Konstantni tipovi i funkcije članice

- Kvalifikator *const* može da se piše i ispred i iza naziva tipa:

```
const char* pc1 = ...;
char const* pc2 = ...;
```

- Pokazivač na konstantan objekat definiše se stavljanjem reči *const* ispred (ili iza) tipa objekta (ali ispred znaka *); konstantan pokazivač definiše se stavljanjem reči *const* ispred samog imena pokazivača, odnosno iza znaka *:

```
const char* pc = ...;
pc[3] = 'a';
pc = ...;

char* const cp = ...;
cp[3] = 'a';
cp = ...;

const char* const cpc = ...
cpc[3] = 'a';
cpc = ...;
```

Greška u prevođenju: *pc[3]* je isto što i **(pc+3)*; kako je *pc* tipa *const char** (pokazivač na konstantan *char*), to je **(pc+3)* tipa *const char*, pa se ne može menjati

Greška u prevođenju: *cp* je tipa *char*const* (konstantan pokazivač na nekonstantan *char*), pa se ne može menjati; *cp[3]* je tipa *char*, pa se može menjati

Greška u prevođenju: *cpc* je tipa *const char*const* (konstantan pokazivač na konstantan *char*), pa se ne može menjati ni on, ni ono na šta on ukazuje

- Referenca može biti na konstantan tip, ali sama referenca ne može da se deklariše kao konstantna, jer ona to svakako jeste (ne postoji operacija koja bi promenila referencu nakon inicijalizacije); ako se *const* upotrebi uz referencu indirektno, npt. u *typedef* deklaracijama, ignoriše se

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;
```

  - 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;
```

  - 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;
```