

# Premeštanje resursa

- ❖ Konstruktor premeštanja i operator dodele premeštanjem treba da preotmu, tj. premeste resurse iz izvorišnog u odredišni objekat, ali tako da izvorišni objekat ostave u konzistentnom stanju, tako da on i dalje bude validan objekat koji se svakako može uništiti (jer se on svakako uništava pozivom destruktora); zato ove operacije po pravilu imaju parametar koji je referenca na nekonstantu
- ❖ Za primer klase *string*, to znači da se objekat član *str* u izvorišnom objektu postavi na *null* vrednost, a ne ostavi na staroj vrednosti, jer preuzeti dinamički niz znakova više nije njegov; u svakom slučaju, njegov destruktork ne sme uništiti taj niz jer on pripada drugom objektu (premešten je):

```
class string {
public:
    string () : str(nullptr) {}
    string (const char* s) : string() { allocate(s); copy(s); }

    string (const string& s) : string(s.str) {}
    string& operator= (const string& s);

    string (string&& s) : string() { move(s); }
    string& operator= (string&& s);

    ~string () { release(); }
    ...
protected:
    void allocate (const char* s) { if (s) str = new char[std::strlen(s)+1]; }
    void copy (const char* s) { if (s) std::strcpy(str,s); }
    void release () { delete [] str; str = nullptr; }

    void move (string&& s) { str = s.str; s.str = nullptr; }
    ...
};

inline string& string::operator= (string&& s) {
    if (this!=&s) {
        release(); move(s);
    }
    return *this;
}
```

# Premeštanje resursa

- ❖ Vratimo se na implementaciju operatora dodele kopiranjem:

```
inline string& string::operator= (const string& s) {  
    if (this != &s) {  
        release();  
        allocate(s.str);  
        copy(s.str);  
    }  
    return *this;  
}
```

- ❖ Ova implementacija ima sledeća dva nedostatka:

- provera na “samododeljivanje” (*self assignment*) u naredbi *if* izvršava se uvek i troši režijsko vreme u svakom pozivu, iako se verovatno nikada u programu neće dogoditi problematična situacija samododeljivanja (jer nema smisla)
- ova implementacija pruža samo osnovni (slabi) nivo garancije od izuzetaka: ako alokacija prostora u funkciji *allocate* ne uspe i baci izuzetak, objekat domaćin (na koga ukazuje *this*) već je izmenjen u operaciji *release*; bilo bi bolje da se obezbedi jaka garancija