

Izvršne strukture u LabView-u

Pregled

1. Petlje – While i For
2. Case struktura
3. Druge struktura

While i For petlje

```
while (y<6)
```

```
... procedura
```

```
end
```

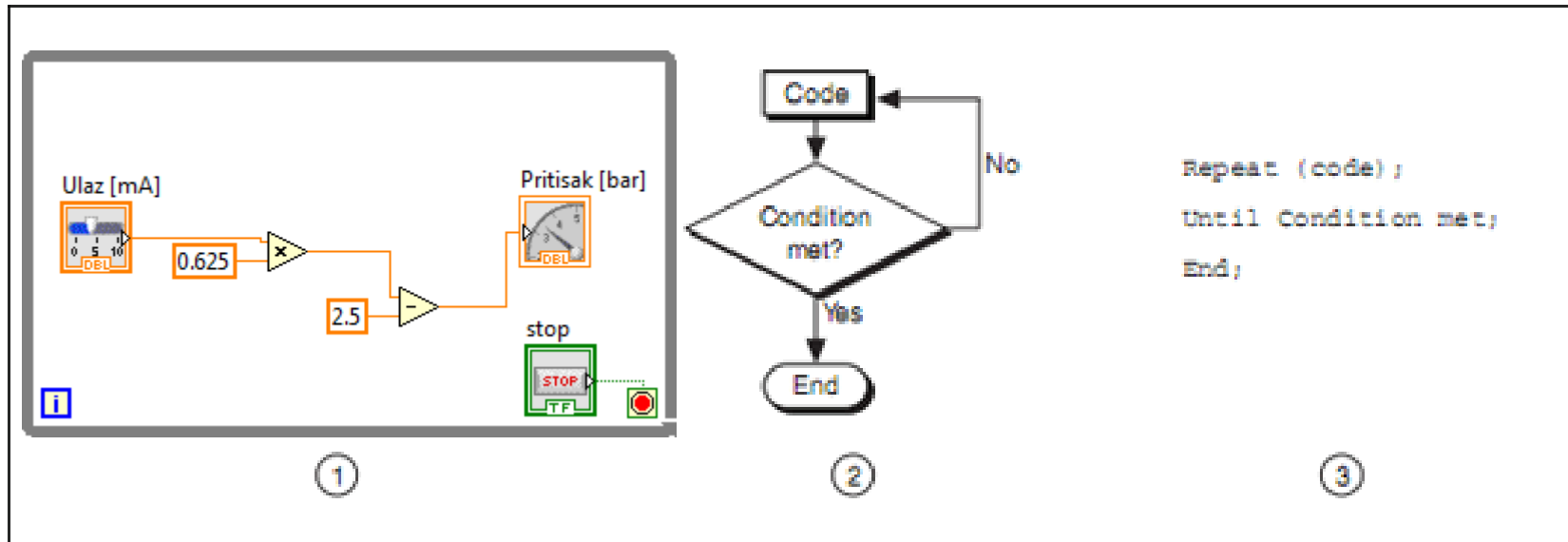
```
for i=1:100
```

```
... procedura
```

```
end
```

