

Kopiranje objekata

- ❖ Pretpostavimo da smo implementirali operatorsku funkciju *operator+* koja vraća rezultat spajanja dva niza znakova iz operanada tipa *string*; taj rezultat mora biti objekat tipa *string* (po vrednosti), jer je on novonapravljeni objekat koji sadrži poseban dinamički niz znakova:

```
string operator+ (const string&, const string&);
```

- ❖ Osim toga, implementirali smo i nestatičku funkciju članicu *substr* koja vraća (ponovo po vrednosti) nov objekat tipa *string* koji sadrži samo podniz datog niza znakova objekta domaćina, počev od zadate pozicije i zadate dužine; za potrebe implementacije ove funkcije, napravili smo još nekoliko funkcija članica klase *string*:

```
inline void allocate (size_t sz) {  
    if (sz+1==0) throw std::length_error;  
    if (sz) str = new char[sz+1];  
}
```

Pomoćna, zaštićena funkcija članica klase *string*

```
inline void copy (const char* s, std::size_t count) {  
    if (!s || !count) return;  
    size_t i = 0;  
    while (i<count && *s) str[i++] = *s++;  
    str[i] = '\\0';  
}
```

Pomoćna, zaštićena funkcija članica klase *string*

```
inline size_t size () const { return str?std::strlen(str):0; }
```

Javna funkcija članica klase *string*

```
inline 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;  
}
```

Javna funkcija članica klase *string*

Kopiranje objekata

- ❖ Pre verzije jezika C++17, svaki rezultat izraza klasnog tipa, uključujući poziv funkcije koja ima povratni tip klase i eksplicitan poziv konstruktora klase, kao rezultat proizvodi privremeni, bezimeni objekat koji se pravi na mestu poziva funkcije, tokom izvršavanja izraza koji sadrži taj poziv i inicijalizuje u trenutku povratka iz te funkcije izrazom iza naredbe *return*

- ❖ Na primer, osnovna semantika (pre verzije C++17) sledeće inicijalizacije

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

je sledeća: rezultat izraza *s1+s2* je privremeni objekat klase *string* koji je inicijalizovan prilikom povratka iz funkcije odgovarajućim konstruktorom; nakon povratka, objekat *s* klase *string* inicijalizuje se pozivom konstruktora kopije sa tim privremenim objektom kao argumentom poziva konstruktora; naravno, ti konstruktori moraju da budu dostupni na odgovarajućim mestima

- ❖ Potpuno ista situacija je i u sledećem slučaju:

```
string s = string("Hello");
```

- ❖ U određenim situacijama, prevodiocu je na raspolaganju optimizacija *izostavljanja kopiranja* (*copy elision*) koju može, ali ne mora da izvrši (pre verzije C++17), čak i kada konstruktori čiji se pozivi izostavljaju imaju bočne efekte