Konstantni tipovi i funkcije članice

- Deklarisanjem pokazivača na konstantan objekat programer najavljuje ("obećava") da ono na šta taj pokazivač ukazuje ne može da se menja *preko tog* pokazivača, što ne znači da je apsolutno konstantno; prevodilac kontroliše ispunjenje te najave dosledno, sprovođenjem sledećih pravila jezika:
  - postoji implicitna konverzija iz tipa pokazivača na nekonstantan tip *T* u tip pokazivača na konstantan tip *T*: time se samo "zateže" konstantnost, odnosno obećava da se preko nekog drugog pokazivača neće izmeniti objekat, u kontekstu (opsegu važenja) tog pokazivača:

```
char* p = ...;
const char* q = p;
...
q = p;
```

Dozvoljeno: samo znači da se objekat na koga ukazuje *q*, koji god da je, neće menjati preko tog *q*, ne znači da je on i generalno konstantan

  - nije dozvoljena implicitna konverzija iz tipa pokazivača na konstantan tip *T* u tip pokazivača na nekonstantan tip *T*, jer bi se time "tiho probila" konstantnost, odnosno omogućilo slučajno narušavanje konstantnosti, bez upozorenja:

```
const char* p = ...;
char* q = p;
...
q = p;
```

Greška u prevođenju, jer bi inače, nakon ovoga, bilo moguće promeniti objekat **q*

  - dozvoljena je eksplicitna konverzija operatorom *const_cast* iz tipa pokazivača na konstantan tip *T* u tip pokazivača na nekonstantan tip *T*; izmena konstantnog objekta preko takvog pokazivača ima nedefinisane efekte:

```
const char* p = ...;
char* q = const_cast<char*>p;
```

Sada je moguće promeniti objekat **q*, ali je efekat toga nedefinisan

Konstantni tipovi i funkcije članice

- String-literali imaju tip *const char[]* (niz konstantnih znakova), pa su u skladu sa tim dozvoljene njihove implicitne konverzije u *const char**, ali ne i u *char**:

```
const char* p = "Hello";
char* q = "World";
```

- Naravno, sve to važi i ukoliko su parametri funkcije pokazivači na konstantan tip: tada funkcija "obećava" da neće izmeniti ono na šta taj parametar ukazuje (jer je opseg važenja parametra lokalna za tu funkciju), pa se funkcija može pozvati i sa pokazivačem na konstantan i sa pokazivačem na nekonstantan objekat (u suprotnom može samo za nekonstantan):

```
void f1 (const char*);
void f2 (char*);
char s1 = …;
const char s2 = …;

void f () {
  f1(&s1);
  f2(&s1);
  f1(&s2);
  f2(&s2);
}
```

- Sva navedena pravila konverzija važe potpuno isto i za reference na konstantne/nekonstante tipove