Konstantni tipovi i funkcije članice

- Svaki objektni tip (tj. tip koji nije referenca ili funkcija) može da bude kvalifikovan kao *konstantan* (*constant*). Objekti ovakvog tipa ne mogu se menjati: neposredan pokušaj operacije nad ovakvim objektom koja bi promenila taj objekat prevodilac prijavljuje kao grešku, dok indirektan pokušaj izmene takvog objekta, npr. preko pokazivača ili reference na nekonstantan objekat rezultuje nedefinisanim ponašanjem (može izazvati čak i izuzetak od hardvera ako je konstantan objekat smešten u deo memorije zabranjen za upis)
- Konstantni objekti moraju biti inicijalizovani u definiciji, jer kasnije svakako ne mogu da se promene. Na primer:

```
const float pi = 3.14;
const char plus = '+';
pi++;
plus = '-';
```

Greške u prevođenju

- Konstantni objekti fundamentalnih tipova mogu da se koriste u konstantnim izrazima koje prevodilac treba da izračuna tokom prevođenja, na primer, u izrazima koje definišu dimenzije nizova:

```
const int MaxNumOfClocks = 100;
…

Clock* clocks[MaxNumOfClocks];
```

- Umesto korišćenja makro zamene *#define*, bolje je koristiti konstantne objekte, jer oni imaju svoj tip koji može biti različit od tipa literala kojim se makro u direktivi *#define* zamenjuje; u određenim situacijama, prevodilac ne mora ni da alocira konstantan objekat za vreme izvršavanja, već da njegovu konstantnu vrednost potpuno iskoristi za vreme prevođenja na svim mestima njenog korišćenja

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 = ...;
```

- 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