LabVIEW

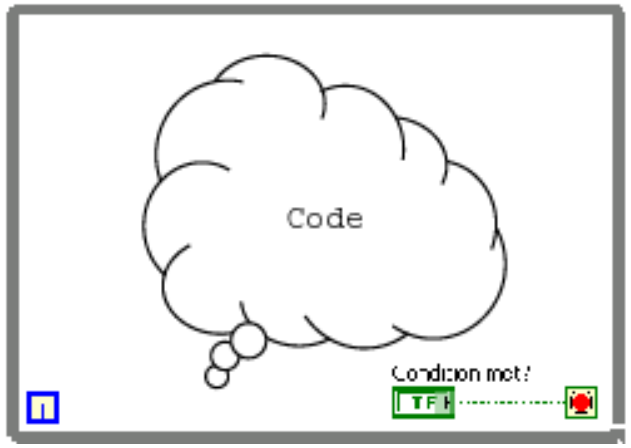
Dr Marko Dimitrijević

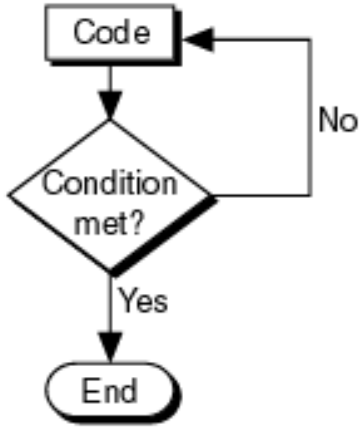
Programske petlje

Programske petlje

- WHILE petlja
- FOR petlja
- Uslovna FOR petlja
- Vremenske funkcije kašnjenja
- Timed loop petlja
- Tuneli, šift registri, povratne petlje

While petlja






```
Repeat (code);
Until Condition met;
End;
```

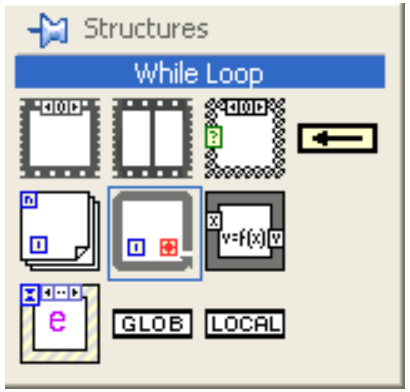
LabVIEW While petlja

Algoritam

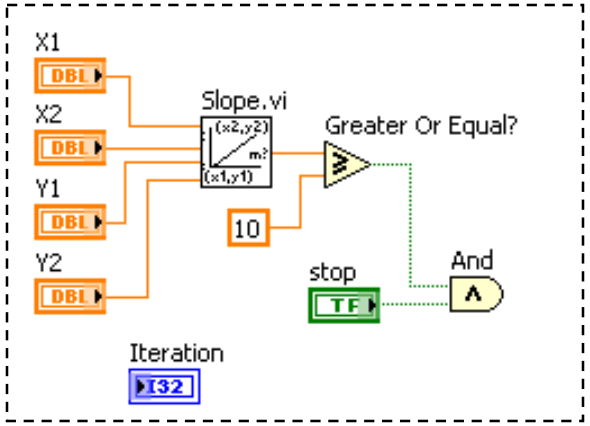
Pseudo kôd

While petlja

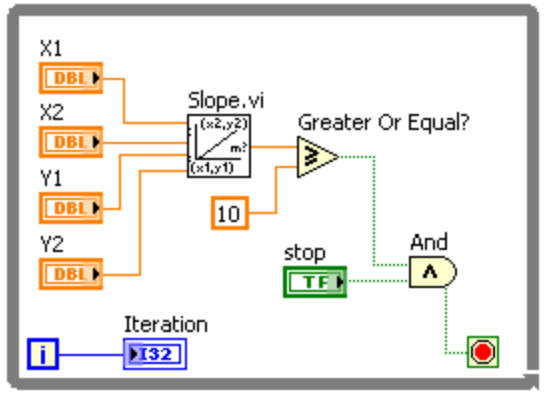
1. Selektovati While petlju



2. Uokviriti kôd koji se ponavlja



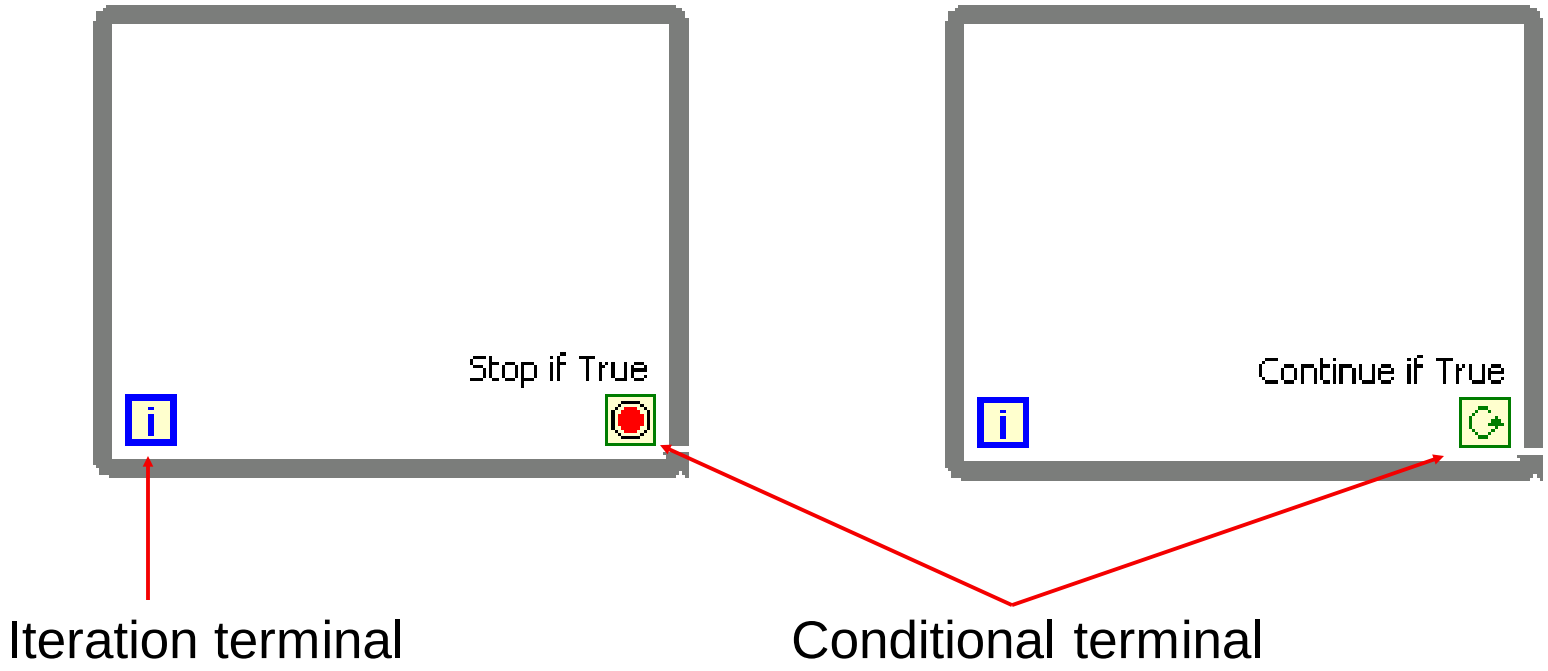
3. Dodati dodatne čvorove i veze



Uslov ponavljanja petlje

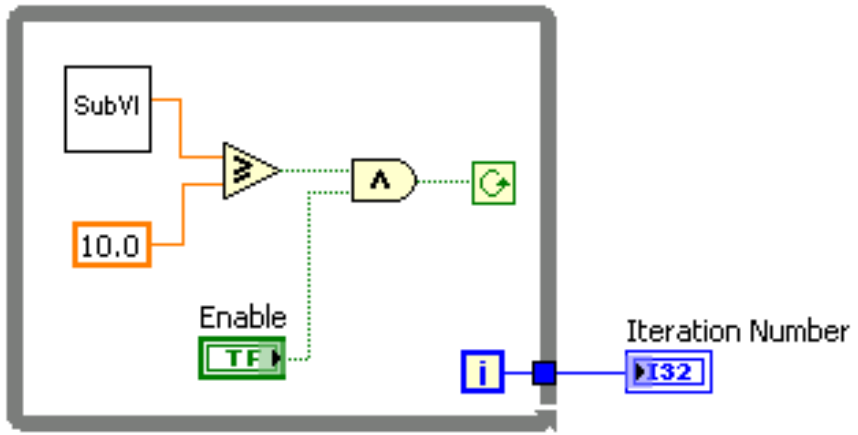
Klikom na Conditional terminal može se definisati uslov ponavljanja petlje

Podrazumevani način: Stop if True (prekini ukoliko je uslov ispunjen)

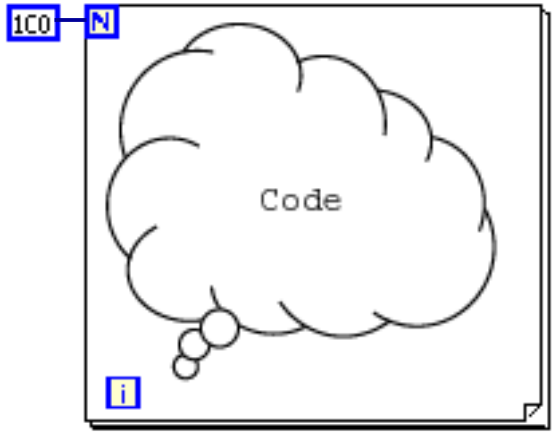


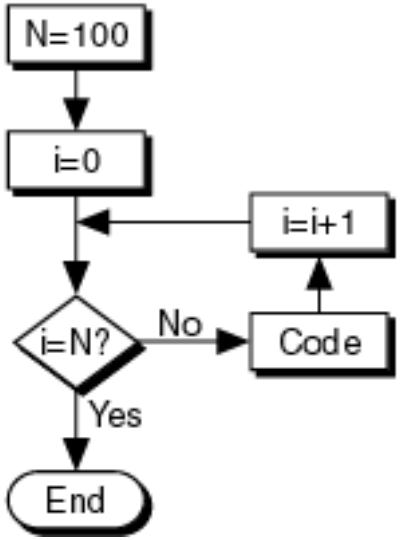
Tuneli strukture

- Tuneli služe za prosleđivanje podataka unutar i izvan petlje.
- Tunel je prikazan kao kvadrat na ivici strukture; boja kvadrata odgovara boji veze vezanoj za tunel, odnosno tipu podatka.
- Kada se podatak prosleđuje petlji, petlja počinje sa izvršavanjem tek nakon što je podatak postavljen na ulazni tunel.
- Podatak je dostupan na izlazni tunel nakon završetka izvršavanja petlje.



