

# Pretprocesiranje

- ❖ Rezultat je to da prevodilac “vidi” isti kod kao i kada su deklaracije u svim fajlovima u kojima se koriste bile pisane eksplicitno, ručno, kao i ranije, ali je razlika za programera značajna: deklaracije su lokalizovane na jednom mestu, u jednom modulu (tj. u njegovom *.h* i *.cpp* fajlu, koji moraju biti konzistentni), pa su modifikacije lakše i manje podložne greškama
- ❖ Posledično, kada se promeni neki *.h* fajl, svi *.cpp* fajlovi koji uključuju taj *.h* fajl moraju ponovo da se prevedu (ovo uključuje i tranzitivne zavisnosti od drugih uključenih *.h* fajlova), što podrazumeva da će prevodilac prevoditi i sve deklaracije u svim direktno ili indirektno uključenim *.h* fajlovima (barem da bi kerirao svoju tabelu simbola)
- ❖ Razvojno okruženje za programiranje (*integrated programming environment, IDE*) može da pomogne i smanji količinu fajlova koje treba prevoditi selektivnim prevođenjem samo onih *.cpp* fajlova koji su promenjeni, kao i onih koji zavise od promenjenih *.h* fajlova (odnosno uključuju te fajlove, uzimajući u obzir i tranzitivne zavisnosti)
- ❖ Ovakav postupak uslovnog prevođenja svih potrebnih fajlova naziva se *make* postupak (ponegde i *build* postupak), a konfiguracioni fajl koji definiše zavisnosti između fajlova *make* fajl
- ❖ Bez ozbira na tu mogućnost, programer treba da se trudi da smanji zavisnosti između fajlova, tako što ne uključuje nepotrebne *.h* fajlove u *.cpp* a posebno u druge *.h* fajlove, ali i ne unosi nepotrebne deklaracije, posebno definicije klasa u *.h* fajlove; ovo je posebno važno u velikim projektima koji mogu da imaju na stotine modula

---

# Pretprocesiranje

---

- ❖ Direktivom *#define* uvodi se nov simbol (identifikator) u pretprocesiranje (u tabelu simbola pretprocesora definisanih ovom direktivom)
- ❖ Opciono se može definisati i *makrozamena*: svaka pojava datog simbola zamenjuje se datim tekstom u nastavku direktive; mogući su i argumenti makrozamene:

```
#define _a_h
#define N 10
#define max(a,b) (((a)>=(b))?(a):(b))

int a[N];

void f (int a, int b){
    ...
    int c = max(x/3,y+1);
    ...
}
```