

Nizovi

- ❖ *Niz (array)* je objekat koji predstavlja uređenu, ograničenu kolekciju objekata, složenih u memoriju jedan iza drugog, deklarisanе veličine
- ❖ Elementi niza mogu biti bilo kog objektnog tipa: fundamentalnog tipa (osim tipa *void*), pokazivačkog tipa, klasnog tipa, nabiranja, ili niza (tako se pravi višedimenzionalni nizovi). Ne postoje nizovi referenci ili funkcija
- ❖ Elementi niza veličine N označeni su indeksima (pozicijama u nizu) počev od 0 zaključno sa $N-1$ i može im se pristupati operatorom *indeksiranja niza* []:

T	T	T	T	T	T	T	T
---	---	---	---	---	---	---	---

```
int a[100];
```

a je niz od 100 elemenata tipa *int*

```
...  
a[0] = a[0] + a[99];
```

Pristup prvom i poslednjem elementu niza *a*

- ❖ Elementi niza mogu biti drugi nizovi — tako se prave multidimenzionalni nizovi:

```
int m[5][7];
```

m niz od 5 elemenata, gde je svaki element tipa niza od 3 elementa tipa *int* — matrica 5x7

```
m[3][5] = 2;
```

Pristupa se 4. elementu niza *m*; on je opet niz; pristupa se 6. elementu tog niza

- ❖ Za vreme izvršavanja, ne proverava se ispravnost indeksa za pristup nizu u smislu da li su u opsegu dozvoljenih vrednosti (unutar granica niza); svaki prestup može da dovede do nekorektnog ponašanja, kao sa pokazivačima

Nizovi

- ❖ Niz može da se inicijalizuje tzv. *agregatnim inicijalizatorom* (*aggregate initializer*) između velikih zagrada; tada ne mora da se navede dimenzija niza, ona se određuje na osnovu broja elemenata u inicijalizatoru i smatra se definisanom:

```
int a[3] = {1, 2, 3};
```

```
int b[] = {4, 5, 6};
```

```
int c[3] = {1};
```

- ❖ Niz može da se deklarise i bez zadavanja (prve) dimenzije (ako je u pitanju niz nizova, sve ostale dimenzije moraju da se zadaju, jer one definišu tip elementa niza); ovaj niz je nekompletnog tipa (osim ako je inicijalizovan agregatnim inicijalizatorom). Za ovakve nizove mogu se praviti pokazivači i reference na njih, ali se ti pokazivači i reference ne mogu inicijalizovati tako da ukazuju na nizove sa poznatim dimenzijama. Takvi pokazivači se mogu dereferencirati i prenositi kao parametri funkcija, ali ne mogu učestvovati u pokazivačkoj aritmetici:

```
extern int a[];
```

```
int (*p)[] = &a;
```

```
int (*q)[2] = &a;
```

- ❖ Nizovi ne mogu da se kopiraju operacijom dodele, tj. ne mogu biti levi operandi operatora dodele. Međutim, ako je niz podatak član nekog klasnog tipa (strukture ili klase), on se može kopirati kao deo objekta kome pripada:

```
int a[10], b[10];
```

```
class X { public: int a[10]; };
```

```
X x1, x2;
```

```
a = b;
```

```
x1 = x2;
```