

Programiranje II

Statički nizovi



Nizovi – polja

- U ovom materijalu ponavljamo gradivo obrađeno u okviru predmeta Programiranje I, a odnosi se na područje nizova ili polja
- Niz je „linearna“ struktura vrijednosti (objekata) istog tipa
- Objekti od kojih se niz sastoji nazivaju se elementi niza
- Elementi niza su numerisani redom 0, 1, 2, 3, ..., n
- Pomenuti brojevi se nazivaju indeksi i preko njih se može direktno pristupiti bilo kojem elementu niza

niz	
0	11.11
1	33.33
2	55.55
3	77.77
4	99.99



Nizovi – polja

- Ako je **niz** ime niza od **n** elemenata, tada se pojedinim elementima može pristupiti na sljedeći način: `niz[0]`, `niz[1]`, `niz[2]`, ..., `niz[n-1]`
- Indeks ukazuje koliko je neki element udaljen od početnog elementa kome je dodijeljen indeks 0
- Nizovi se koriste pri radu s više podataka istog tipa koji obično čine neku cjelinu
- U memoriji niz zauzima kontinuirani prostor

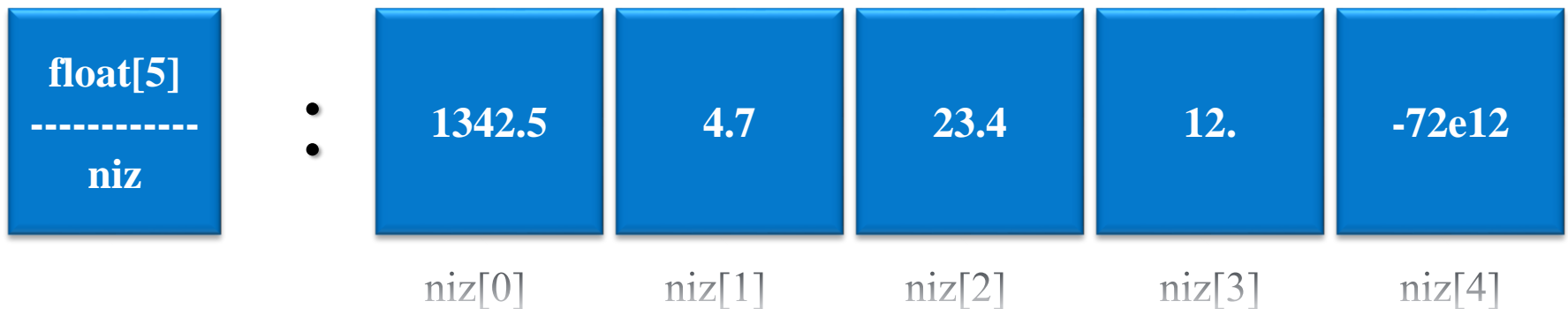
niz	
0	11.11
1	33.33
2	55.55
3	77.77
4	99.99



Nizovi – polja

- U nastavku je prikazan primjer inicijalizacije niza pri samoj deklaraciji

```
float niz[]={1342.5, 4.7, 23.4, 12., -72e12};
```





Osnove korištenja nizova

- Deklaracija niza:

```
tip_podatka ime_niza[dimenzija];  
double niz[5];
```

- Dodjela vrijednosti elementima niza:

```
niz[2]=55.55;  
niz[0]=11.11;  
niz[1]=33.33;
```

- Ispis elemenata niza:

```
cout<<"niz[0]="<<niz[0]<<endl;  
cout<<"niz[1]="<<niz[1]<<endl;  
cout<<"niz[2]="<<niz[2]<<endl;
```

niz	
0	11.11
1	33.33
2	55.55
3	77.77
4	99.99

Učitavnje i ispis elemenata niza



```
#include <iostream>
using namespace std;
```

```
void main() {
    const int max = 5;
    double niz[max];

    cout << "Upisite " << max << " brojeva: ";
    for (int i=0; i<max; i++)
        cin >> niz[i];

    cout << "Upisali ste sljedece brojeve: ";
    for (int i=0; i<max; i++)
        cout << niz[i] << "\t";
}
```

niz

0	11.11
1	33.33
2	55.55
3	77.77
4	99.99

Upisite 5 brojeva: 11.11 33.33 55.55 77.77 99.99

Upisali ste sljedece brojeve: 11.11 33.33 55.55 77.77 99.99



Inicijalizacija niza prilikom deklaracije

- Slično kao i sa skalarnim varijablama, niz je moguće inicijalizovati prilikom njegove deklaracije, npr:

```
int niz[4] = {22, 33, 66, 55};  
char slova[3] = {'a', 'b', 'c'};
```

