## Premeštanje resursa

\* Vratimo se na implementaciju funkcije *substr*:

```
inline string string::substr (size_t pos, size_t count) const {
    size_t sz = size();
    if (pos>=sz) throw std::out_of_range;
    if (pos+count>sz) count = sz - pos;
    string s;
    s.allocate(count);
    s.copy(str+pos,count);
    return s;
```

Zadatak: implementirati funkciju *string::substr(...)&&* 

Zadatak: implementirati istu funkciju ali tako da vraća referencu na dvrednost

- \* Ukoliko se ova funkcija pozove za privremeni objekat kao objekat domaćin (\*this), bespotrebno će alocirati prostor za vraćeni objekat tipa string, ako već alocirani niz znakova u tom objektu ubrzo nestaje; umesto toga, za traženi podniz može se iskoristiti isti taj prostor koji se može preoteti od privremenog objekta domaćina
- \* Kako se radi o nestatičkoj funkciji članici, potrebno je razlikovati funkcije koje će biti pozvane u slučajevima da je objekat domaćin privremeni objekat ili nije, odnosno da je dvrednost ili lvrednost; to se može uraditi navođenjem znakova & (za lvrednost) ili && (za dvrednost) u deklaraciji nestatičke funkcije članice:

```
class string {
    ...
    string substr (size_t pos, size_t count) const &;
    string substr (size_t pos, size_t count) &&;
    string substr (size_t pos, size_t count) &&;
    ...
};

string s1 = "Hello world!";

string s2 = s1.substr(0,5);

string s3 = string("Hello world!").substr(0,5) + (s1+s2).substr(0,5);
Poziva se string::substr(0,5);
```

## Kategorije vrednosti

- \* Svaki izraz (rezultat operatora, literal ili ime varijable) ima dva svojstva: tip (type) i kategoriju vrednosti (value category)
- \* Dakle, vrednost (value) je izraz i svaki izraz jeste vrednost koja pripada jednoj od kategorija vrednosti
- \* Ideja za razvrstavanje na kategorije potiče od sledećih svojstava entiteta kog vrednost predstavlja:
  - entitet se može (ili ne može) *identifikovati* (*identifiable*): može se odrediti da li se jedan izraz odnosi na isti entitet kao drugi izraz, recimo poređenjem adresa objekata ili funkcija koje ti izrazi identifikuju
  - entitet se može (ili ne može) *premeštati (movable*): konstruktor premeštanja, operator dodele premeštanjem i druge preklopljene funkcije koje implementiraju premeštanje mogu da se vežu za takav izraz
- \* Izrazi mogu biti jedne od sledećih disjunktnih kategorija:
  - ako ima identitet i ne može se premeštati, ima kategoriju lvrednosti (lvalue, od left value)
  - ako ima identitet i može se premeštati, ima kategoriju *xvrednosti* (*xvalue*, nema posebne motivacije za naziv, mada se često asocira sa *expiring value*)
  - ako nema identitet i može se premeštati, ima kategoriju čdvrednosti (prvalue, od pure rvalue, "čista dvrednost")
  - izrazi koji nemaju identitet i ne mogu se premeštati se ne koriste (ne postoje jer nemaju smisla)
- \* Izraz koji ima identitet, bez obzira na to da li se može ili ne može premeštati, dakle koji je lvrednost ili xvrednost, naziva se glvrednost (glvalue, od generalized lvalue)
- \* Izraz koji se može premeštati, bez obzira na to da li ima ili nema identitet, dakle koji je čdvrednost ili xvrednost, naziva se *dvrednost* (*rvalue*, od *right lvalue*)