčním voláním (nejsou inline) mají v definici třídy pouze prototyp. Tělo metody se uvede ve zdrojovém souboru

hlavičkový soubor:

```
class CComplex{ public:  
    int Metoda1(void){return 1;} // implicitně inline  
    inline int Metoda2(); // implicitně inline díky klíčovému slovu  
    int Metoda3(); // není inline ani tělo => funkční volání  
};  
// jsme-li mimo třídu, musí být CComplex::  
int CComplex::Metoda2(){...; return 2;}
```

zdrojový soubor (include hlavička):

```
int CComplex::Metoda3(){...;...;...;return 3;}
```