

---

# Enkapsulacija

---

- ❖ U nekim situacijama deo interfejsa (tj. neki članovi) jedne klase ne treba da bude dostupan svim ostalim delovima programa, bez razlike, već samo nekoj “povlašćenoj” operaciji neke druge klase ili mnogim / svim operacijama te druge klase
- ❖ U osnovi, takvi članovi morali bi biti javni, kako bi bili dostupni toj drugoj klasi, ali su oni onda dostupni i ostalim delovima programa kojima je ta klasa dostupna, bez izuzetka; na taj način može biti kompromitovana enkapsulacija posmatrane klase
- ❖ Jezik C++ pruža delimičnu podršku za rešavanje ovakvih situacija u vidu tzv. *prijateljskih funkcija* (*friend function*) ili *prijateljskih klasa* (*friend class*):
  - prijateljska funkcija  $f$  nekoj klasi  $X$  je funkcija koja nije članica te klase  $X$ , već je članica neke druge klase ili je globalna (nije članica nijedne klase), a ima pravo pristupa do privatnih i zaštićenih članova klase  $X$
  - prijateljska klasa  $F$  nekoj klasi  $X$  je klasa čije sve funkcije članice imaju pravo pristupa do privatnih i zaštićenih članova klase  $X$
- ❖ Prijateljske funkcije ili klase klasi  $X$  specifikuju se navođenjem njihovih deklaracija u bilo kom delu definicije klase  $X$  (svejedno u kom delu, tj. iza bilo kog specifikatora *public*, *protected*, *private*)

---

# Enkapsulacija

---

❖ Na primer, želimo da samo operacija *configure* klase *Configurator* može da kreira objekte klase *Controller*:

```
class Controller {  
public:  
    ...  
protected:  
    friend void Configurator::configure();  
    Controller (Reader*, Translator*);  
    ...  
private:  
    ...  
};  
void Configurator::configure () {  
    ...  
    ..new Controller(...)...  
    ...  
}
```

❖ Ako je cela klasa prijatelj nekoj drugoj klasi, onda su sve njene funkcije članice prijatelji toj drugoj klasi:

```
friend class Configurator;
```

❖ Za “prijateljstvo” važi sledeće:

- ono se ne može “preoteti”, jer ne može bilo ko da sebe proglasi prijateljem neke klase, pošto bi to narušilo enkapsulaciju te klase; “prijateljstvo” se *odobrava*: prijatelji se deklarišu u samoj klasi koja im odobrava pristup
- ono se ne nasleđuje: ako je klasa *B* prijatelj klasi *X*, a klasa *D* nasleđuje klasu *B*, klasa *D* nije implicitno prijatelj klasi *X*
- ono nije ni tranzitivna relacija: ako je klasa *B* prijatelj klasi *A*, a klasa *C* prijatelj klasi *B*, klasa *C* nije implicitno prijatelj klasi *A*