- Ako je broj elemenata u listi manji od veličine niza, preostali elementi se inicijalizuju vrijednošću nula (0), npr :

```
float nizF[5] = {2.5, 1.2};  
//Prva dva elementa niza nizF sadrže vrijednost 2.5 i  
//1.2, a ostali vrijednost 0
```

- Veličinu niza je moguće izostaviti, npr :

```
int niz[] = {55, 33, 66};
```

- Veličina niza **niz** je u ovom slučaju 3 tj. jednaka je broju elemenata liste, što je moguće izračunati formulom:

```
int velicina_niza = sizeof(niz)/sizeof(int);
```

- Ako polje nije inicijalizovano, sadržaj elemenata prije dodjele vrijednosti je sasvim slučajan!

Prosječna vrijednost niza



```
#include <iostream>
using namespace std;
void main() {
    int suma=0;
    float prosjek = 0;
    const int max = 6;
    int ocjene[max] = {6, 8, 9, 7, 10, 9};
    cout<<"Student je dobio sljedece ocjene: ";
    for (int i=0; i<max; i++) {
        cout <<ocjene[i] <<" ";
        suma += ocjene[i];
    }
    prosjek=float(suma)/max;
    cout <<"\nProsjecna ocjena je: " <<prosjek<<endl;
    system("pause");
}
```

```
Student je dobio sljedece ocjene: 6 8 9 7 10 9
Prosjecna ocjena je: 8.16667
```


Niz kao parametar funkcije



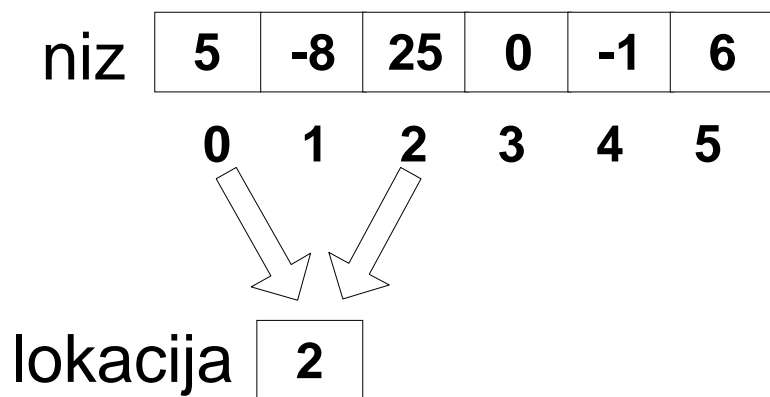
```
#include <iostream>
using namespace std;
void ispisiNiz(int niz[], int max) {
    for(int i=0; i<max; i++)
        cout<<"niz["<<i<<"]="<<niz[i]<<endl;
}
void ucitajNiz(int niz[], int max) {
    for(int i=0; i<max; i++) {
        cout<<"niz["<<i<<"]=" ";
        cin>>niz[i];
    }
}
void main() {
    const int max = 3;
    int niz[max];
    ucitajNiz(niz,max);
    ispisiNiz(niz,max);
    system("pause");
}
```

Napomena: nizovi se uvijek predaju po referenci

Pronalaženje najvećeg elementa niza



```
//funkcija koja vraca lokaciju najveceg clana niza
int NajveciClan(int niz[],int max){
    int lokacija = 0;
    for(int i=1;i<max;i++)
        if(niz[lokacija]<niz[i])
            lokacija = i;
    return lokacija;
}
```



Komentar: Pretpostavimo da je prvi element niza najveći, te zapamtimo njegov indeks (0). Zatim provjeravamo da li među preostalim elementima postoji neki element čija je vrijednost veća od vrijednosti trenutno najvećeg elementa. Ako takav element postoji, zapamtimo njegov indeks kao lokaciju elementa sa najvećom vrijednošću. Na kraju posjedujemo indeks elementa (lokaciju) sa najvećom vrijednošću i tu vrijednost funkcija vraća.

