



# STRUKTURE PODATAKA

## PROLEĆNI SEMESTAR 2023/2024

# UVOD

*Prof.dr Leonid Stoimenov*  
*Računarstvo i informatika*  
*Katedra za računarstvo*  
*Elektronski fakultet u Nišu*

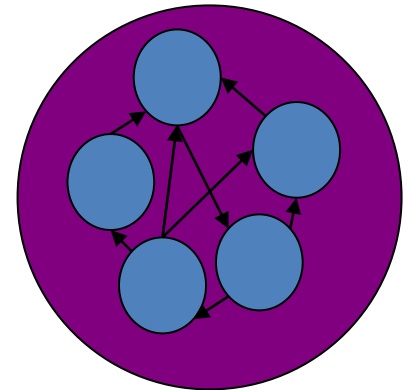
# UVOD - PREGLED

- Definicija i pregled struktura podataka
- Strukture podataka u SWE
- Kategorizacija struktura podataka i svojstva klasa struktura
- Pseudo kod
- Analiza algoritama



# DEFINICIJA STRUKTURE I PODATKA

- Šta je **struktura**?
  - Struktura je unutrašnji raspored, sklop, povezanost sastavnih delova neke celine
- Šta je **podatak**?
  - Podatak je kolekcija činjenica, pojmova ili događaja predstavljena na unapred dogovoreni način



# DEFINICIJA STRUKTURE PODATAKA

- **Struktura podataka** je kolekcija elemenata podataka među kojima postoje određene relacije [Knuth, p.289]
  - ovim relacijama se podaci iz kolekcije podataka povezuju u strukturu podataka
- **Struktura podataka** je sistematski način organizovanja i pristupa podacima [Goodrich 2004, p.108]



# KOLEKCIJA STRUKTURA PODATAKA

- Liste
  - Niz (String)
  - Polje (Array)
  - Lančana lista (Linked list)
- Asocijativna polja (rečnici ili mape)
- Tablice
- Red (Queue, FIFO)
- Magacin (Stack, LIFO)
- Dvostrani red (Deque)
- Datoteke (Files)
- Baze podataka (Databases)
- Stablo (Tree)
  - Binarno
  - Binarno ulančano stablo
  - Binarno stablo traženja
  - Stablo traženja po M putanja
  - B - stablo
  - B+ - stablo
  - B\* - stablo
  - Balansirano stablo
  - Gomila (Heap)
  - Opšte stablo
  - N-arna stabla
- Graf (Graph)
  - Orijentisan
  - Neorijentisan

Zašto ima mnogo struktura podataka?

# KATEGORIZACIJA

## STRUKTURA PODATAKA

- (K1) Zavisnost od fizičkog objekta (implementacije)
  - **Apstraktne (logičke):** nezavisne od fizičkog objekta
  - **Memorijske (fizičke):** zavisne od fizičkog objekta
- (K2) Broj neposrednih prethodnika i sledbenika
  - **Linearne:** jedan prethodnik, jedan sledbenik
  - **Nelinearne:** jedan ili više prethodnika, jedan ili više sledbenika
    - **Hijerarhijske:** jedan prethodnik, više sledbenika
    - **Mrežne:** više prethodnika, više sledbenika
- (K3) Postojanost objekata i veza u toku života strukture
  - **Statičke:** ne menja se
  - **Dinamičke:** menja se



# APSTRAKTNE (LOGIČKE) STRUKTURE PODATAKA

- **Apstraktna struktura podataka** je apstraktno skladište podataka definisano
  - skupom operacija koje se mogu izvršiti nad podacima
  - i **složenošću izračunavanja** tih operacija,
  - bez uzimanja u obzir konkretne implementacije strukture podataka
- Ovaj pojam je vrlo blizak pojmu **Apstraktni tip podataka (Abstract Data Type – ADT)** u programskim jezicima



# MEMORIJSKE (FIZIČKE) STRUKTURE PODATAKA

- Kada se podaci i veze apstraktne (logičke) strukture podataka smeste na konkretni memorijski medijum dobija se **memorijska (fizička) struktura podataka**
- Veze apstraktne i memorijske strukture podataka
  - Jedna apstraktna struktura podataka **može imati više** memorijskih reprezentacija, tj. više memorijskih struktura podataka
  - Jedna memorijska struktura uvek se preslikava samo u **jednu** apstraktnu strukturu





