

# Premeštanje resursa

- ❖ Opisani problem predstavlja izvornu motivaciju za uvođenje tzv. *semantike premeštanja* (*move semantics*) u jezik C++ (počev od verzije C++11): umesto da se alocirani resursi pridruženi privremenom objektu kopiraju, oni mu se mogu *preoteti*, tj. *premestiti* u objekat koji se tim privremenim objektom inicijalizuje ili kom se taj privremeni objekat dodeljuje, pošto privremenom objektu svakako više nisu potrebni
- ❖ Tako je (od verzije C++11) moguće definisati konstruktore i operatore dodele, pa i one koji kao argument mogu da prime objekat istog tipa, koji će se pozivati u slučaju da se kao argument pojavi *dvrednost* (*rvalue*), odnosno rezultat izraza koji nije *lvrednost* (*lvalue*)
- ❖ Konstruktor koji prima referencu na dvrednost iste klase naziva se *konstruktor premeštanja* (*move constructor*); nestatička operatorska funkcija članica `operator=` koja prima ovakvu referencu na dvrednost iste klase naziva se *operator dodele premeštanjem* (*move assignment*)
- ❖ Parametri koji se vezuju za dvrednosti su tzv. *reference na dvrednosti* (*rvalue reference*) i označavaju se u deklaraciji sa dva znaka `&`:

```
class string {  
public:  
    ...  
    string (const string& s);  
    string& operator= (const string& s);  
  
    string (string&& s);  
    string& operator= (string&& s);  
    ...  
};
```

Konstruktor kopije

Operator dodele kopiranjem

Konstruktor premeštanja

Operator dodele premeštanjem

---

# Premeštanje resursa

---

- ❖ Konstruktor premeštanja poziva se kada se objekat inicijalizuje izrazom koji predstavlja dvrednost, a takav je privremeni objekat, pod uslovom da prevodilac ne izostavlja kopiranje:

```
string s = s1 + s2;
```

```
void f (string);
```

```
f(s1+s2);
```

- ❖ Osim toga, kao specijalan slučaj, konstruktor premeštanja se poziva i kada se povratna vrednost funkcije inicijalizuje imenovanim objektom sa automatskim trajanjem skladišta, ovaj put uključujući i formalni parametar, opet osim ako prevodilac ne vrši NRVO:

```
inline string substr (size_t pos, size_t count) const {  
    ...  
    string s;  
    ...  
    return s;  
}
```

- ❖ Ukoliko klasa nema konstruktor premeštanja, u ovakvim slučajevima pozivaće se konstruktor kopije, pa postojanje konstruktora premeštanja ne treba da menja semantiku programa: konstruktor premeštanja može da popravi performanse i smanji količinu kopiranja resursa, ali pošto njegova aktivacija nije uvek određena (u slučajevima opcionih izbegavanja kopiranja), logika programa ne sme da zavisi od njegovog ponašanja
- ❖ Slično važi i za operator dodele premeštanjem: poziva se kada je desni operand dvrednost