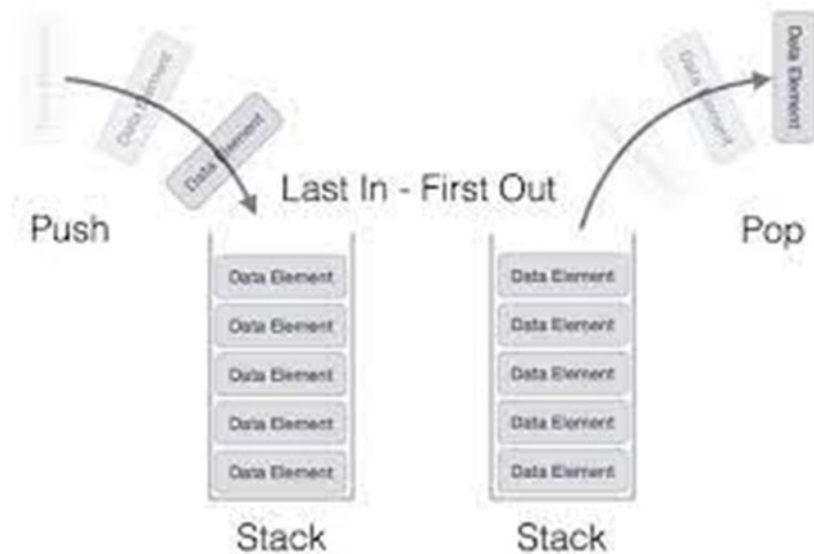


# Stek struktura podataka

**Stek** je linearna struktura podataka koja prati specifičan protokol prilikom izvršavanja operacija. Protokol je da se operacije moraju izvršavati po redu **LIFO**(Poslednji Došao Prvi Otišao eng. *Last In First Out*) ili **FILO**(Prvi Došao Poslednji Otišao eng. *First In Last Out*).

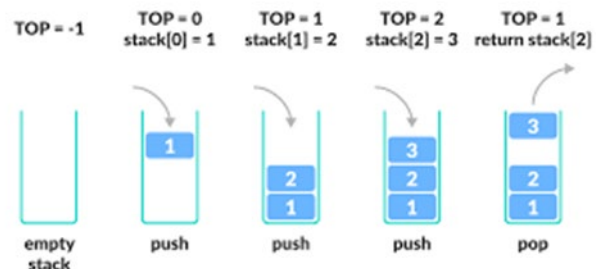


U životu možemo naći mnogo primera steka. Uzmimo primer tanjira poredanih jedan na drugi. Tanjir koji je na vrhu je prvi koji će biti uklonjen, tj. tanjir koji je na dnu ostaje u “*steku*” duže od svih ostalih. Dakle, u ovom slučaju možemo videti praćenje reda LIFO/FILO.

Stek može biti **fiksne** veličine, ili biti **dinamično** implementiran gde se veličina steka može menjati. U slučaju steka ograničene veličine, ako dođe do pokušaja dodavanja još elemenata u pun stek, to dovodi do **greške prekoračenja steka** (eng. *Stack Overflow*). Slično tako, u slučaju da se operacija izbacivanja elementa iz steka (eng. *pop*) vrši nad praznim stekom, to takođe dovodi do **greške prekoračenja steka** (eng. *Stack Underflow*).

## Operacije nad stekom

Stek predstavlja sekvencijalnu reprezentaciju elemenata u formatu linearne strukture podataka. Stek se sastoji iz **ograničenog dna**, a sve operacije se obavljaju na **vrhu**. Kada god je novi element dodat u stek preko operacije **umetanja** (eng. *push*), vrednost promenljive **top** koja sadrži indeks elementa na vrhu se **inkrementira** za jedan, a kada je



elemenat **izbačen iz steka**(*eng. pop*), vrednost promenljive **top** se **dekrementira** za jedan. Pokazivač na vrh steka je takođe poznat i kao **stek pointer** (*eng. Stack Pointer*).

Stek se smatra ograničenom strukturom podataka kako samo mali broj operacija možemo izvršavati nad njim. Pored operacija umetanja(*eng. push*) i izbacivanja(*eng. pop*), neke implementacije mogu sadržati naprednije operacije kao što su:

- *Peek* – Proveravanje vrednosti elementa sa vrha steka bez njegovog brisanja
- *Duplicate* – Kopiranje vrednosti sa vrha steka u promenljivu i umetanje iste
- *Swap* – Operacija koja menja mesta elementu na vrhu steka i elementu koji je odmah ispod njega
- *Rotate* – Premestiti određen broj elemenata sa vrha steka ili ga rotirati

Softverska implementacija koncepta steka se izvršava u vidu **niza** ili **ulančane liste** gde je vrh praćen sa **promenljivom**, u slučaju niza, ili sa **pokazivačem**, u slučaju ulančane liste. Mnogi programski jezici imaju već ugrađene mogućnosti koje pomažu implementaciju steka.

Hardverski stekovi su implementirani u svrsi alokacije memorije i pristup istoj koristeći ograničen izvor i veličinu. U ovom slučaju se za čuvanje stek pointera koriste stek registri.