For petlja



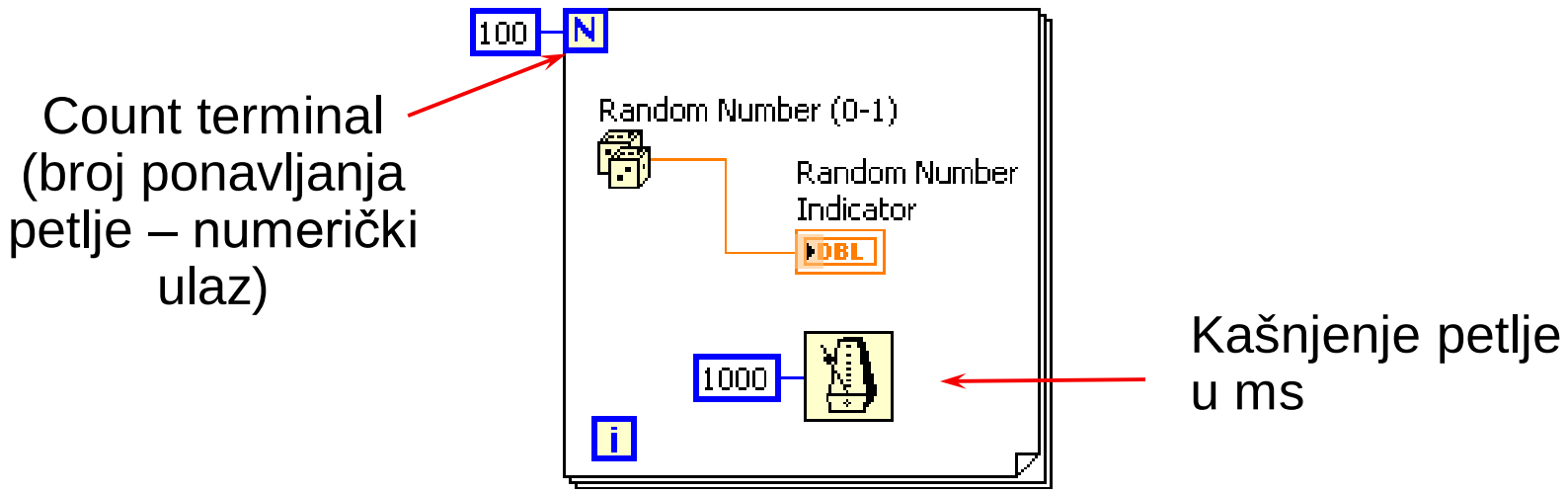



```
N=100;
i=0;
Until i=N:
  Repeat (code; i=i+1);
End;
```

LabVIEW For petlja Algoritam Pseudo kôd

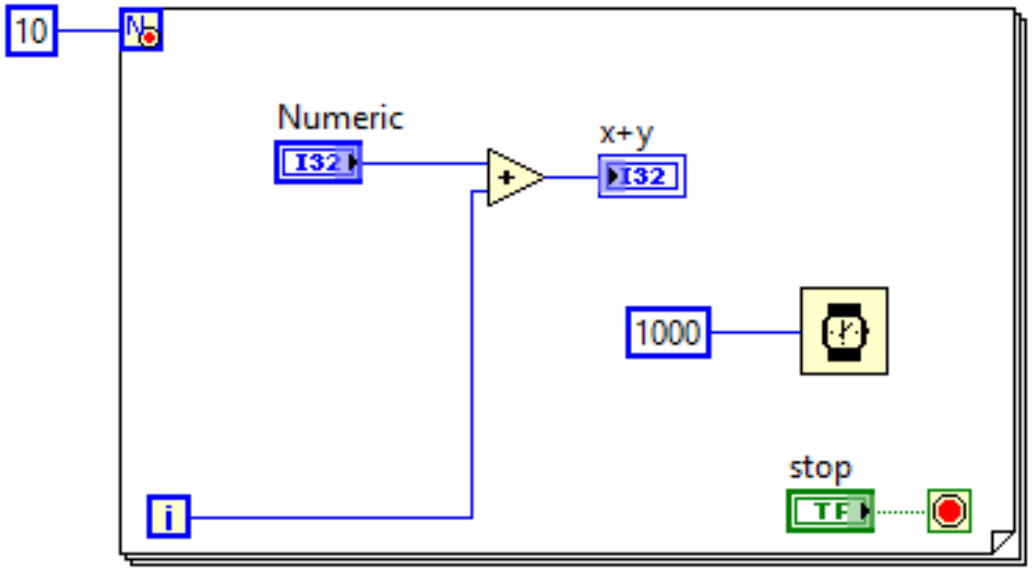
For petlja

- Nalazi se u Functions/Structures paleti
- Uokviriti kod koji se ponavlja i povezati čvorove
- Izvršava kod unutar petlje određeni broj puta



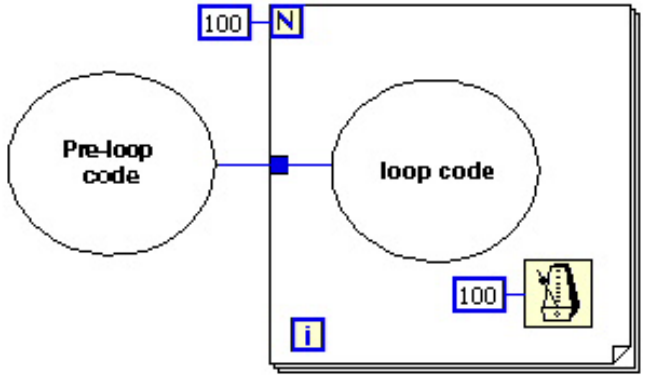
Uslovna For petlja

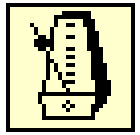
- Uslovna For petlja ima uslovni terminal, slično kao i While petlja
- Petlja će zaustaviti izvršavanje ukoliko je uslov zadovoljen
- Desni klik na strukturu selektovati **Conditional terminal**



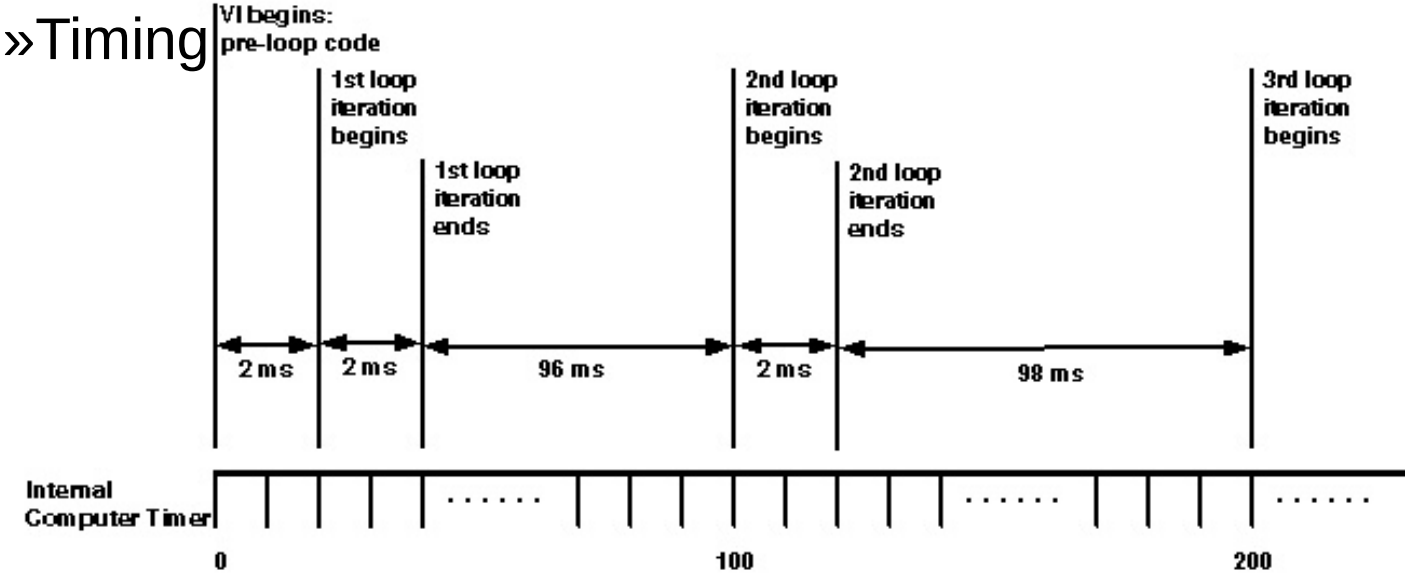
Wait funkcije

Čekaj do sledećeg celobrojnog umnoška u ms



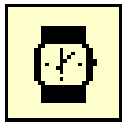


Programming»Timing

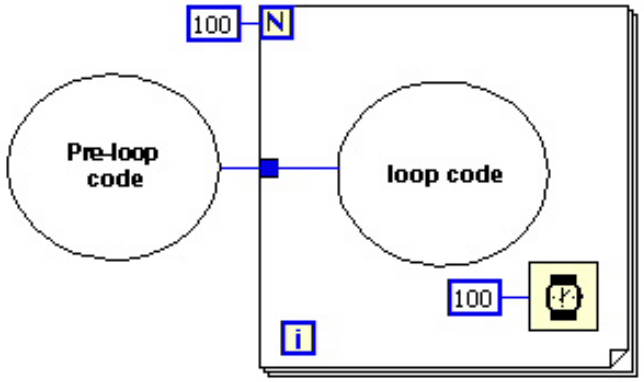


Wait funkcije

Wait (ms)

