



Funded by the
Erasmus+ Programme
of the European Union



This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



KI103 - JAVA 1: OSNOVE PROGRAMIRANJA U JAVI

Višedimenzionalni nizovi

Lekcija 08

PRIRUČNIK ZA STUDENTE

KI103 - JAVA 1: OSNOVE PROGRAMIRANJA U JAVI

Lekcija 08

VIŠEDIMENZIONALNI NIZOVI

- ✓ Višedimenzionalni nizovi
- ✓ Poglavlje 1: Višedimenzionalni nizovi
- ✓ Poglavlje 2: Domaći zadaci
- ✓ Zaključak

Copyright © 2017 – UNIVERZITET METROPOLITAN, Beograd. Sva prava zadržana. Bez prethodne pismene dozvole od strane Univerziteta METROPOLITAN zabranjena je reprodukcija, transfer, distribucija ili memorisanje nekog dela ili čitavih sadržaja ovog dokumenta., kopiranjem, snimanjem, elektronskim putem, skeniranjem ili na bilo koji drugi način.

Copyright © 2017 BELGRADE METROPOLITAN UNIVERSITY. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, without the prior written permission of Belgrade Metropolitan University.

▼ Uvod

UVOD

Jedan niz promenljivi može da sadrži reference ka velikom skupu podataka ili objekata.

U predavanju ćemo se upoznati sa:

- Kreiranjem i radom sa višedimenzionalnim nizovima;
- Upotrebom višedimenzionalnih nizova u Java programima;
- Dvodimenzionalnim nizovima

▼ Poglavlje 1

Višedimenzionalni nizovi

PREDGOVOR

Podatci u nekoj tabeli ili matrici se mogu predstaviti dovodimenzionalnim nizom.

U ovom poglavlju ćete se upoznati sa višedimenzionalnim nizovima, kroz primere korišćenja dvodimenzionalnih nizova.

Izlaganje prate pažljivo birani primeri. Nakon savladavanja primera, akcenat stavite na zadatke za samostalni rad i testove za proveru znanja.

Na kraju lekcije se nalaze domaći zadaci koje bi trebalo da uradite nakon uspešno savladanog gradiva.

▼ 1.1 Kreiranje i rad sa višedimenzionalnim nizovima

DEFINISANJE VIŠEDIMENZIONALNOG NIZA

Dvodimenzionalni nizovi se popularno nazivaju i matrice ili tabele.

Nizovi o kojima smo govorili do sada poseduju samo jednu "dimenziju" - dužinu. Obični nizovi se zato često nazivaju jednodimenzionalni nizovi. U Javi se mogu koristiti i nizovi koji imaju dve "dimenzije" - širinu i visinu, tri "dimenzije" - širinu, visinu i dubinu, i tako dalje. Zato se o ovim nizovima govori kao dvodimenzionalnim, trodimenzionalnim ili višedimenzionalnim nizovima.

U Javi se svaki tip podataka može koristiti za bazni tip nekog (jednodimenzionalnog) niza. Specifično, kako je neki tip niza takođe običan tip u Javi, ukoliko je bazni tip nekog niza baš takav tip, dobija se niz nizova. Na primer, jedan celobrojni niz ima tip **int[]**, a to znači da automatski postoji i tip **int[][]** koji predstavlja niz celobrojnih nizova - ovaj "niz nizova" se kraće zove dvodimenzionalni niz.

Ništa nije neobično da se dalje posmatra i tip **int[][][]** koji predstavlja niz dvodimenzionalnih nizova, odnosno trodimenzionalni niz. Naravno, ova linija razmišljanja se može nastaviti i tako se dobijaju višedimenzionalni nizovi, mada se nizovi čija je dimenzija veća od tri zaista retko primenjuju.

U daljem tekstu predavanja ćemo se baviti samo na dvodimenzionalnim nizovima, jer se svi koncepti u vezi s njima lako proširuju na više dimenzija. Dvodimenzionalni nizovi se popularno nazivaju i matrice ili tabele.

Deklarisanje promenljive tipa dvodimenzionalnog niza je slično deklarisanju jednodimenzionalnog niza - razlika je samo u dodatnom paru uglastih zagrada, na primer:

```
int[][] a;
```

deklariše se promenljiva a čiji sadržaj može biti referenca na objekat dvodimenzionalnog niza tipa `int[][]`. Konstruisanje aktuelnog dvodimenzionalnog niza se vrši operatorom `new` kao i u slučaju jednodimenzionalnog niza, na primer, naredbom dodele:

```
a = new int[3][4];
```

konstruiše se dvodimenzionalni niz veličine $3 * 4$ i referenca na njega se dodeljuje prethodno deklarisanjoj promenljivoj a. Kao i obično, ove dve posebne naredbe se mogu spojiti u jednu:

