

# Kopiranje objekata

❖ Posmatrajmo sada izračunavanje sledećeg izraza, pri čemu su sve imenovane varijable objekti tipa *string*:

```
s = s1 + " " + s2.substr(2,7) + " " + s3.substr(0,2);
```

❖ Izračunavanje ovog izraza teče ovako (pretpostavka je da je verzija jezika pre C++17):

- drugi argument poziva *operator+(s1, " ")* je tipa *const char[]*, pa se implicitno konvertuje u *const char\** ugrađenom konverzijom ("rastakanje niza u pokazivač"), a potom i korisnički definisanom konverzijom u *string*, pozivom konstruktora konverzije *string(const char\*)*; rezultat konverzije je privremeni objekat kojim se inicijalizuje formalni parametar; pošto je to referenca, ona se vezuje za taj privremeni objekat
- rezultat ovog poziva biće jedan privremeni objekat tipa *string*, označimo ga ovde sa *x*, koji je inicijalizovan povratnim izrazom; taj privremeni objekat ima svoj pridruženi dinamički niz znakova koji sadrži konkatenciju niza znakova iz *s1* i niza znakova " "; ukoliko prevodilac ne sprovodi *RVO*, ovaj privremeni objekat biće inicijalizovan pozivom konstruktora kopije, ako je operand izvršene naredbe *return* tipa *string*
- unutar pozvane funkcije *s2.substr(2,7)* pravi se automatski objekat *s* koji alokira prostor za traženi podstring; ovim automatskim objektom se inicijalizuje privremeni objekat koji predstavlja rezultat ovog poziva, označimo ga ovde za *y*; ukoliko prevodilac ne sprovodi *NRVO*, ponovo će se pozivati konstruktor kopije
- poziva se sada operatorska funkcija za drugi operator *+* u izrazu; parametri ove funkcije su reference koje se vezuju za navedene privremene objekte, tj. poziva se *operator+(x,y)*

i tako dalje. Na kraju, poziva se operatorska funkcija *operator=* čiji je desni operand privremeni objekat napravljen kao rezultat podizraza sa desne strane znaka *=*

---

# Kopiranje objekata

---

- ❖ Prema tome, u ovakvim i sličnim izrazima, vrši se potencijalno mnogo poziva konstruktora kopije ili operatora dodele kopiranjem koji kopiraju sadržaj, tj. alociraju nove pridružene resurse samo da bi kopirali taj isti sadržaj iz privremenih objekata, koji ubrzo potom nestaju
- ❖ Kopiranje resursa iz privremenog objekta je prilično beskoristan posao i čist režijski trošak za vreme izvršavanja, pošto privremeni objekat iz kog se pridruženi sadržaj kopira (uz potencijalnu alokaciju novog resursa) ubrzo nestaje, i *sigurno se ne može* koristiti za bilo šta drugo, jer je on nedostupan - on je svakako implicitan rezultat izraza koji se koristi eventualno samo kao operand (argument) naredne operacije i ni za šta više
- ❖ Količina kopiranja je utoliko veća ukoliko:
  - prevodilac ne sprovodi optimizaciju izostavljanja kopiranja na svim mestima na kojima je to moguće i dozvoljeno pravilima jezika
  - se parametri u funkcije prenose po vrednosti, a ne po referenci
  - funkcije vraćaju rezultate kopiranjem automatskih objekata
  - se objekti ugrađuju u druge objekte po vrednosti (a ne preko pokazivača), posebno u apstraktne strukture podataka (kolekcije, kontejnere), što je često slučaj, pa se umetanjem u strukturu i vraćanjem vrednosti iz strukture objekti kopiraju (operatorima dodele ili konstruktorima kopije)