

# Opseg važenja klase

- ❖ Oblast važenja imena deklarisanog unutar definicije klase je od tačke deklaracije do ostatka definicije te klase, ali se proteže i kroz tela svih funkcija članica te klase i izvedenih klasa, čak i ako su ta tela navedena van definicije klase ili pre date deklaracije
- ❖ Unutar definicije klase mogu se navoditi deklaracije ne samo objekata i funkcija, nego i tipova (*typedef* sinonimi, enumeracije, klase itd.); na primer:

```
class DatabaseManager {  
public:  
    enum DBStatus { ok, failed, refused };  
    DatabaseManager (const char* name);  
    DBStatus openConnection();  
    DBStatus closeConnection();  
    DBStatus getConnectionStatus();  
    DBStatus performQuery (const char* sqlQuery);  
  
private:  
    char* name;  
    ...  
};
```

Ova enumeracija bitna je samo za ovu klasu, odnosno za njen interfejs i implementaciju, pa je prirodno da bude logički “upakovana” u tu klasu, kao i da ima opseg važenja te klase; osim toga, nema potrebe da “prlja” globalni prostor imena i generiše potencijalni sukob imena u njemu

---

# Opseg važenja klase

---

❖ Ime iz oblasti važenja klase može se koristiti samo na jedan od sledećih načina:

- Unutar oblasti važenja iste klase ili oblasti važenja izvedenih klasa:

```
DatabaseManager::DatabaseManager(const char* nm) {  
    ...  
    name = new char[sizeof(nm)+1];  
    ...  
}
```

- Kao desni operand operatora . kom je levi operand objekat te klase ili klase izvedene iz te klase:

```
DatabaseManager* p = new DatabaseManager(...);  
...  
(*p).openConnection();
```

- Kao desni operand operatora -> kom je levi operand pokazivač na objekat te klase ili klase izvedene iz te klase:

```
p->openConnection();
```