LabView ovo radi grafički

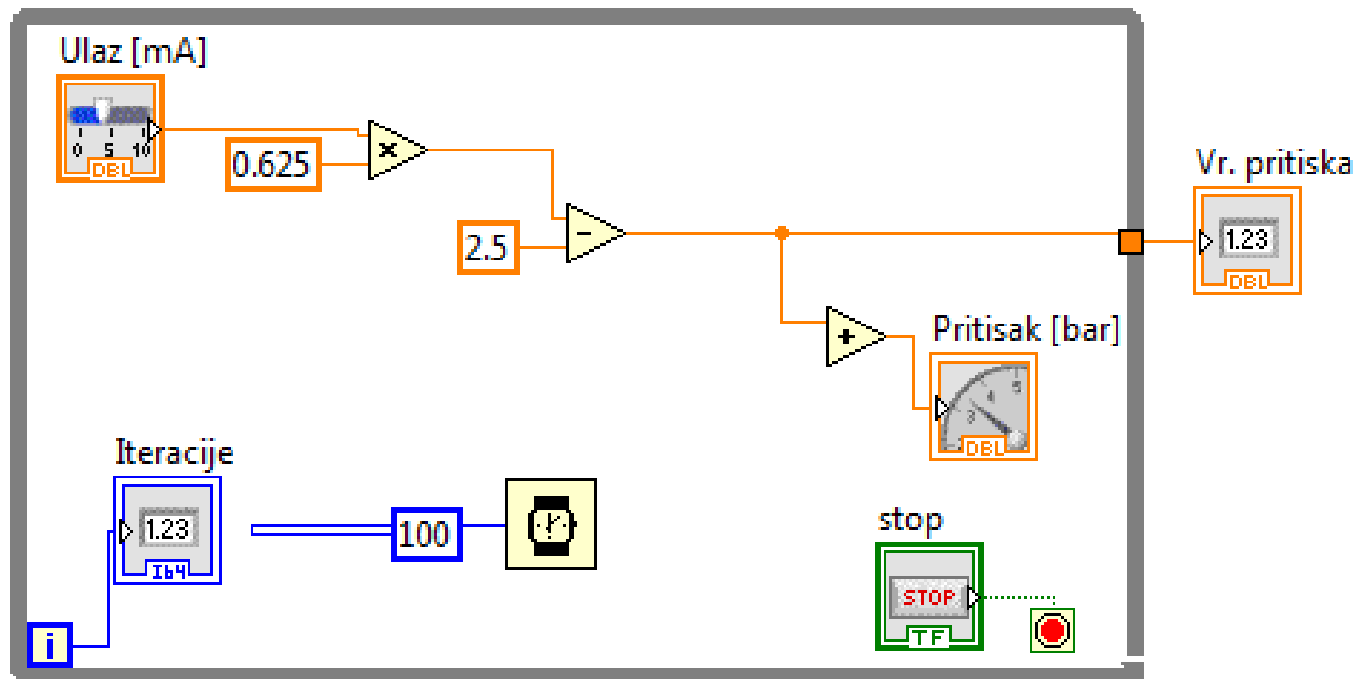
While petlje



- While petlja se izvršava sve dok uslovni terminal na dobije određenu boolean vrednost.
- Iteracije se broje od 0!
- Izvršavanje se obavlja “što je brže moguće” – nije specificirano vreme
- Korisno uvesti “wait” za uštedu resursa!

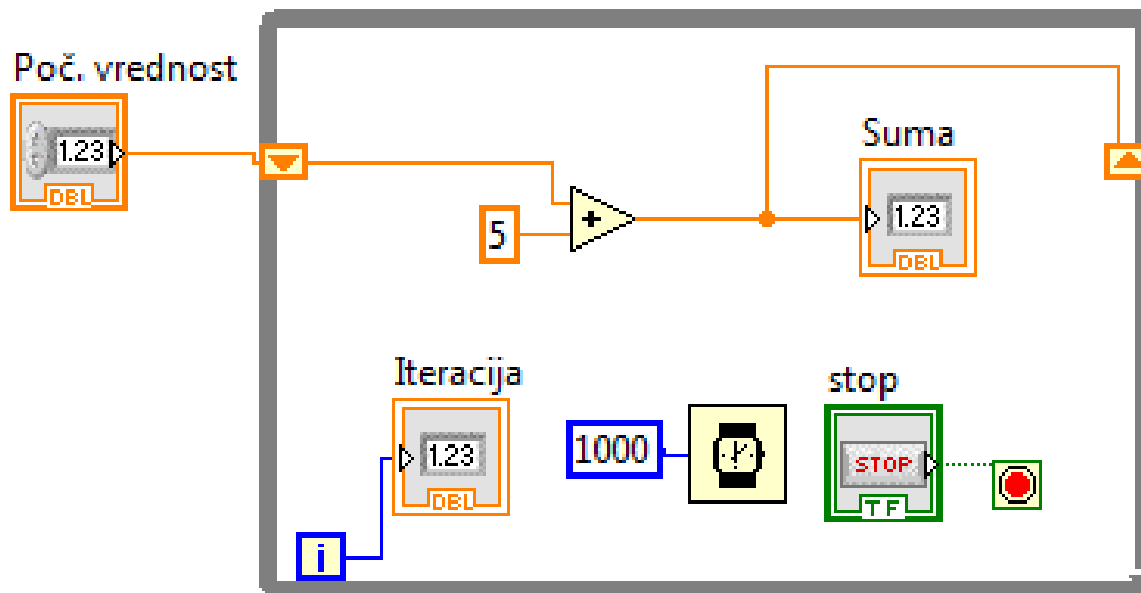
Value tunnel

- Tunnel - prosleđuje se brojčana vrednosti van petlje
- Tek nakon zaustavljanja!

