

STEK

Stek (engl. stack) je dinamička struktura podataka koja podržava operacije dodavanja i brisanja elemenata. Posebnost tih operacija je što se elementi steka dodaju ili oduzimaju isključivo s vrha stoga. Taj pristup je poznat kao Last-In-First-Out – LIFO. Veliku primenu nalazi kao struktura podataka u procesorskim jedinicama (CPU) jer omogućuje izvršavanje funkcijskih poziva.

Primena steka

Postoji nekoliko važnih primena steka, od koje ističemo primjenu u implementaciji funkcijskih poziva unutar računala. U arhitekturi računala, stek je struktura podataka koja omogućava spremanje informacija o aktivnim subrutinama (funkcijama) unutar računarskog programa kojeg često zovemo stek poziva ili kontrolni stek. Upravljenje tim stekom je najčešće automatizirano i sakriveno u jezicima višeg nivoa.

Implementacija steka

Stek se može implementovati na dva načina:

- statička implementacija (upotrebom polja)
- dinamička implementacija (upotrebom povezane liste)

Statička implementacija koristi polje za spremanje stek elemenata. Postupak je jednostavan i fleksibilan, ali je problem što se unapred mora odrediti veličina steka. Kako bi se to zaobišlo može se koristiti dinamička implementacija primenom pokazivača, odnosno uz pomoć povezane liste.

Metoda za rad sa stekom:

- empty – proverava da li je stek prazan
- size – veličina steka
- top - pokazivač na vrh steka
- push – dodavanje elementa u stek
- pop - brisanje elementa iz stek

Primer koda za prikazivanje elemenata steka:

```
#include< iostream >
#include< stack >
usingnamespacestd ;
int main ()
{
    stack mystack ;
    for(inti =0; i <<"Brisanje steka...";
    while(! mystack . empty () )
    {
        cout <<" "<< mystack . top () ;
        mystack . pop () ;
    }
    cout << endl ;
    return0;
}
```