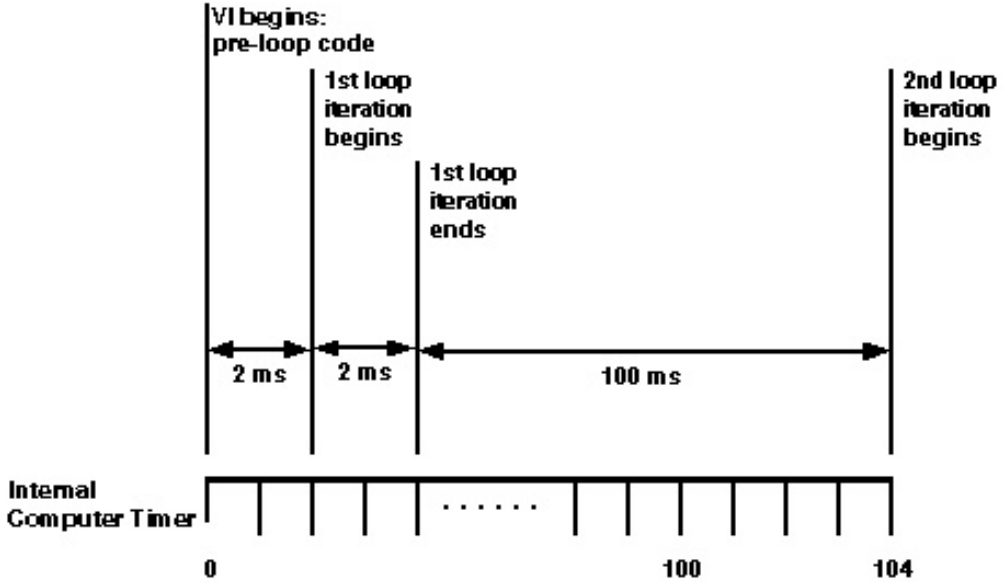


Programming»Timing



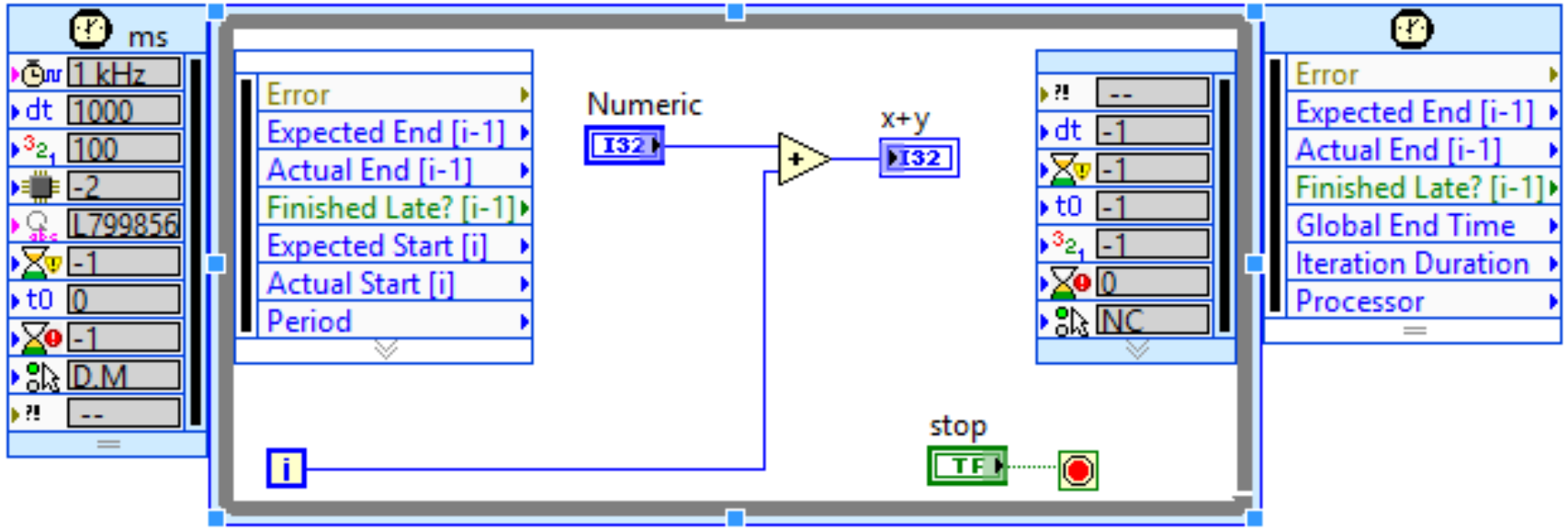
Time Delay

Programming»Timing



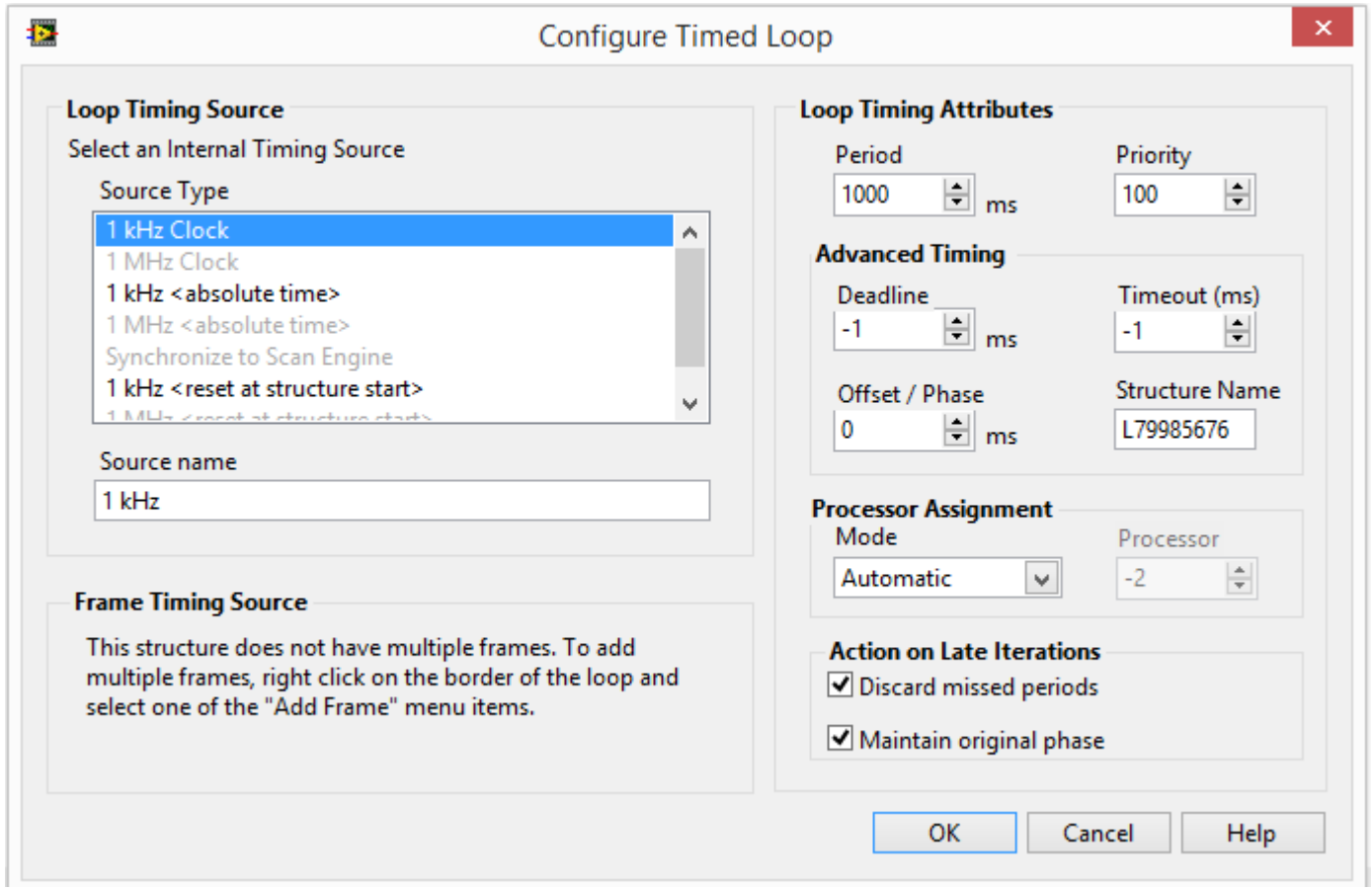
Timed loop petlja

- Timed loop petlja omogućava precizno vremensko izvršavanje, sinhronizaciju i mulitreading, izbor CPU
- Uglavnom se koristi kod real-time procesa



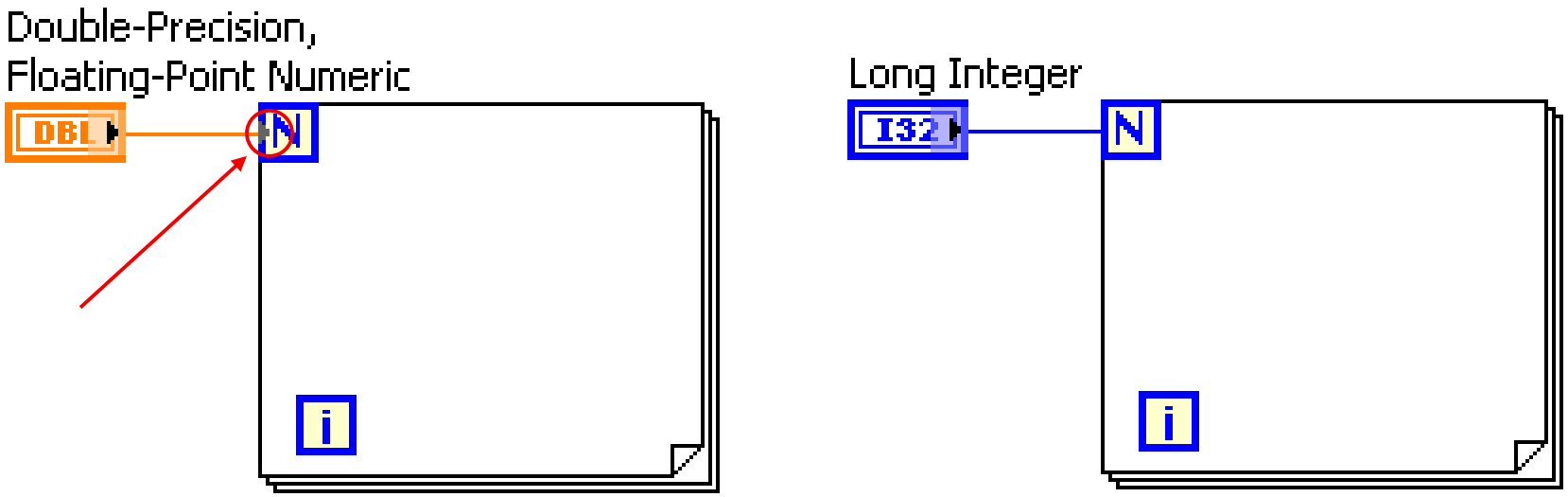
Timed loop petlja

- Konfiguracija petlje



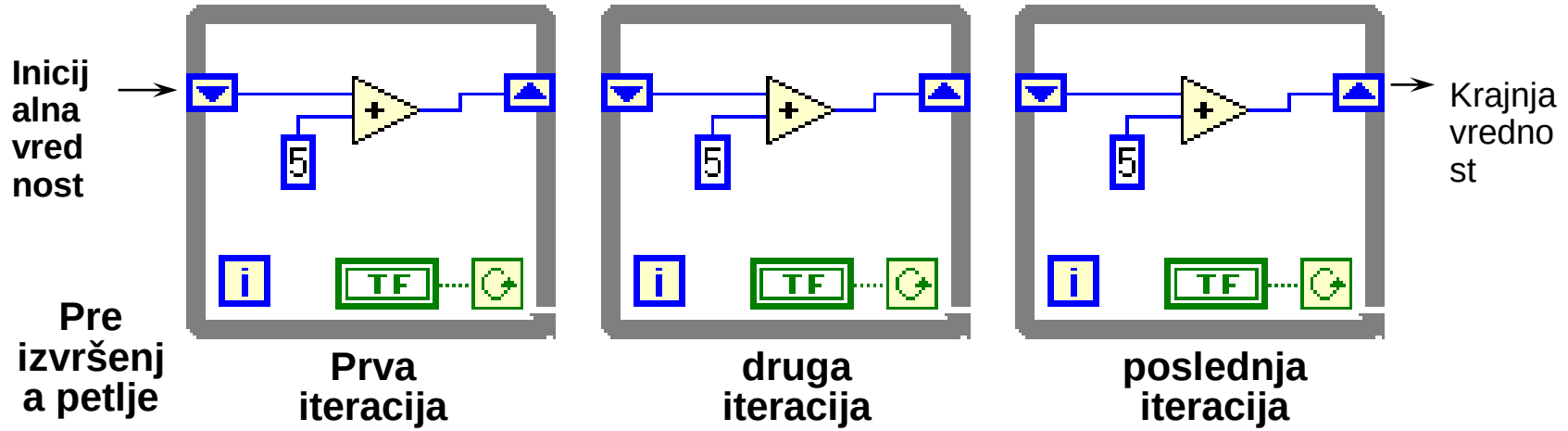
Numerička konverzija

- Podrazumevana numerička vrednost je broj dvostuke preciznosti (8 bajta) ili long-integer (4 bajta)
- LabVIEW automatski konvertuje brojeve u potreban tip
- For Loop count terminal uvek konvertuje u celi broj (integer)
- Tačla na ulazu ukazuje na izvršenu konverziju