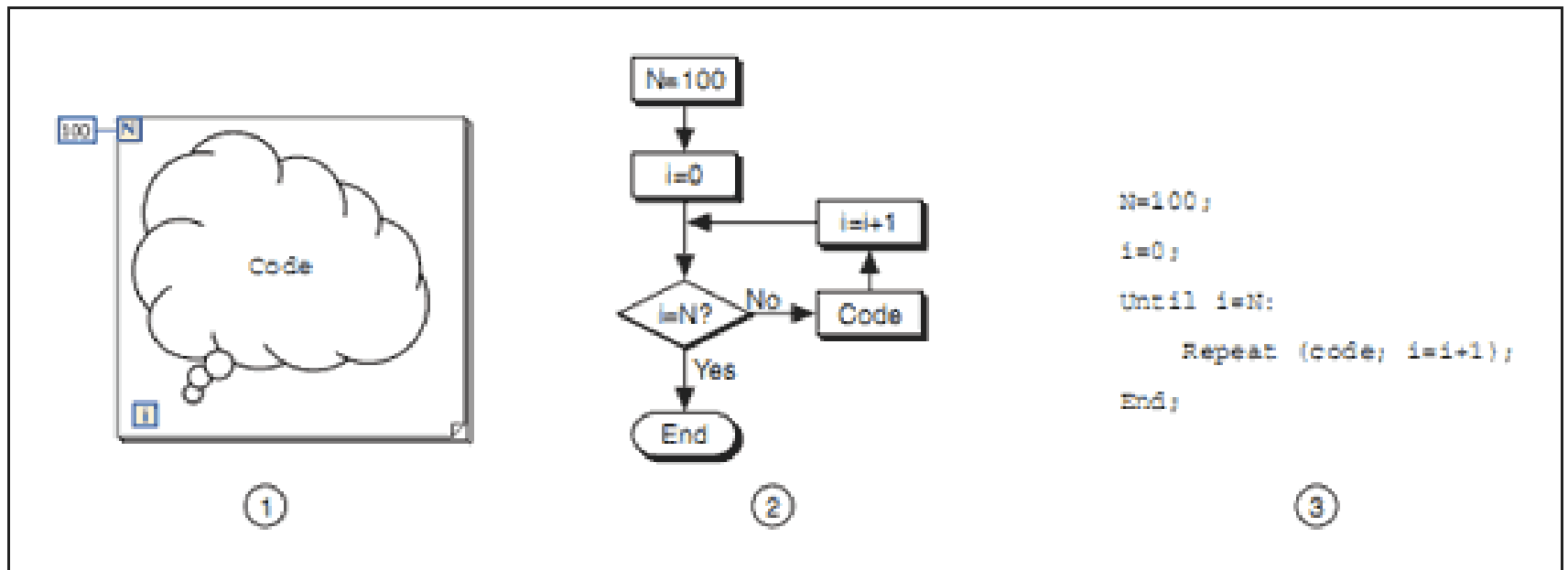


Shift register

- Vrednost se ne čuva sama po sebi (žice nemaju “memoriju”) – PROBLEM!
- Shift register – čuva poslednju vrednost i prosleđuje je u sledeću iteraciju; polimorfan, prilagođava se tipu podataka koji mu se prosleđuje



