

# Programski jezici i strukture podataka

10

# Stek

- Stek je mehanizam za prihvatanje podataka.
- Stek je linearni tip podataka koji se ponaša kao memorijski mehanizam tipa „poslednji-u, prvi-iz“ (na engleskom "last-in, first-out" - LIFO).
- Najčešće korišćena analogija za stek u svakodnevnom životu je gomila tanjira koja se može naci u vecini restorana. Poslednji tanjir stavljen na vrh gomile uzima se prvi.

# Stek

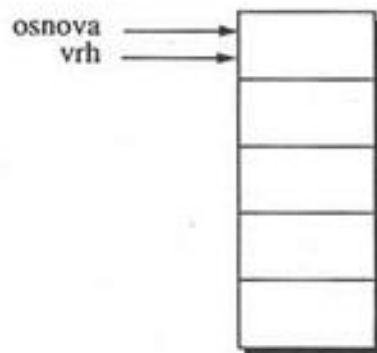
- Savremeni računari su mašine orijentisane na rad sa stekom.
- Stekovi se koriste u razne svrhe, kao što su:
  - čuvanje stanja mašine za vreme dok prekidne rutine opslužuju zahteve višeg prioriteta,
  - predaja parametara funkcijama,
  - vođenje evidencije o preklapajućim prozorima u interfejsima tako dalje.

# Stek

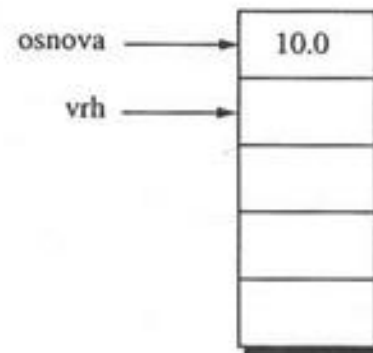
- Struktura steka je linearna.
- Dozvoljen je pristup samo prvom elementu.
- Dodati se može samo ispred prvog elementa.
- Funkcije za rad sa stekom:
  - push
  - pop

### STEK (poslednji-u, prvi-iz)

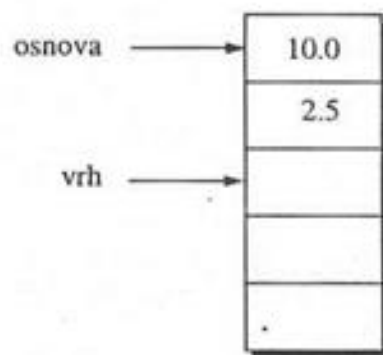
a) Prazan stek



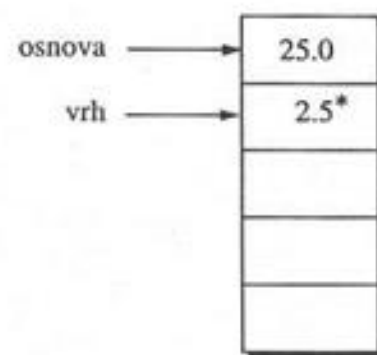
b) Posle push(10.0)



c) Posle push(2.5)



d) Posle push(pop()\*pop())



\*Vrednost ostaje na steku, ali će pri sledećem pozivu funkcije push() biti prepisana novim sadržajem.

# Tip podataka - stek

- Stek ima osobinu tipa podataka.
- Realizacija steka može biti sekvencijalna ili spregnuta.
- Najjednostavnija implementacija steka je pomoću niza i pratećih pokazivača ili indeksa koji pokazuju gde podatak treba da se upiše i odakle da se pročita, sekvencijalna realizacija.

**(z39.c)**

# Spregnuta realizacija steka

- Spregnuta realizacija steka celobrojnih vrednosti upotrebom jednostruko spregnute liste. (**z40.c**)

**SEKVENCA**

# Definicija

- Sekvenca se definiše kao uređeni par:

$$D=(S(D),r(D))$$

sa sledećim osobinama:

- struktura je linearna
- dozvoljen je pristup svakom elementu
- ukloniti se mogu samo svi elementi odjednom
- novi se dodaje na kraj

# Upotreba sekvence

- Sekvenca se koristi kao mehanizam organizovanja podataka na masovnim, sekundarnim, memorijama.
- Recimo film se smešta kao sekvenca blokova.
- Prilikom pisanja, novi blok se dodaje iza poslednjeg.
- Brisanje, ako je moguće, obavlja se odjednom nad svim elementima, a pristupa se od bloka do bloka.

# Opreacije nad elementima sekvence

- **Pristup** je moguć bilo kom elementu sekvence, ali krećući se od elementa kome se prethodno pristupalo. (ako se prethodno pristupalo trećem elementu a sada je potrebno pristupiti 25. elementu, obaviće se sekvencijalni prolazak 4., 5., ..., 23. i 24. elementa da bi se stiglo do 25. elementa, slično kretanju kabine lifta).
- **Uklanjanje** je moguće samo nad svim elementima sekvence odjednom, brisanje sekvence.
- **Dodavanje elementa** je moguće na kraju sekvence, iza poslednjeg elementa.