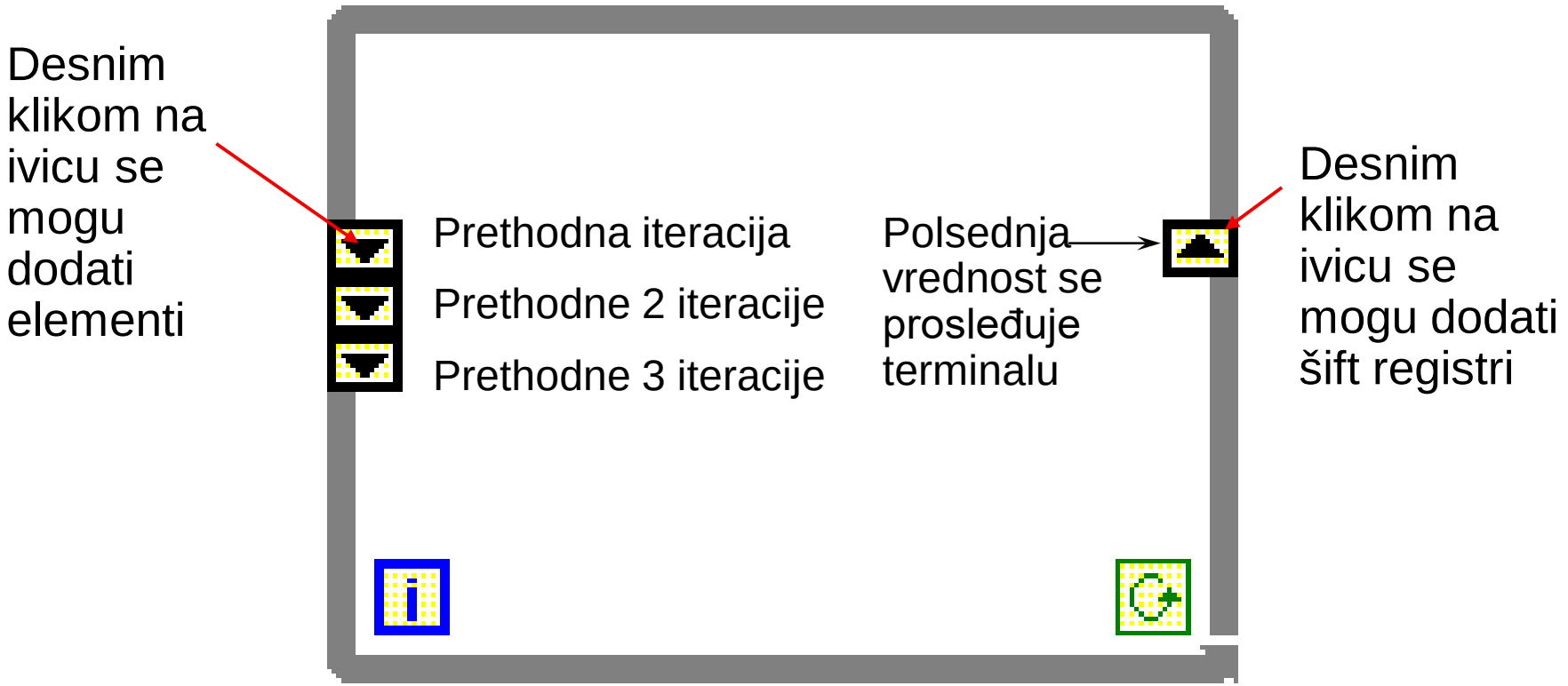


Vrednosti iz prethodne iteracije petlje – šift registar

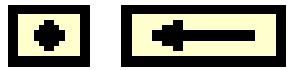
- Šift registar se postavlja na levoj/desnoj ivici strukture
- Desni klik na ivicu i selektovati Add Shift Register
- Desni terminal sadrži podatak koji je rezultat izvršavnja petlje
- Levi terminal prosleđuje podatak na početku svake iteracije



Šift registri



Povratne petlje (feedback node)



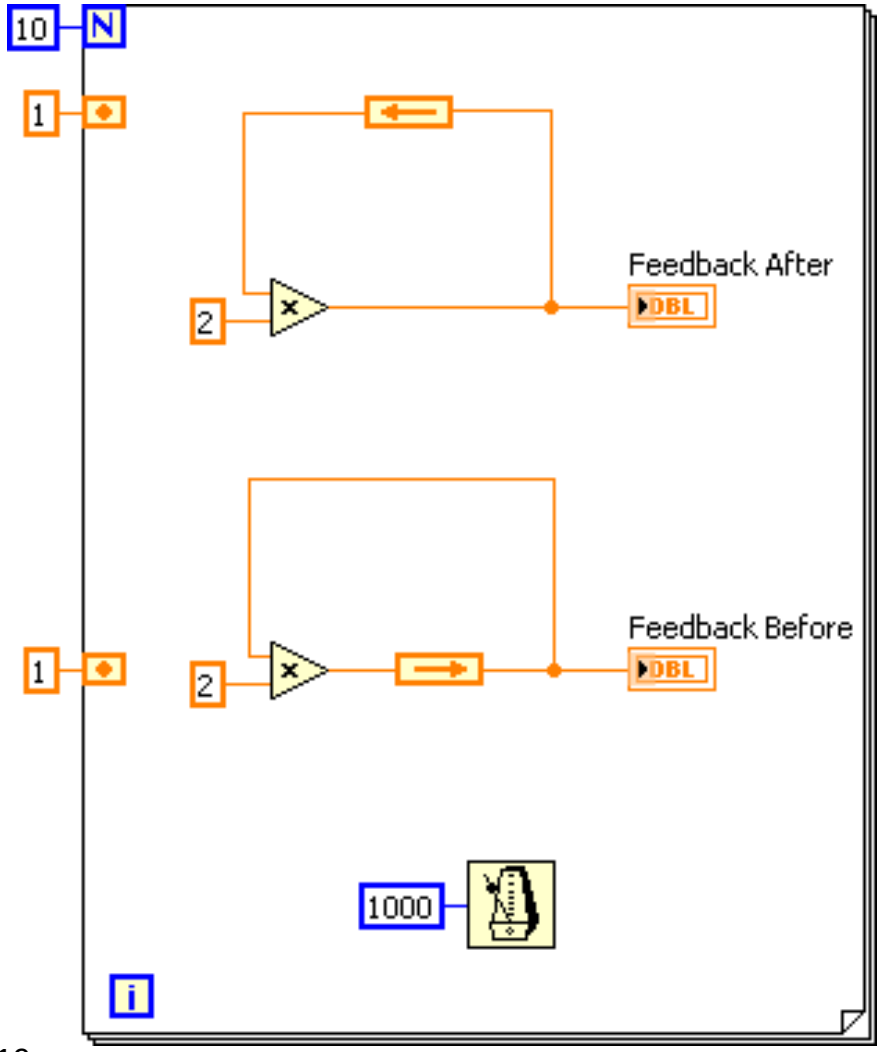
- Povratne petlje će se javiti automatski ukoliko se u For petlji ili While petlji poveže subVI, funkcija ili grupe subVI i funkcija na ulaz istog subVI, funkcije ili grupe.
- Povratna petlja čuva vrednost podatka do okončanja izvršenja iteracije petlje, i prosleđuje podatke sledećoj iteraciji. Može da prosledi podatak bilo kog tipa.

Povratne petlje

- Povezati izlaz funkcije na njen ulaz, čime se automatski kreira povratna petlja

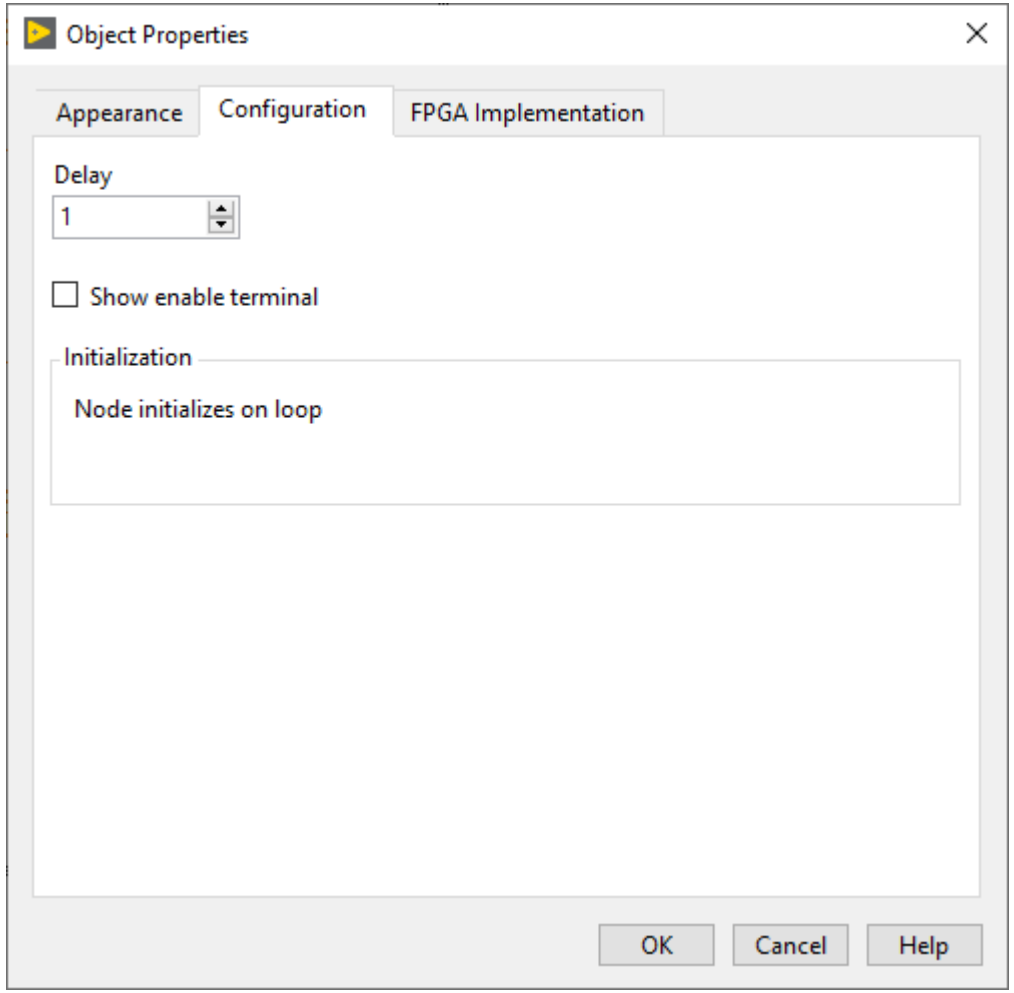
  ili

- Postaviti povratnu petlju iz **Functions»Structures** palete

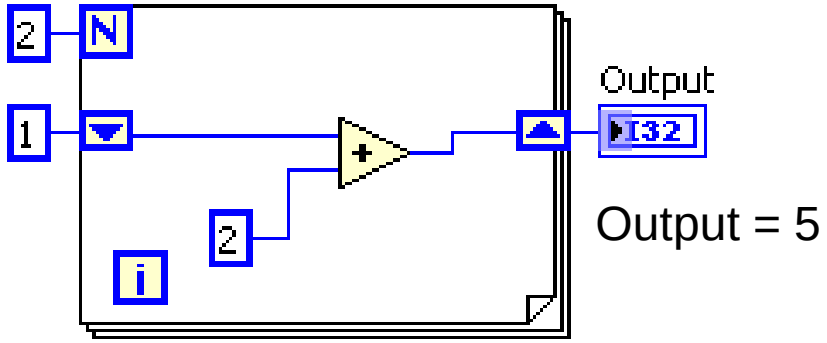


Povratne petlje

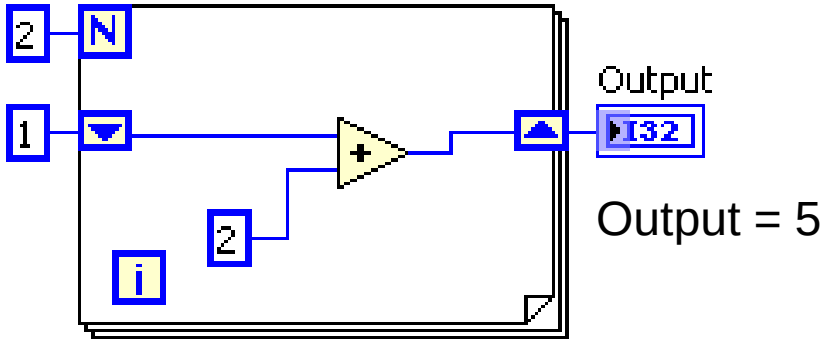
- Povratna petlja se može konfigurisati pozivom opcije **Properties** iz kontekstnog menija
- U dijalogu se može konfiguriati Delay, parametar koji određuje broj iteracija