Broj dana od pocetka godine...



```
#include <iostream>
using namespace std;
void main() {
    int Datum[3]; //15 3 2013
    int Mjesec[]={31,28,31,30,31,30,31,31,30,31,30,31};
    cout<<"Unosenje datuma (dan,mjesec,godina): ";
    for(int i=0;i<3;i++)
        cin>>Datum[i];
    cout<<endl;
    //prestupna godina-osnovna provjera?
    if (Datum[2]%4==0)
        Mjesec[1]=29;
    int Suma=Datum[0];
    for(int i=0;i<Datum[1]-1;i++)
        Suma=Suma+Mjesec[i];
    cout<<"Od 01.01."<<Datum[2]<<" do "<<Datum[0]<<"." <<Datum[1];
    cout<<"."<<Datum[2]<<"."<<" proslo je "<<Suma<<" dana."<<endl;
}
```

```
Unosenje datuma (dan,mjesec,godina): 15 3 2013
Od 01.01.2013 do 15.3.2013. proslo je 74 dana.
```

Rekurzija i nizovi



```
#include <iostream>
using namespace std;

int sumaRekurzivno(int niz[], int max) {
    if (max < 0)
        return 0;
    return niz[max] + sumaRekurzivno(niz, max-1);
}

void main() {
    const int max = 5;
    int niz[max] = {1, 2, 3, 4, 5};
    cout << "Suma: " << sumaRekurzivno(niz, max-1);
    cout << endl;
    system("pause");
}
```

```
Suma: 15
Press any key to continue . . .
```



Dvodimenzionalni nizovi

```
int niz[3][5] = {{11,12,13,14,15}, {21,22,23,24,25} };
```

<div>int [3] [5]</div> <div>niz</div>	:	11	12	13	14	15
		niz[0][0]:	niz[0][1]:	niz[0][2]:	niz[0][3]:	niz[0][4]:
		21	22	23	24	25
		niz[1][0]:	niz[1][1]:	niz[1][2]:	niz[1][3]:	niz[1][4]:
		0	0	0	0	0
		niz[2][0]:	niz[2][1]:	niz[2][2]:	niz[2][3]:	niz[2][4]:

Dvodimenzionalni nizovi i rekurzija



```
#include <iostream>
using namespace std;
//dimenzije matrice
const int redova = 3;
const int kolona = 3;
int sumaRekurzivno(int matrica[][kolona],int red, int kol){
    if(red<0)
        return 0;
    int broj = matrica[red][kol];
    if(kol==0)
        return broj + sumaRekurzivno(matrica,red-1,kolona-1);
    return broj + sumaRekurzivno(matrica,red,kol-1);
}
void main(){
    int matrica [redova][kolona]={{1,2,3},{4,5,6},{7,8,9}};
    cout<<"Suma: "<<sumaRekurzivno(matrica,redova -1,kolona-1);
    cout<<endl;
    system("pause");
}
```

Suma: 45

Press any key to continue . . .



