
Opseg važenja klase

- ❖ Tip definisan unutar oblasti važenja klase se koristi upravo u ovakvim situacijama, kada je on bitan samo za interfejs ili implementaciju te klase. Umesto ugnežđivanja takvog tipa, mogao bi se on deklarirati i globalno, ali to je lošije nego ugnežđivanje iz sledećih razloga:
 - ugnežđeni tipovi su “logički upakovani” u okružujuću klasu, što povećava razumljivost programa jer naglašava značaj i upotrebu takvog tipa
 - ugnežđeni tipovi pripadaju oblasti važenja klase, pa ne “prljaju” globalni prostor imena, odnosno ne uzrokuju sukob imena
 - mogu biti enkapsulirani, jer mogu biti privatni ili zaštićeni, ukoliko su bitni samo za implementaciju, a ne i za interfejs klase

Prostori imena

- ❖ Pre uvođenja koncepta *prostora imena* (*namespace*) u jezik C++, imena deklarirana izvan svih definicija klasa i tela funkcija imala su *globalni opseg važenja*, što znači da su bila dostupna od tačke deklarisanja do kraja jedinice prevođenja (fajla)
- ❖ Kada globalna imena imaju eksterno vezivanje, uzrokuju češće pojave sukoba imena (*name clash*, *name conflict*), pa je vrlo rizično upotrebljavati globalna opšta imena zbog velike šanse da se sukobe sa imenom iz nekog drugog dela programa ili biblioteke, npr. sa nazivima opštih tipova podataka (*date*, *time* itd.), struktura podataka (*list*, *stack*, *queue* itd.) i slično
- ❖ Sa ciljem bolje organizacije programa u velikim projektima, uveden je koncept *prostora imena* (*namespace*)
- ❖ Prostor imena je posebna oblasti važenja koja, slično klasi, predstavlja “logičko pakovanje” entiteta, odnosno njihovih imena koja se van tog ospega važenja mogu koristiti samo na određeni način, kvalifikovanim imenovanjem preko operatora `::`. Za razliku od klase, prostor imena nije tip, pa nema instance kao klasa; prostor imena je samo logički paket entiteta
- ❖ Prostor imena se deklarira pomoću ključne reči *namespace*, iza koje sledi niz bilo deklaracija uokvirenih u velike zagrade `{}`:

```
namespace users {  
    class User {  
        ...  
    };  
    class Role {  
        ...  
    };  
    class UserGroup {  
        ...  
    };  
}
```