# STATIČKE I DINAMIČKE STRUKTURE PODATAKA

- Struktura podataka je **statička** ako su svojstva strukture podataka fiksna (zadata u deklaraciji strukture) i ne mogu se menjati u toku izvršenja programa
- Struktura podataka je **dinamička** ako se svojstva strukture podataka mogu menjati u toku izvršenja programa



# ULOGA STRUKTURA PODATAKA U RAZVOJU SOFTVERA

- U razvoju softvera strukture podataka imaju važnu ulogu
- **Važno je uočiti:**
  - Da su za različite probleme pogodne različite strukture podataka
  - Da su neke strukture podataka prilagođene za rešavanje određenih problema



# IZBOR STRUKTURE PODATAKA (1)

- Izbor strukture podataka je **jedan korak** u razvoju softvera
- Izabrana struktura podataka **bitno utiče na efikasnost** implementacije algoritama za rešavanje konkretnih problema
- Nakon izbora strukture podataka algoritmi obrade su predefinisani
- Lista sa koje se može birati struktura podataka je duga
- Za dobar izbor je neophodno dobro poznavanje struktura podataka



# IZBOR STRUKTURE PODATAKA (2)

- Izbor strukture podataka za konkretni problem obično počinje izborom **apstraktne strukture podataka**
- Dobro izabrana struktura podataka dozvoljava da se kritične operacije izvršavaju optimalnim korišćenjem resursa, kao što su vreme izvršenja i memorijski prostor
- U uži izbor može ući veći broj struktura podataka
- **Tek nakon analize algoritama** koji se pridružuju izabranim strukturama podataka može se doneti **konačna odluka** o strukturi podataka koja će se koristiti



# ALATI ZA OPIS I IMPLEMENTACIJU ALGORITAMA

- Notacija za opis algoritama
  - Pseudokod
- Implementacija algoritama
  - C++
  - C



# PSEUDOKOD

## PODRAZUMEVANA NOTACIJA ZA OPIS ALGORITMA

- Opis algoritma visokog nivoa
- Strukturniji od govornog jezika
- Manje detaljan od programskog jezika
- Ima osnovne algoritamske strukture
- Skriva detalje projektovanja i implementacije



# PSEUDOKOD - DETALJI

## ○ Kontrola toka

- `if(...) then... [else ...] [endif]`
- `while(...) ...endwhile`
- `repeat... until(...)`
- `for ...endfor`
- Uvlačenje linija koda može zameniti velike zagrade

## ○ Izrazi

- $\leftarrow$  ili `=` dodela (kao u C++)
- `=` testiranje jednakosti (`==` u C++)
- dozvoljene matematičke funkcije



# PSEUDOKOD - PRIMER

**Primer:** Pseudokod funkcije (metode) **arrayMax(A,n)** koja vraća maksimalni element polja A od **n** elemenata

## Algoritam arrayMax(A,n)

**Ulaz:** polje A od n celih brojeva

**Izlaz:** maksimalni element **tekMax** u A

1.     **tekMax**  $\leftarrow$  A[0]
2.     **for** i  $\leftarrow$  1, n - 1
3.         **if** (A[i] > **tekMax**) **then**
4.             **tekMax**  $\leftarrow$  A[i]
5.     **endfor**
6.     **return** **tekMax**

**call arrayMax(A,n)** // poziv procedure arrayMax





# PSEDOKOD I C++

- Pseudokod

## Algoritam

**arrayMax(A,n)**

**Ulaz:** polje **A** od **n** celih brojeva

**Izlaz:** maksimalni element u **A**

**tekMax**  $\leftarrow$  **A**[0]

**for** **i**  $\leftarrow$  1, **n** - 1

**if** (**A**[**i**] > **tekMax**) **then**

**tekMax**  $\leftarrow$  **A**[**i**]

**endfor**

**return** **tekMax**

- C++

```
int arrayMax(int A[ ], int n)
{

    int tekMax = A[0];
    for (int i = 1; i < n; i++)
        if (A[i] > tekMax)
            tekMax = A[i];
    return tekMax;
}
```

# PITANJA, IDEJE, KOMENTARI