For petlje



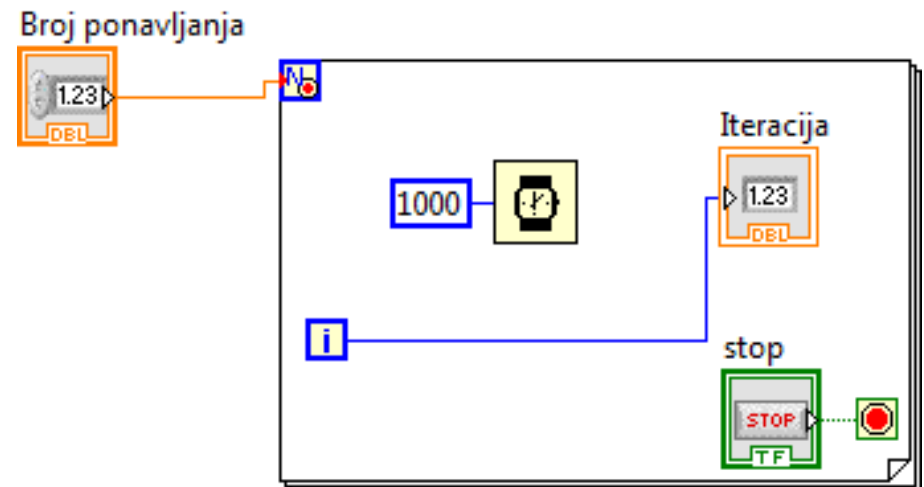
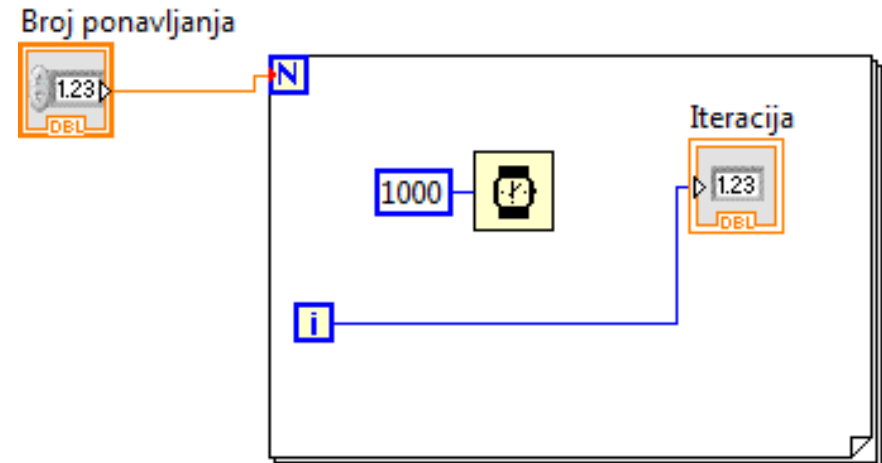
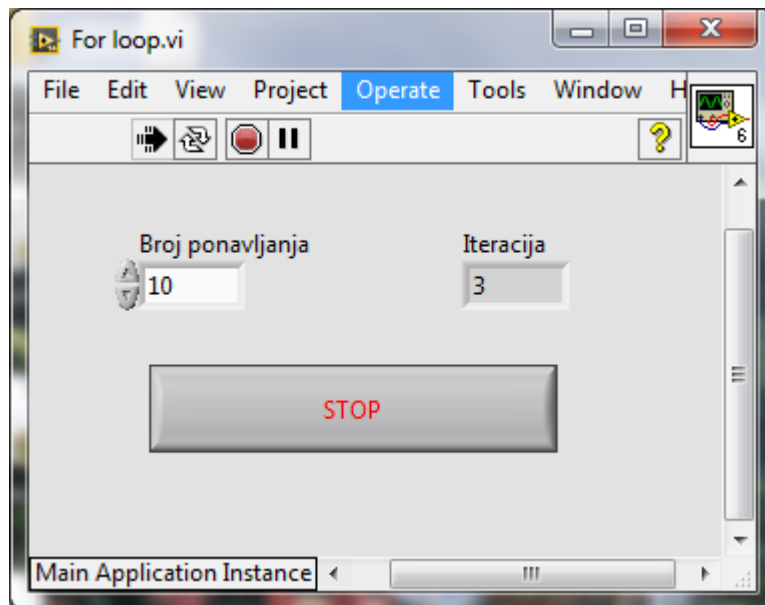
For petlja se izvršava tačno određen broj puta

N – terminal broja izvršavanja

i – terminal iteracija (počinje od 0)