Višedimenzionalni nizovi

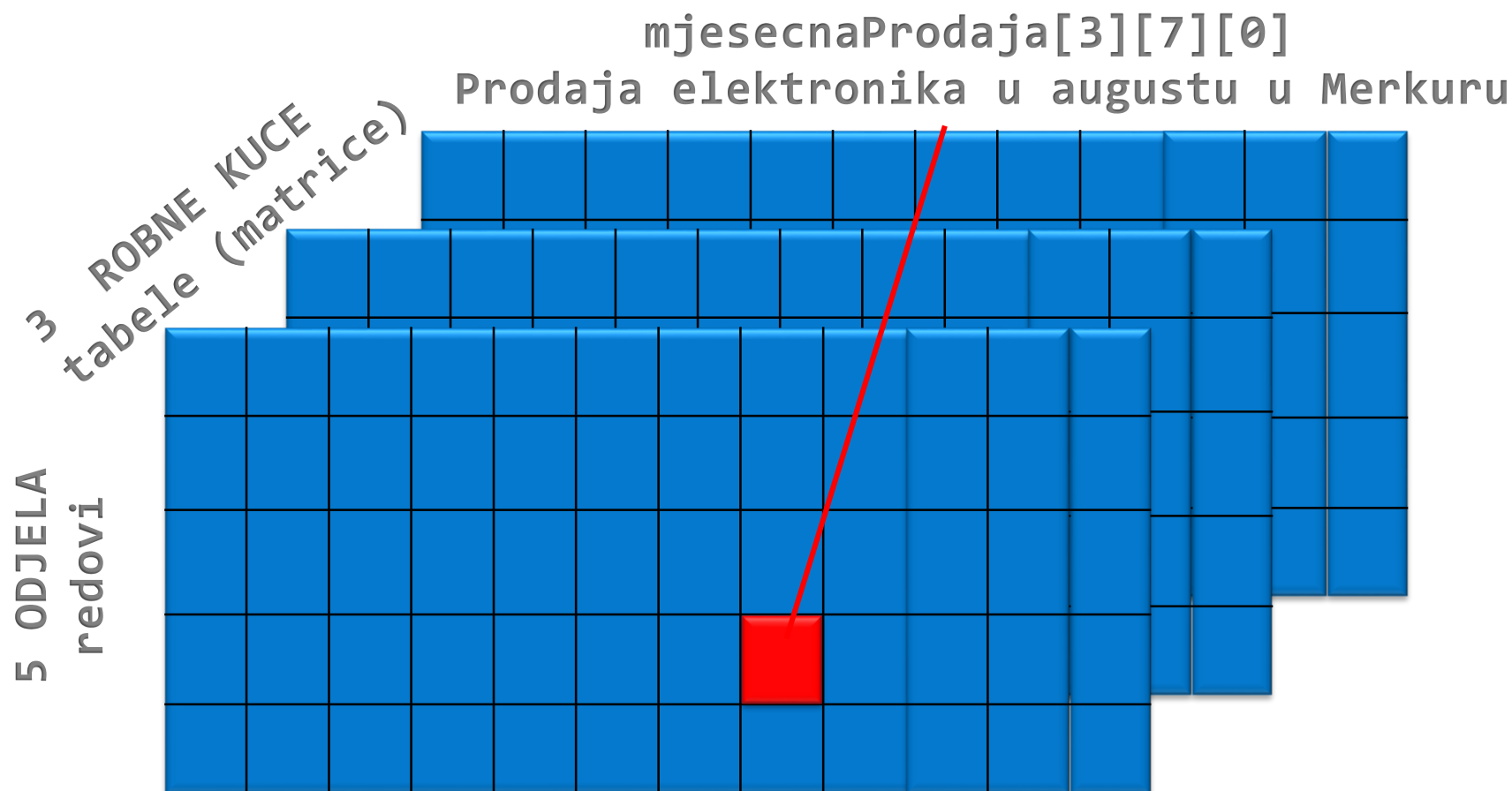
- Pored uobičajene dvije dimenzije, nizovi mogu imati i veći broj dimenzija. U nastavku je prikazan primjer kreiranja trodimenzionalnog niza.

```
const int Br_ODJELA = 5;
// muski, zenski, djecji, elektronika, namjestaj
const int Br_MJESECI = 12;
const int Br_ROBNIH_K = 3;
// Merkator, Robot, Merkur

int mjesecnaProdaja[Br_ODJELA][Br_MJESECI][Br_ROBNIH_K];
//redovi kolone tabele
```



Višedimenzionalni nizovi



- Za vježbu, prethodnom primjeru dodajte i četvrtu dimenziju koja će označavati godinu evidentiranja prodaje



KRAJ PREZENTACIJE