

\$ OOP 02

\$ Nizovi i matrice

Name: Ognjen Milinković<sup>†</sup>

Date: 8. mart 2021.

---

<sup>†</sup>ognjen\_milinkovic@matf.bg.ac.rs

## § Sadržaj

### 1. Nizovi

Struktura niz

Nizovi u javi - primeri

`java.util.Arrays`

### 2. Matrice

Realizacija matrice

Matrice u javi - primeri

## 1. Nizovi

Struktura niz

Nizovi u javi - primeri

java.util.Arrays

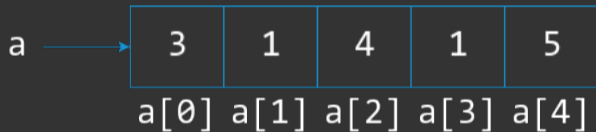
## 2. Matrice

Realizacija matrice

Matrice u javi - primeri

## § Šta je niz?

- > Struktura podataka
- > Omogućava slučajan pristup (eng. random access)
- > Susedni elementi niza se u memoriji nalaze na susednim pozicijama



## § Dobre strane

- > Pristup proizvoljnom elementu
- > Lako se kešira
- > Iteracija kroz strukturu je izuzetno brza (keširanje i lokalnost suseda)

- > Zahteva uniforman blok memorije
  - \* Nekada teško naći usled fragmentacije memorije
- > Operacije dodavanja<sup>1</sup> i brisanja su skupe
  1. Alokacija (potencijalno velikog) bloka memorije
  2. Kopiranje originalnog niza u novi blok
  3. Primena traženog ažuriranja

---

<sup>1</sup>Možemo implementirati vremenski efikasnije uz alokaciju većeg prostora

## § Adresiranje

> Najčešće sintaksa za niz a za element na poziciji i izgleda:

```
int x = a[i]
```

> operator [] računa adresu elementa na poziciji i koji potom vraća

> C++ omogućava da definišemo sopstveni operator indeksiranja

## § Adresiranje - kako izračunati adresu?

Neka je:

- > `iaddr` - adresa `i`-tog elementa koji želimo
- > `a` - adresa početka niza
- > `i` - indeks elementa koji želimo (indeksiramo od 0)
- > `T` - tip elemenata koji su u nizu
- > `sizeof` - funkcija ili operator koji nam vraća veličinu tipa `T` u bajtovima

Onda slučajni pristup elementu dobijamo računanjem po formuli:

- >  $iaddr = a + i * sizeof(T)$



## § Pravljenje niza

U javi postoje dva načina da se napravi niz dužine  $n$

1. `int[] a = new int[n]`

2. `int a[] = new int[n]`

## § Inicijalizacija niza

> Niz možemo inicijalizovati na sledeći način:

```
int[] a = {10, 20, 42, 99};
```

## § Podrazumevana vrednost - Java standard

Java standard kaže:

- > Each class variable, instance variable, or array component is initialized with a default value when it is created (§15.9, §15.10).
- > For type int, the default value is zero, that is, 0.

Više o temi na adresi:

- > [docs.oracle.com/javase/specs/jls/se7/html/jls-4.html#jls-4.12.5](https://docs.oracle.com/javase/specs/jls/se7/html/jls-4.html#jls-4.12.5)

## § Primer 1

```
public static void main(String[] args) {  
    int[] a = new int[4];  
    a[0] = 10;  
    a[1] = 20;  
    a[2] = 42;  
    a[3] = 99;  
    System.out.println("Duzina niza: " + a.length);  
}
```

## § Primer 2

```
public static void main(String[] args) {  
    int[] a = {10, 20, 42, 99};  
    System.out.println("Duzina niza: " + a.length);  
}
```

## § Ispis niza - for

> Možemo koristiti standardnu for petlju

> Dužinu niza dobijamo sa `a.length`

```
public static void main(String[] args) {  
    int[] a = {10, 20, 42, 99};  
  
    for (int i = 0; i < a.length; i++)  
        System.out.print(a[i] + " ");  
  
    System.out.println();  
}
```