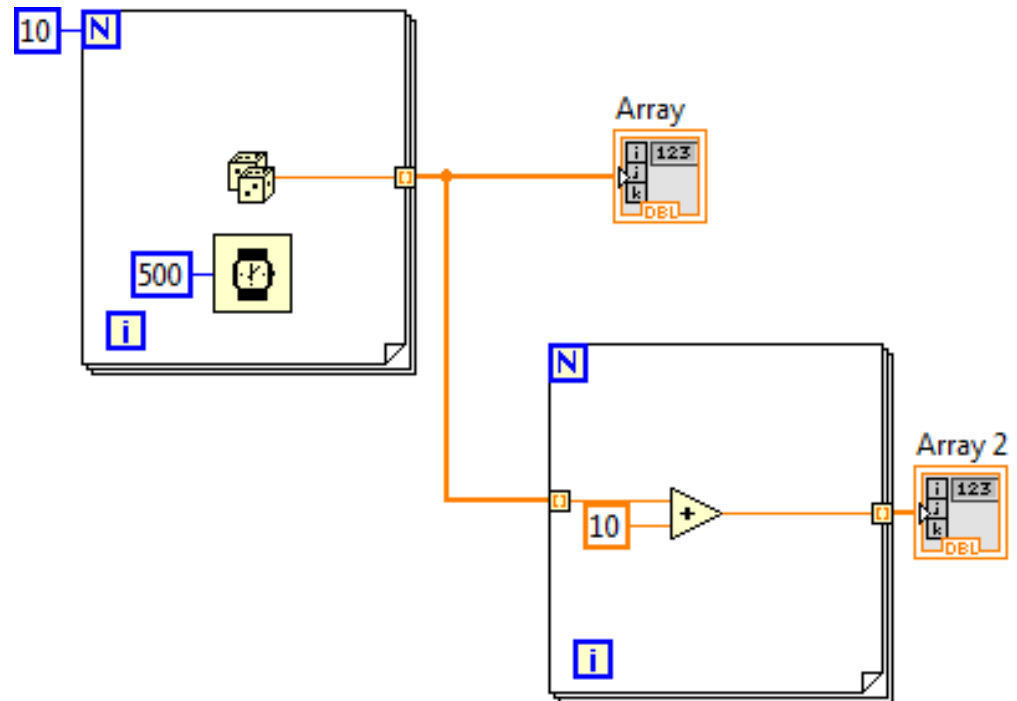
For petlje

- Prebrojiva petlja
- Moguće dodati uslovni završetak



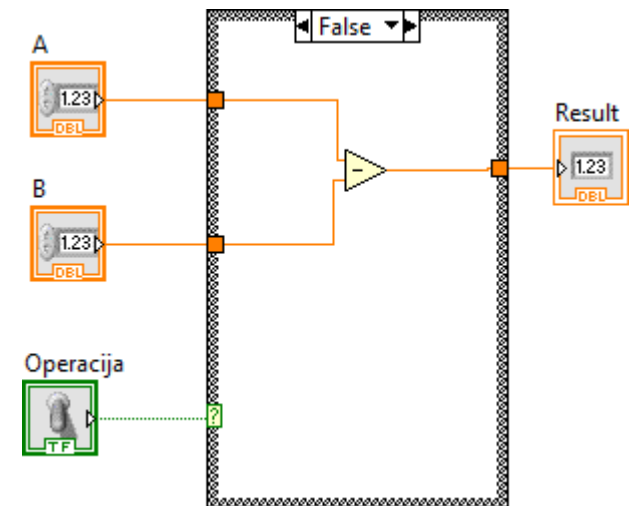
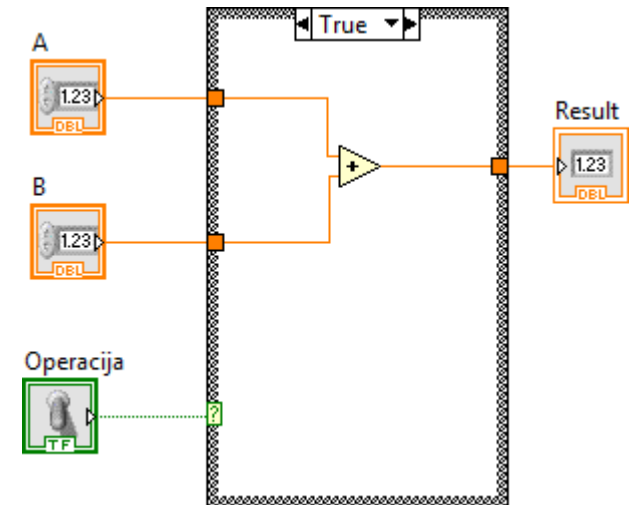
Automatsko indeksiranje petlje