Inicijalizacija šift registra i povratnih petlji

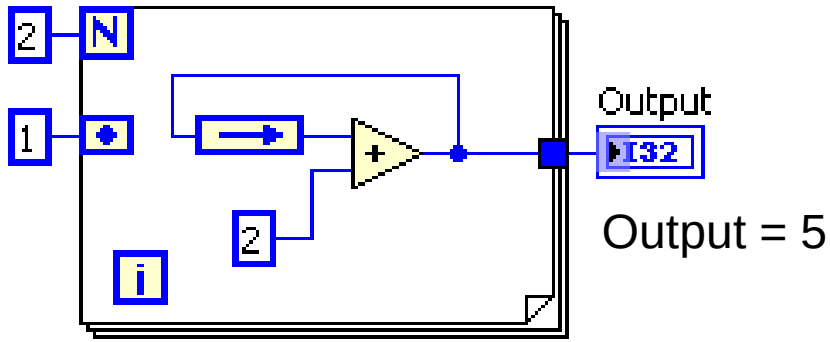


Output = 5

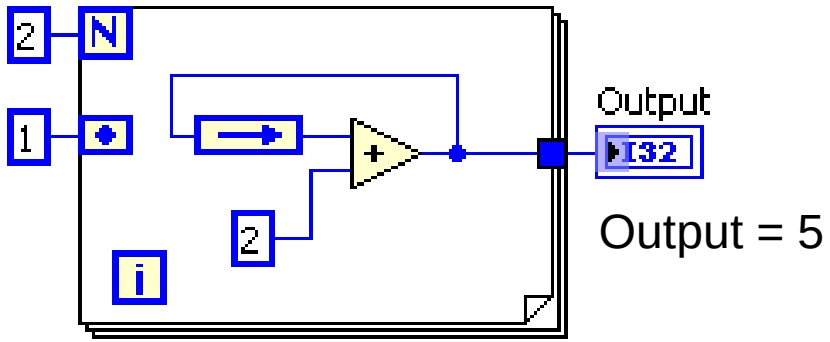


Output = 5

pokreni → kraj → pokreni

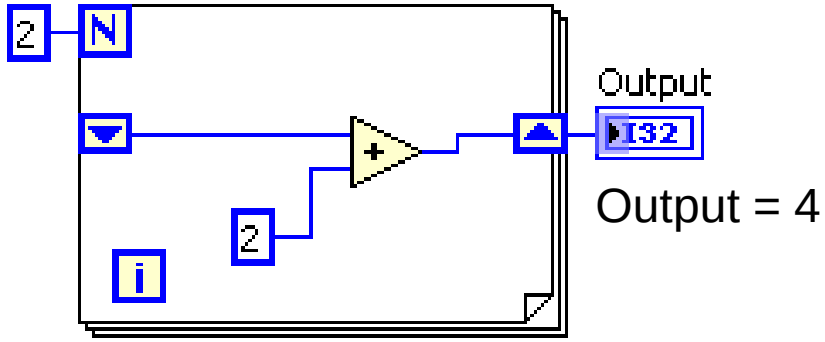


Output = 5

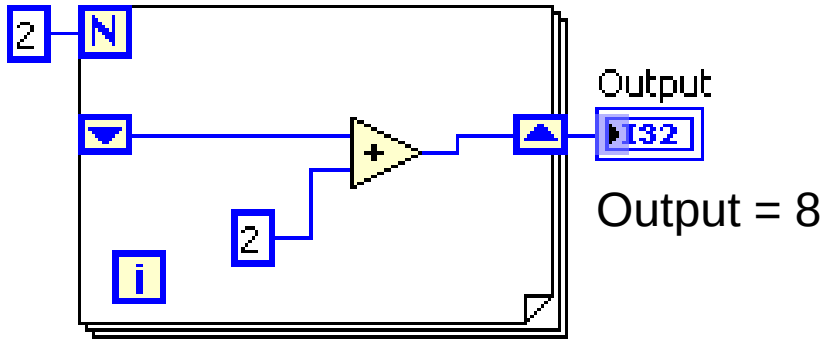


Output = 5

Neinicijalizovani šift registri i povratnih petlji

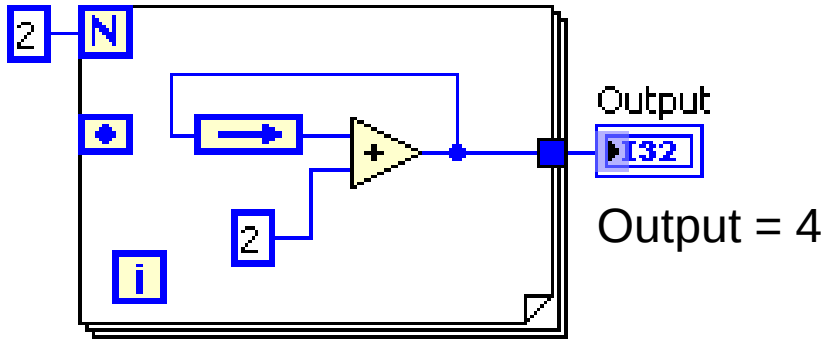


Output = 4

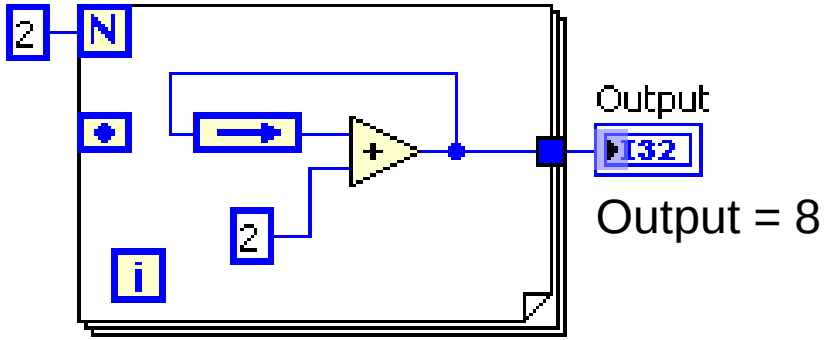


Output = 8

pokreni → kraj → pokreni



Output = 4



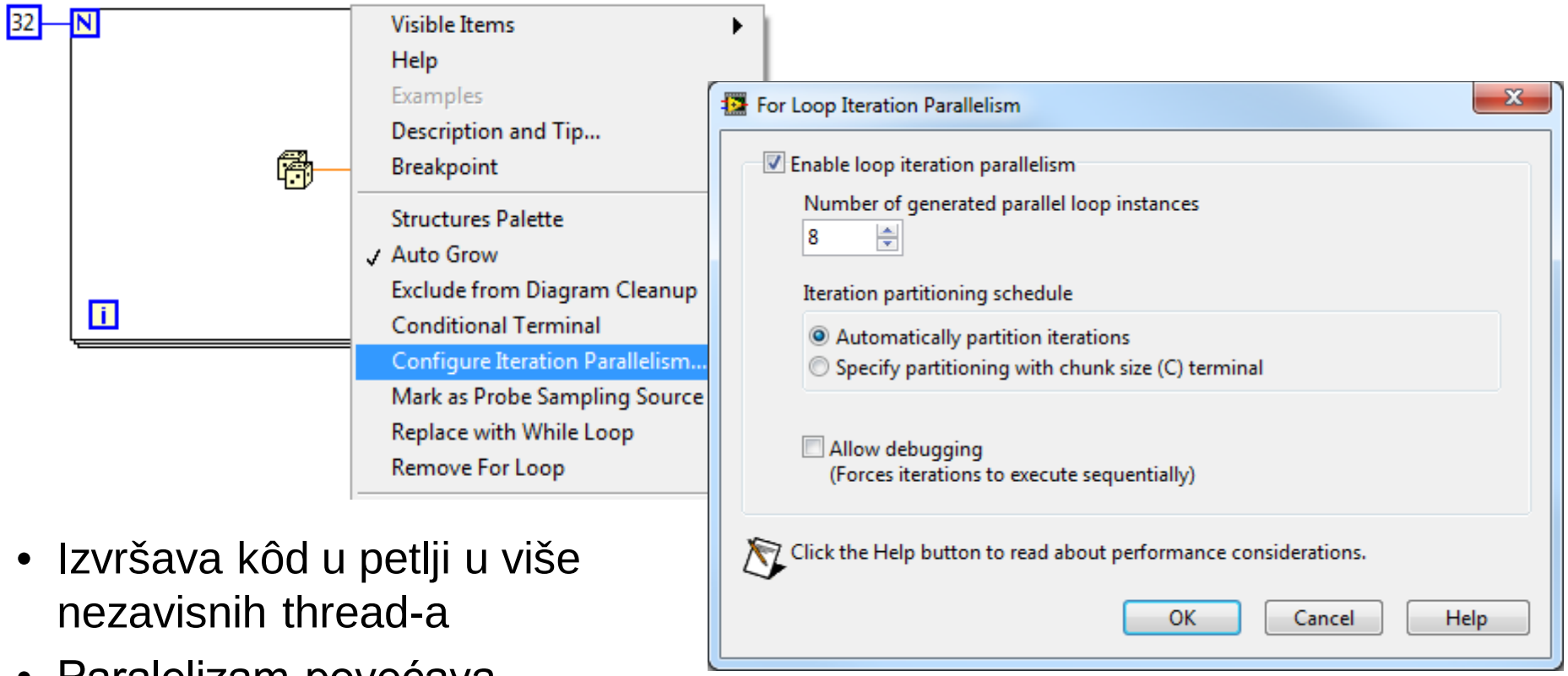
Output = 8

Vežba 3 – Sabiranje prirodnih brojeva od 1 do N

Sabiranje brojeva

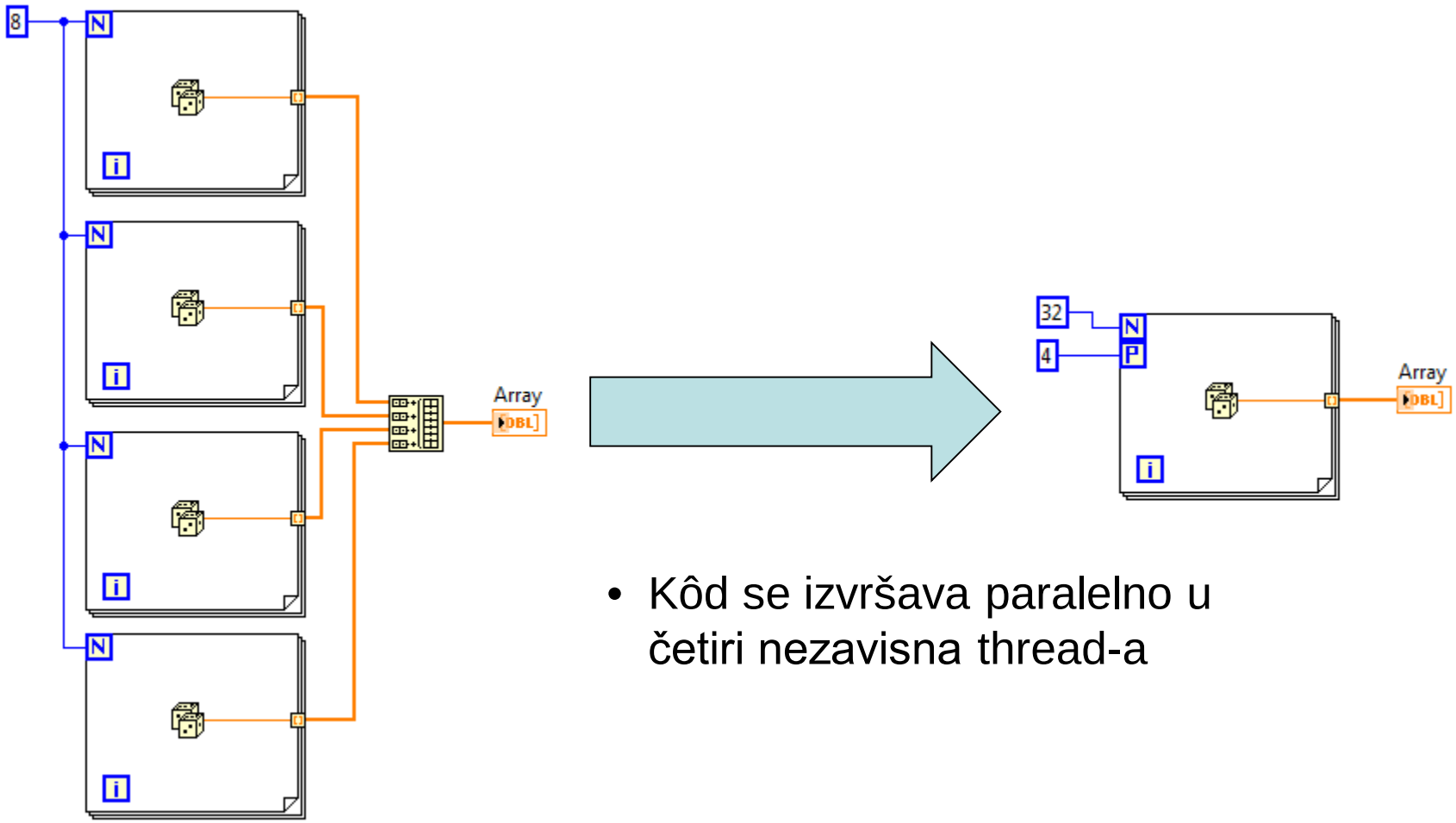
- Realizujte VI koji sabira prirodne brojeve od 1 do N
- Vrednost N se zadaje kontrolom
- Vrednost zbira ispisuje na indikatoru, na kraju izvršavanja
- Za realizaciju koristiti FOR petlju!

