# Premeštanje resursa

- Pomoćna operacija *swap* izgleda ovako:

```
friend void swap (string& first, string& second) {
  using std::swap;
  swap(first.str,second.str);
}
```

- Ona je deklarisana kao prijateljska funkcija koja nije članica klase *string*, kako bi bila deklarisana u prostoru imena u kom je i klasa *string*, da bi je postupak potrage po argumentu (*argument dependent lookup, ADL*) pronašao za nekvalifikovanu pretragu u tom opsegu (inače ne bi, da je statička funkcija članica); ovo je samo zato da bi ta funkcija bila dostupna i na drugim mestima za slične potrebe, recimo u apstraktnim strukturama podataka koje treba da rade iste stvari (zamenu vrednosti npr. u svojim *swap* funkcijama koje koriste za premeštanje)
- Treba primetiti da

```
using std::swap;
swap(first.str,second.str);
```

  u opštem slučaju ne znači isto što i:

```
std::swap(first.str,second.str);
```

- Prva varijanta dozvoljava da postupak ADL za argumente poziva funkcije *swap* nađe i funkciju koja bolje odgovara tipovima argumenata a nije u prostoru imena *std*, gde je najpre i traži, ali da ako takvu ne nađe, drugu traži i u prostoru imena *std*. Druga varijanta zahteva da se takva funkcija strogo traži samo u prostoru imena *std*
- Za konkretan slučaj, pošto se radi o prostim pokazivačima tipa *char**, rezultat će biti isti, ali u slučaju drugih, npr. korisničkih tipova, rezultat može biti različit

# Premeštanje resursa

- Ostaje još i da se definiše konstruktor premeštanja, koji može da bude pozivan za argumente koji su dvrednosti. On vrši jednostavnu zamenu operacijom *swap*, pri čemu je najpre svoj pokazivač *str* postavio na vrednost *null* podrazumevanom konstrukcijom:

```
inline string::string (string&& other) : string() {
  swap(*this,other);
}
```

- Privremeni objekat kojim se inicijalizuje objekat za koji se konstruktor izvršava neće imati ništa da uništava, jer će u njegov pokazivač *str* ovom zamenom biti upisana *null* vrednost
- Kako sve operacije *swap* takve da ne bacaju izuzetke (*non-throwing*, *noexcept*), ovakva implementacija obezbediće jaku garanciju od izuzetaka; osim toga, one su veoma efikasne, jer vrše prostu i brzu razmenu vrednosti