- Ukoliko je potrebno formirati niz izlaznih podataka
- Moguće je uključiti ili isključiti automatsko indeksiranje
- Primenljivo na while i for petlje
- For – podrazumevano uključeno; While - isključeno



Case struktura

- Ekvivalent uslovnom izvršavanju
- if *USLOV1*
... procedura1
elsif *USLOV2*
... procedura 2
...
end



Case struktura

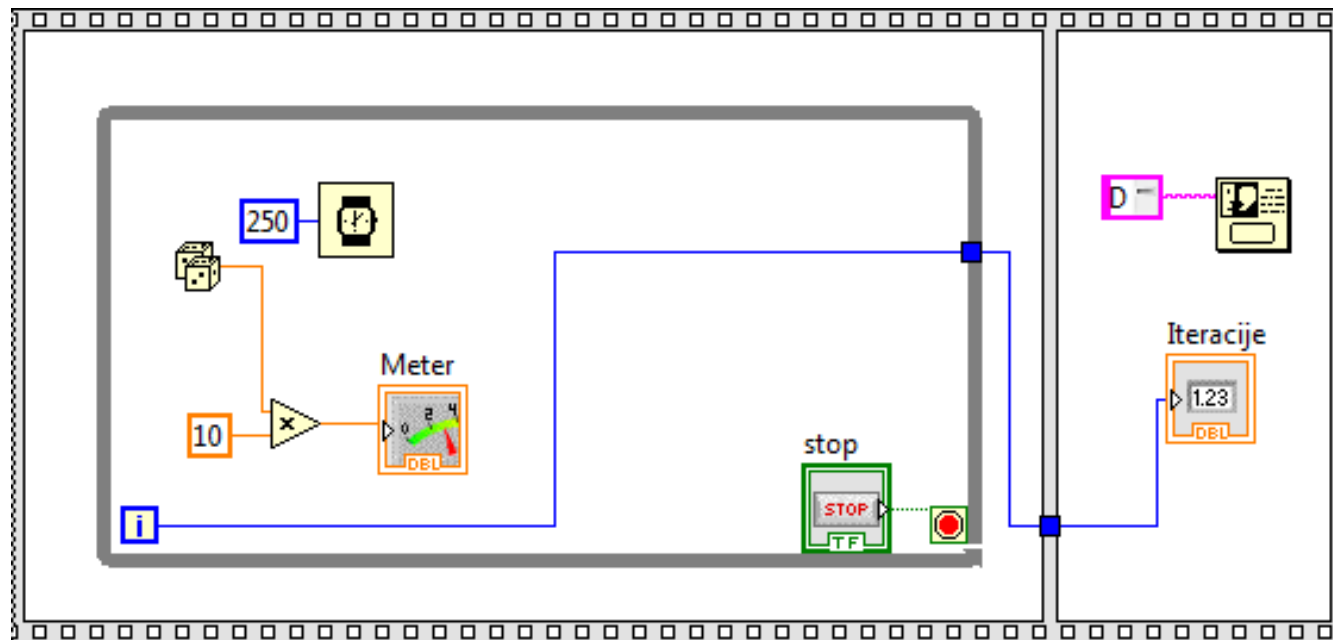
- Može se kreirati više ulaznih i izlaznih tunela za Case strukturu.
- Ulazi su dostupni svim case-ovima, ali case-ovi ne moraju da koriste svaki ulaz.
- Mora se definisati izlazni tunel za svaki case.
- Selektor može biti tipa bool, numeric, string, array...
- Case struktura može da ima *default* case, a ne mora, ali onda se moraju navesti svi mogući slučajevi.

Event struktura

- Slična Case strukturi, ali je upravljana događajima.
- Događaj može biti bool, numeric, string,...
- Čeka (timeout) sve dok se ne desi neki od predviđenih događaja
- Timeout se postavlja na terminal u levom gornjem uglu (default=-1)
- Može imati više događaja

Sequence struktura

- Uređuje redosled izvršavanja koda
- Flat sequence



Sequence struktura

- Uređuje redosled izvršavanja koda
- Stacked sequence