## § Ispis niza - foreach

- > Java poseduje i foreach petlju (JDK 1.5)
- > Koristi ključnu reč `for`
- > Oslobađa nas indeksiranja i brige o dužini strukture
- > Indeksirajuće `for` petlje se retko koriste u praksi

```
public static void main(String[] args) {  
    int[] a = {10, 20, 42, 99};  
  
    for (int element: a)  
        System.out.print(element + " ");  
  
    System.out.println();  
}
```

## § Prenos niza funkciji

```
static double sumiraj(double[] a) {
    double sum = 0;
    for (int i = 0; i < a.length; i++)
        sum += a[i];
    // for (double element: a)
    //     sum += element;
    return sum;
}

public static void main(String[] args) {
    double[] a = {10.1, 23.5, 42.42, 99};
    System.out.println("Suma: " + sumiraj(a));
}
```



## § Vraćanje niza iz funkcije

```
static int[] obrni(int[] a) {  
    int[] tmp = new int[a.length];  
  
    for (int i = 0; i < a.length; i++)  
        tmp[i] = a[a.length-1-i];  
  
    return tmp;  
}
```

```
public static void main(String[] args) {  
    int[] a = {10, 20, 42, 99};  
  
    int[] b = obrni(a);  
    for (int e: a)  
        System.out.print(e + " ");
```

```
System.out.println();
```

## § java.util.Arrays

- > Sadrži veliki broj statičkih funkcija za rad sa nizovima
- > Funkcije za sortiranje, pretragu, poređenje...
- > Pogledati okačene primere

```
public static void main(String[] args) {  
    int[] a = {3, 1, 4, 1, 5, 9, 2, 6, 5};  
    Arrays.sort(a);  
    for (int e: a)  
        System.out.print(e + " ");  
    System.out.println();  
}
```

## § Pročitati

- > [www.tutorialspoint.com/java/java\\_arrays.htm](http://www.tutorialspoint.com/java/java_arrays.htm)
- > [www.tutorialspoint.com/java/util/java\\_util\\_arrays.htm](http://www.tutorialspoint.com/java/util/java_util_arrays.htm)

## 1. Nizovi

Struktura niz

Nizovi u javi - primeri

java.util.Arrays

## 2. Matrice

Realizacija matrice

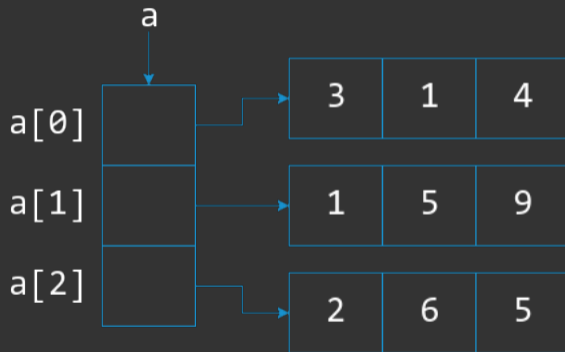
Matrice u javi - primeri

§ Matrix



## § Matrice

> Realizujemo kao niz nizova<sup>2</sup>

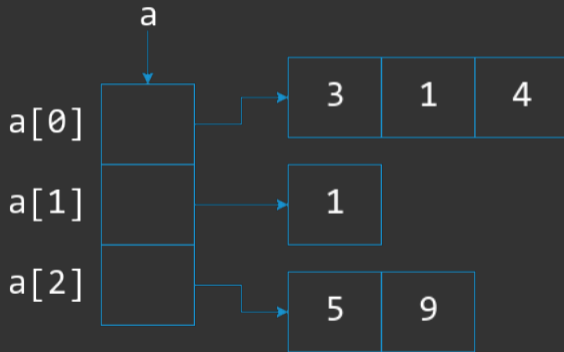


---

<sup>2</sup>Da li vidimo neku pravilnost u brojevima?