Paralelizam u For petlji



- Izvršava kôd u petlji u više nezavisnih thread-a
- Paralelizam povećava performanse aplikacije

Paralelizam u For petlji



- Kôd se izvršava paralelno u četiri nezavisna thread-a

Paralelizam u For petlji

- Nije moguće paralelizovati kôd koji se izvršava sekvencijalno, gde ulazni parametri petlje zavise od izlaznih parametara čije vrednosti se dobijaju iz prethodne iteracije.
- Većina petlji bez šift registara i povratne petlje se može paralelizovati.

Pregled

- Dve osnovne strukture za ponavljanje izvršenja koda: While petlja i For petlja
- Kontrola vremena izvršenja je moguća pomoću Wait Until Next ms Multiple funkcije, Wait (ms) funkcije, ili Time Delay Express VI.
- Tačke za zaokruženje se javljaju kada LabVIEW zaokružuje brojeve na terminalu kako bi ih uskladio prema tipu na drugom terminal
- Povratne petlje i šift registri služe za prenos podatak iz jedne iteracije petlje u narednu
- Koristite šift registre ukoliko je potrebno pristupiti podacima iz više nego jedne prošle iteracije

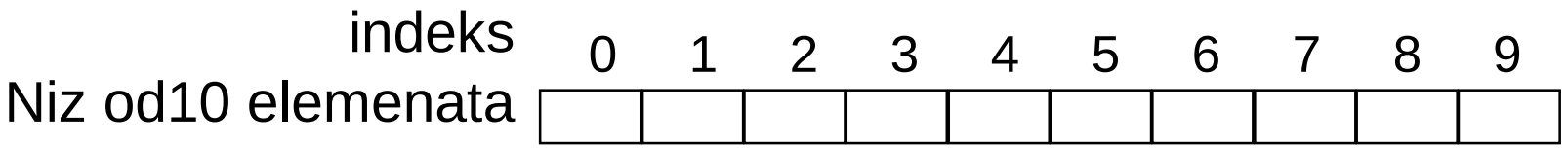
Nizovi

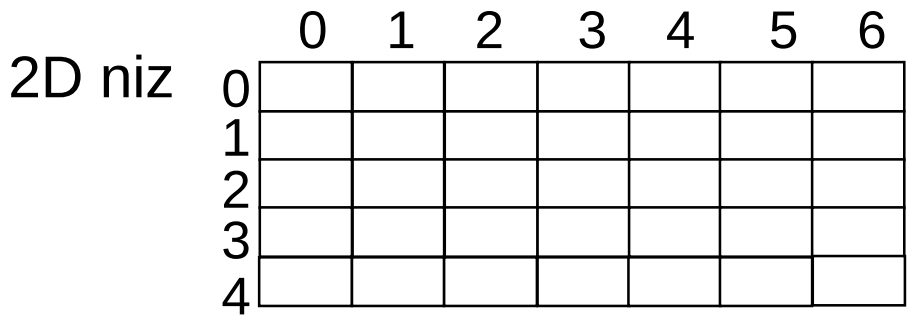
Nizovi

- Uvod
- Automatsko indeksiranje nizova
- Funkcije nizova
- Polimorfizam

Nizovi

- Niz je skup pobrojanih elemenata istog tipa
- Mogu biti jednodimenzionalni ili višedimenzionalni
- Elementima niza se pristupa preko indeksa; prvom elementu odgovara indels 0

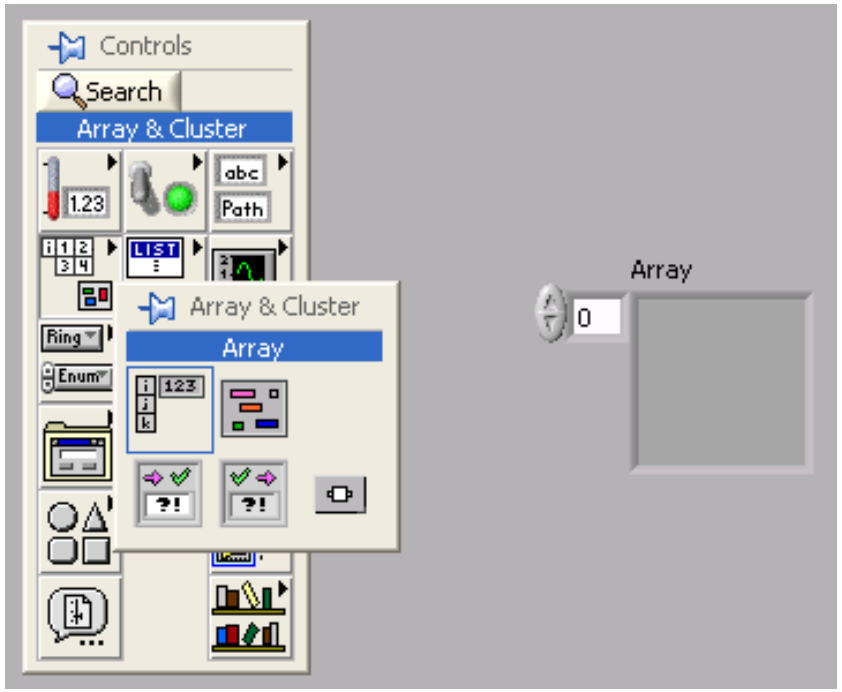




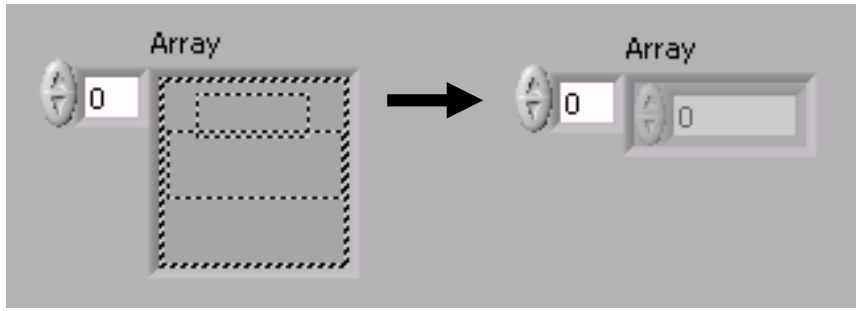
Pet redova puta sedam kolona, 35 elemenata

Kontrole i indikatori nizova

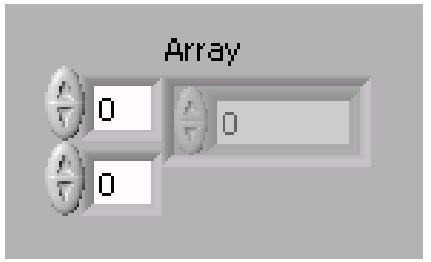
1. Selektovati Array okvir iz Controls palete



2. Postaviti objekt unutar okvira niza

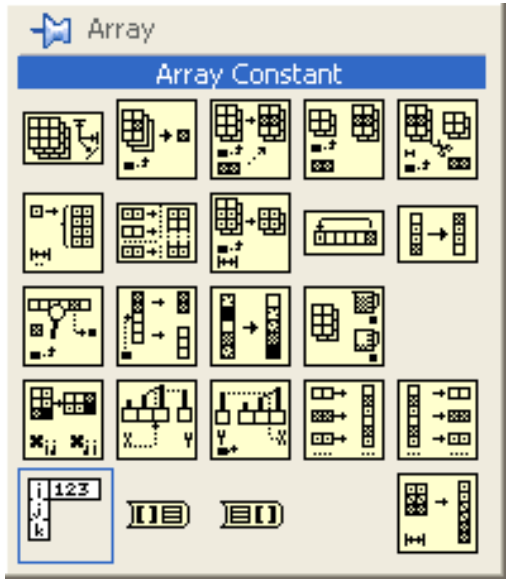


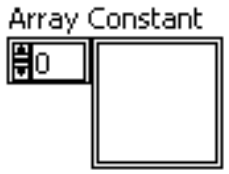
Dodati dimenziju za 2D nizove



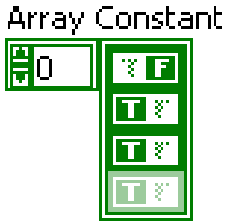
Konstantni niz

1. Selektovati Array Constant iz Array palete





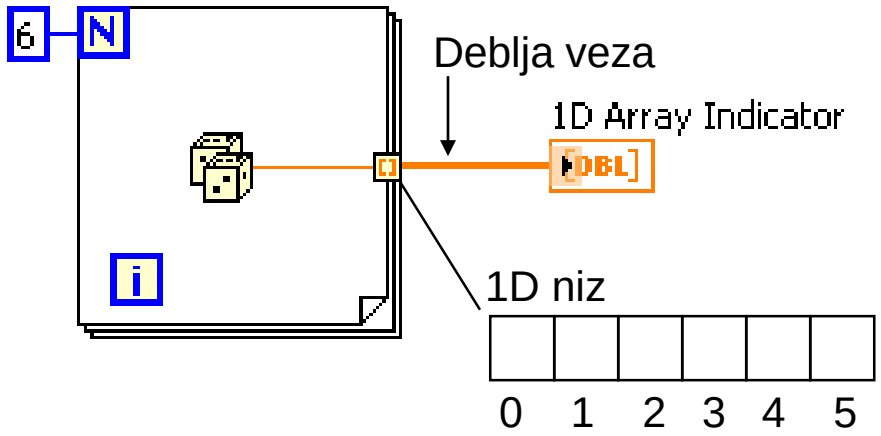
2. Postaviti objekt unutar okvira konstantog niza



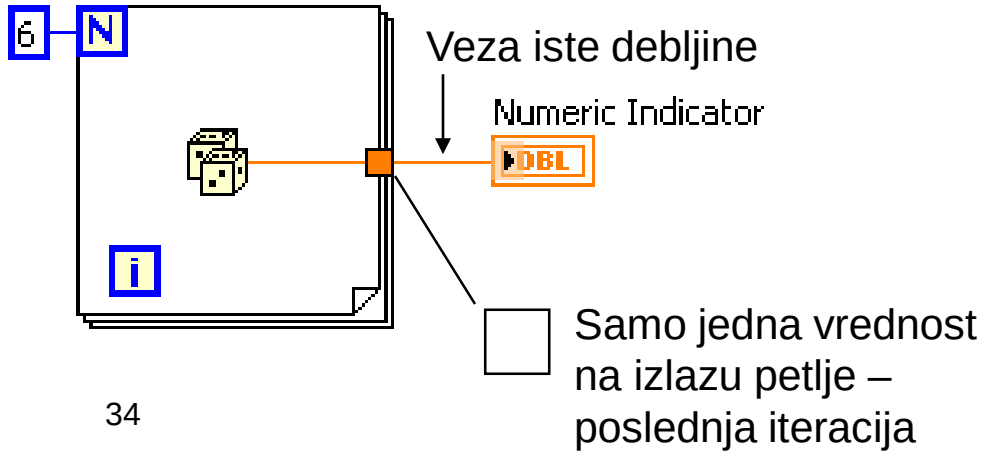
Autoindeksiranje

- Petlje mogu da akumuliraju podatke u niz pomoću opcije autoindeksiranja
- For petlje podrazumevaju opciju autoindeksiranja
- While petlje prosleđuju vrednost samo poslednje iteracije
- Desnim klikom na tunel, izborom **Tunel Mode/Last Value**

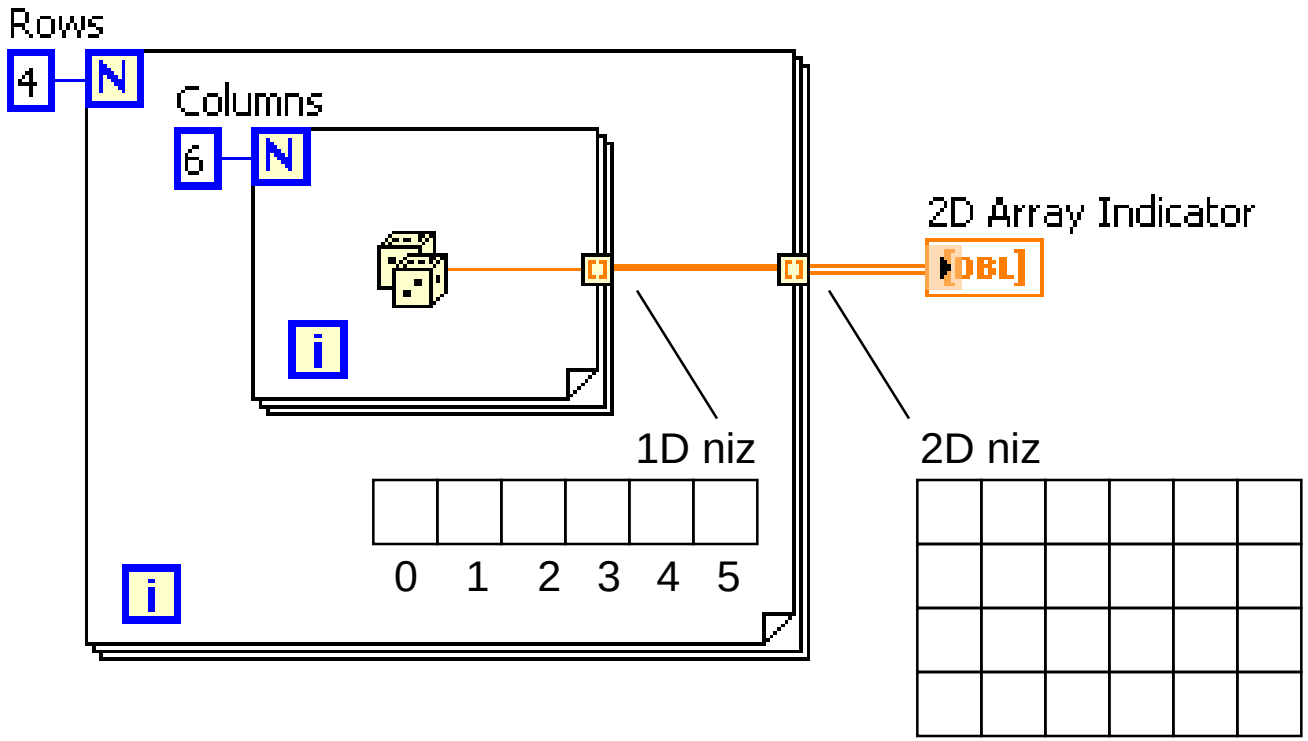
Autoindeksiranje omogućeno



Autoindeksiranje onemogućeno



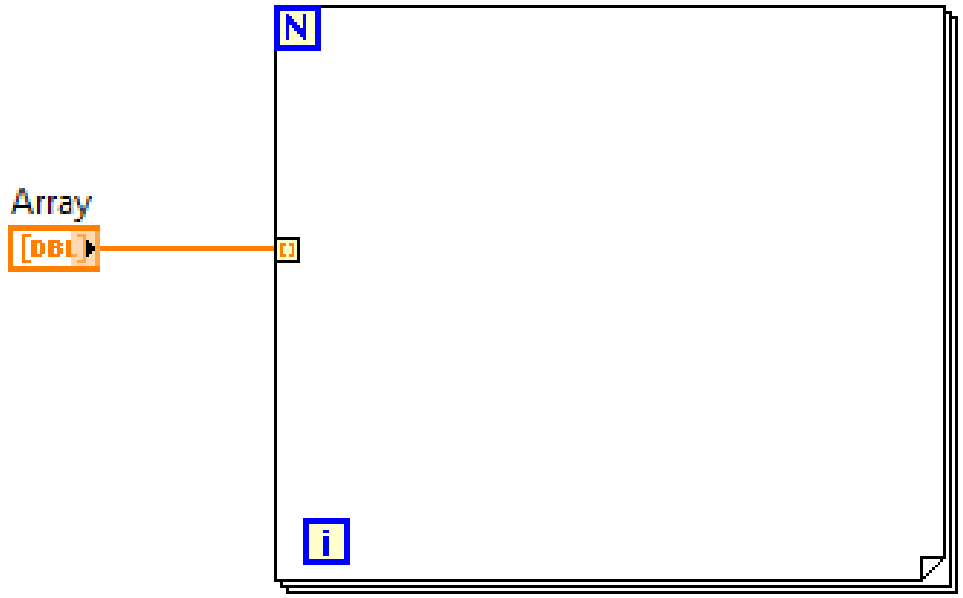
Realizacija 2D nizova



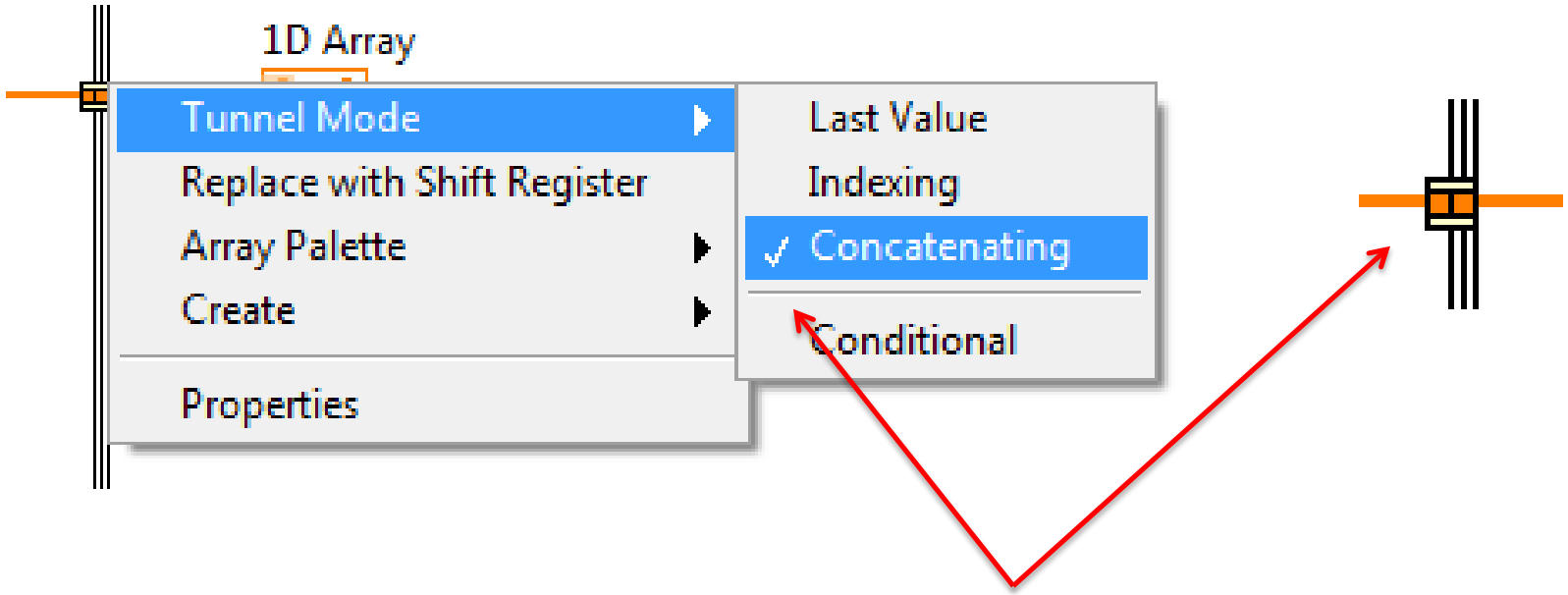
- Unutrašnja petlja kreira kolone
- Spoljna petlja pakuje u redove

Autoindeksiranje ulaznog niza

- Ulazni niz se može iskoristiti umesto Loop count terminala
- Broj elemenata niza će biti jednak broju ponavljanja petlje
- Run strelica neće prijavljivati grešku

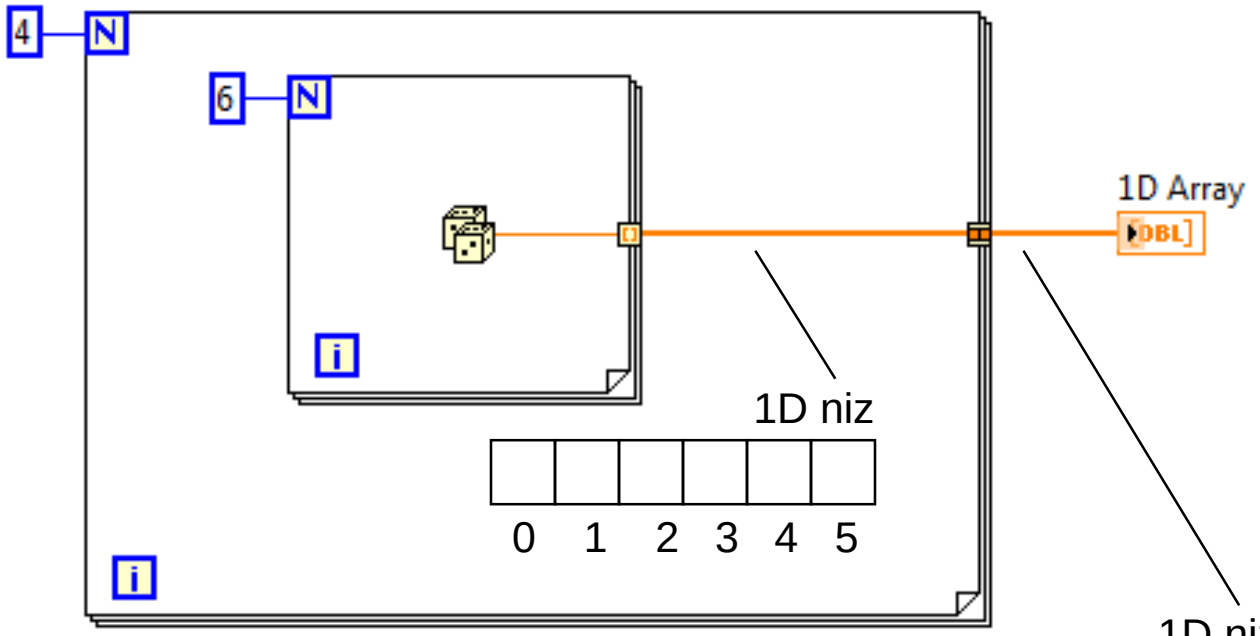


Konkatenacija nizova



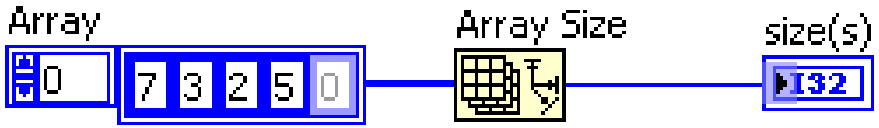
- Konkatenacija nizova
- Tunel dobija specifičan oblik