## § Matrice

- > Možemo praviti zabavne konstrukcije - npr. testerasti niz (eng. jagged array)



## § Pitanja

U primeru sa prethodnog slajda:

> Šta označava  $a[0]$ ?



## § Pitanja

U primeru sa prethodnog slajda:

> Šta označava a[0]?

> // Niz dužine 3

## § Pitanja

U primeru sa prethodnog slajda:

> Šta označava `a[0]`?

> `// Niz dužine 3`

> Šta označava `a[0][2]`?

## § Pitanja

U primeru sa prethodnog slajda:

> Šta označava a[0]?

> // Niz dužine 3

> Šta označava a[0][2]?

> // int vrednosti 4

## § Pitanja

U primeru sa prethodnog slajda:

> Šta označava `a[0]`?

> `// Niz dužine 3`

> Šta označava `a[0][2]`?

> `// int vrednosti 4`

> Šta označava `a.length`?

## § Pitanja

U primeru sa prethodnog slajda:

> Šta označava `a[0]`?

> `// Niz dužine 3`

> Šta označava `a[0][2]`?

> `// int vrednosti 4`

> Šta označava `a.length`?

> `// broj vrsti matrice a koji je jednak 3`

## § Pitanja

U primeru sa prethodnog slajda:

> Šta označava `a[0]`?

> `// Niz dužine 3`

> Šta označava `a[0][2]`?

> `// int vrednosti 4`

> Šta označava `a.length`?

> `// broj vrsti matrice a koji je jednak 3`

> Šta označava `a[2].length`?

## § Pitanja

U primeru sa prethodnog slajda:

> Šta označava `a[0]`?

> `// Niz dužine 3`

> Šta označava `a[0][2]`?

> `// int vrednosti 4`

> Šta označava `a.length`?

> `// broj vrsti matrice a koji je jednak 3`

> Šta označava `a[2].length`?

> `// broj kolona u vrsti 2 matrice a koji je jednak 2`

## § Pravljenje matrice v1

```
public static void main(String[] args) {
    int[][] a = new int[3][3];
    a[0][0] = 3; a[0][1] = 1; a[0][2] = 4;
    a[1][0] = 1; a[1][1] = 5; a[1][2] = 9;
    a[2][0] = 2; a[2][1] = 6; a[2][2] = 5;

    for (int i = 0; i < a.length; i++) {
        for (int j = 0; j < a[i].length; j++)
            System.out.print(a[i][j] + " ");
        System.out.println();
    }
}
```



## § Pravljenje matrice v2

```
public static void main(String[] args) {
    int[][] a = {
        {3, 1, 4},
        {1, 5, 9},
        {2, 6, 5}
    };

    for (int i = 0; i < a.length; i++) {
        for (int j = 0; j < a[i].length; j++)
            System.out.print(a[i][j] + " ");
        System.out.println();
    }
}
```

## § Prenos u funkciju

```
static void ispisi(int[] [] a) {
    for (int i = 0; i < a.length; i++) {
        for (int j = 0; j < a[i].length; j++)
            System.out.print(a[i][j] + " ");
        System.out.println();
    }
}

public static void main(String[] args) {
    int[] [] a = { {3, 1, 4}, {1, 5, 9}, {2, 6, 5} };
    ispisi(a);
}
```

## § Vraćanje iz funkcije

```
static int[][] generisi() {  
    int[][] tmp = { {3, 1, 4}, {1, 5, 9}, {2, 6, 5} };  
    return tmp;  
}
```

```
public static void main(String[] args) {  
    int[][] a = generisi();  
    ispisi(a); // sa prethodnog slajda  
}
```

## Zahvalnica

Ovu prezentaciju je napravio kolega Nemanja Mićović.  
Hvala kolegi Mićoviću što se saglasio da koristimo ovu prezentaciju.