Konkatenacija nizova

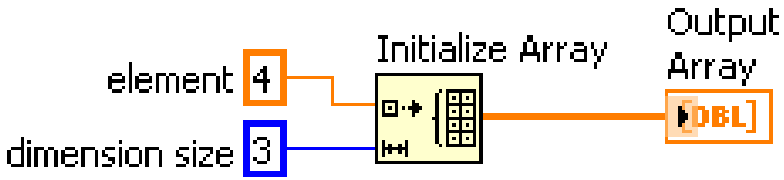


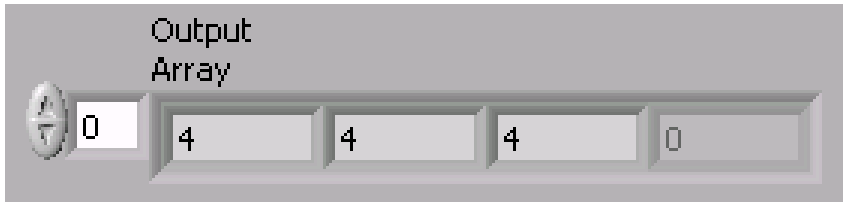
Izlazni niz je 1D niz, nastao spajanjem nizova formiranih u unutrašnjoj petlji

Najčešće korišćene funkcije



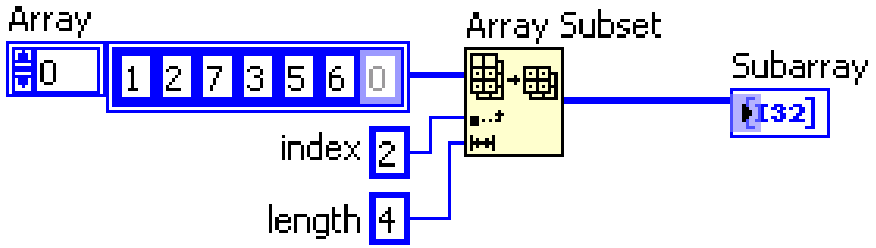
Array Size



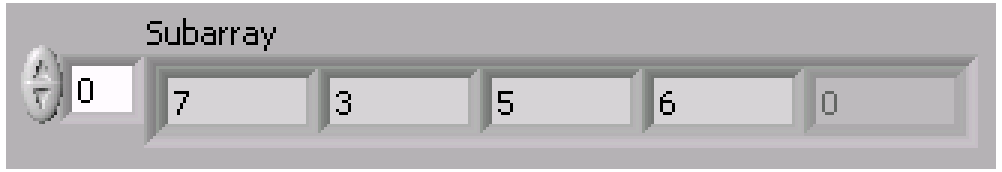


Initialize Array

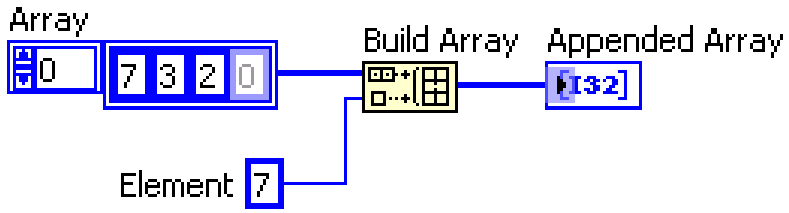
Najčešće korišćene funkcije



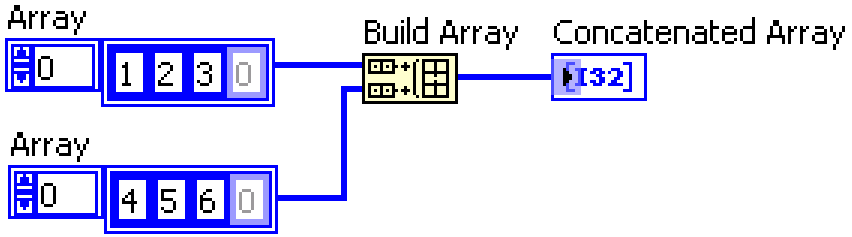
Array Subset



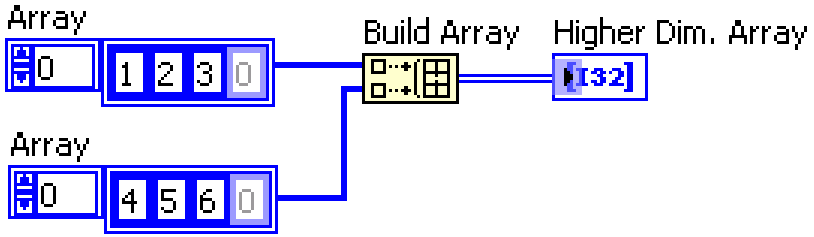
Build Array funkcija



Dodavanje elementa

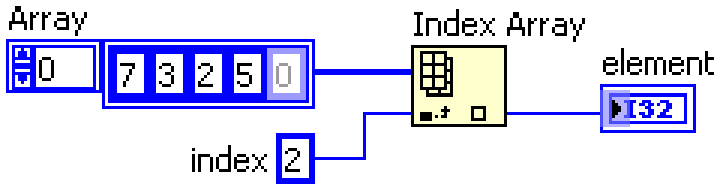


Spajanje nizov

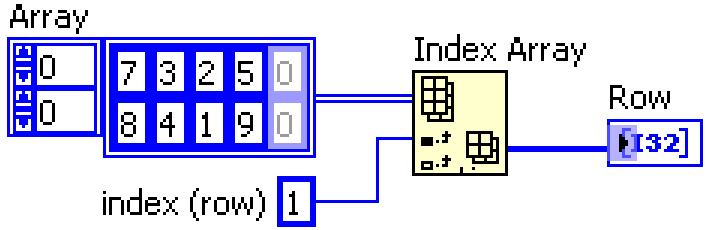


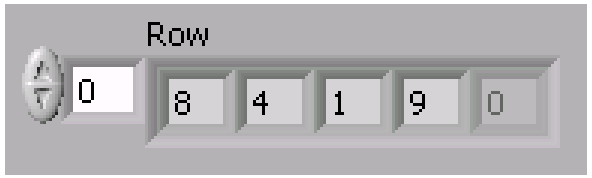
Dodavanje dimenzije *(default)*

Index Array funkcija

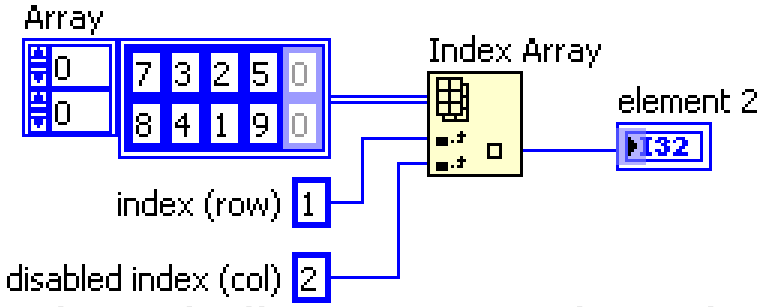


Ekstrakcija elementa





Ekstrakcija reda

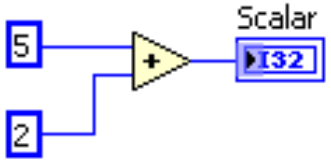
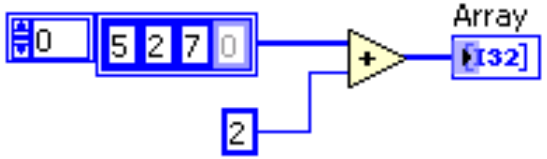
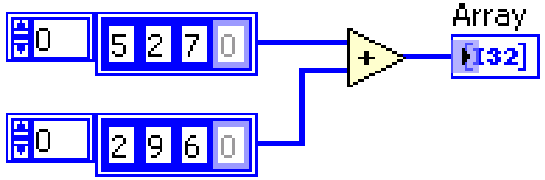
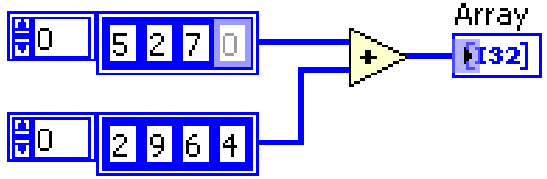


Ekstrakcija elementa iz reda

Polimorfizam

Ulazi funkcija mogu biti različiti tipovi

Sve LabVIEW aritmetičke funkcije su polimorfne

Kombinacija		rezultat	
Skalar + skalar		skalar	
Niz + skalar		niz	
Niz + niz		niz	
Niz + niz		niz	

Pregled

- Nizovi su indeksirani skupovi elemenata istog tipa. Mogu biti numerički, logički, nizovi karaktera ili složenijih tipova.
- Indeks niza počinje od nule, završava se sa n – 1, gde je n broj elemenata niza.
- Kontrola ili indikator niza se mogu realizovati selekcijom **Array** na **Controls»Array & Cluster** paleti, postavljanjem na front panel i prevlačenjem kontrole ili indikatora u okvir niza.
- Ukoliko se niz veže za ulazni tunel For ili While petlje, procesiraće se svaki element niza u iteraciji petlje ukoliko je omogućeno autoindeksiranje.
- LabVIEW podrazumeva autoindeksiranje u For petlji i klasičan tunel u While petlji.
- Polimorfizam je mogućnost funkcije da prilagodi ulazne podatke.

Vežba 4 – Rastojanje između dve tačke u prostoru

Rastojanje tačaka

- Realizujte VI koji izračunava rastojanje dve tačke A i B u prostoru
- Tačke su zadate koordinatama u Dekartovom koordinatnom sistemu $A(x_1, y_1, z_1)$ i $B(x_2, y_2, z_2)$
- Rastojanje se ispisuje na indikatoru, na kraju izvršavanja

Vežba 5 – Skalarni proizvod vektora

Skalarni proizvod

- Realizujte VI koji izračunava skalarni proizvod vektora A i B u prostoru
- Vektori su zadati koordinatama u Dekartovom koordinatnom sistemu $A(x_1, y_1, z_1)$ i $B(x_2, y_2, z_2)$
- Skalarni proizvod se ispisuje na indikatoru, na kraju izvršavanja

Vežba 6 – Vektorski proizvod vektora

Vektorski proizvod

- Realizujte VI koji izračunava vektorski proizvod vektora A i B u prostoru
- Vektori su zadati koordinatama u Dekartovom koordinatnom sistemu $A(x_1, y_1, z_1)$ i $B(x_2, y_2, z_2)$
- Vektroski proizvod se ispisuje na indikatoru, na kraju izvršavanja