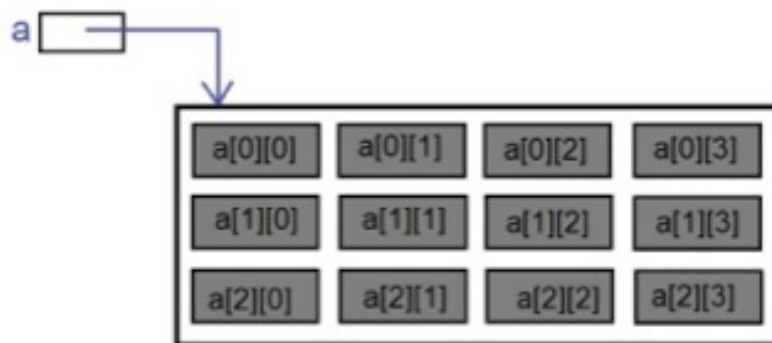
```
int[][] a = new int[3][4];
```

DVODIMENZIONALNI NIZ DIMENZIJA 3X4

Elementi matrice predstavljaju obične, promenljive koje imaju dvostruke indekse `a[i][j]`: prvi indeks pokazuje red i drugi indeks kolonu kojima se element nalazi u matrici.

Slika pokazuje da je jedan dvodimenzionalni niz na koji ukazuje promenljiva a najbolje zamisliti kao matricu (ili tabelu) elemenata koja ima tri reda i četiri kolone.

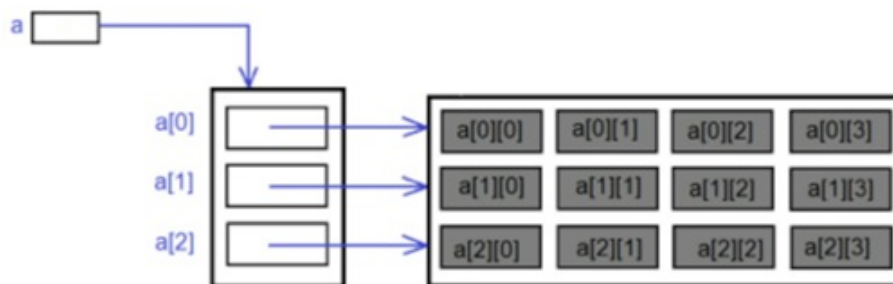
Elementi matrice predstavljaju obične, u ovom slučaju celobrojne promenljive koje imaju dvostruke indekse **`a[i][j]`**: prvi indeks pokazuje red i drugi indeks kolonu kojima se element nalazi u matrici. Pritom treba imati u vidu da numeracija redova i kolona matrice ide od nule, a ne od jedan.



Slika 1.1.1 Primer dvodimenzionalnog niza veličine 3 x 4

Ipak tačnije, dvodimenzionalni niz **a** u prethodnom primeru nije matrica, nego je zaista niz celobrojnih nizova. Tako, izrazom **new**

int[3][4] se zapravo konstruiše niz od tri celobrojna niza, od kojih svaki ima četiri elementa. Pravi izgled dvodimenzionalnog niza **a** je zato onaj koji je prikazan na slici 2::



Slika 1.1.2 Prava slika dvodimenzionalnog niza dimenzija 3 x 4

Prava slika dvodimenzionalnog niza je komplikovanija za razumevanje i, srećom, može se u većini slučajeva zanemariti i dvodimenzionalni niz smatrati matricom elemenata. Ponekad je ipak potrebno znati da svaki red matrice predstavlja zapravo jedan niz za sebe. Ovi nizovi u prethodnom primeru su tipa **int[]** i na njih ukazuju promenljive čija su imena **a[0]**, **a[1]** i **a[2]**. Te promenljive i nizovi se u programu mogu koristiti na svim mestima gde su dozvoljeni obični celobrojni nizovi. Na primer, neki red matrice može biti argument u pozivu metoda čiji je parametar tipa **int[]**.

DEKLARACIJA I INICIJALIZACIJA DVODIMENZIONALNOG NIZA

Dvodimenzionalni nizovi su objekti. Promenljiva su reference na takve objekte

Dvodimenzionalni nizovi su takođe objekti. Promenljiva kao što je bila naša brojPoena su reference na takve objekte. Dvodimenzionalni niz se deklarise na sledeći način:

```
int[][] mojNiz ;
```

Ova deklaracija kaže da se očekuje da će promenljiva **mojNiz** sadržati referencu na 2D niz sa vrednostima tipa **int**. Ako se ništa više ne zada, ova promenljiva je inicijalizovana na **null**.

Deklaracija

```
int[][]mojNiz = new int[3][5] ;
```

kaže da promenljiva **mojNiz** može da sadrži referencu na 2D niz sa vrednostima tipa **int**. Ovim se kreira objekat sa 3 vrste i 5 kolona. Referenca na taj objekat se stavlja u promenljivu **mojNiz**. Svi elementi niza su inicijalizovani na 0.

Deklaracija

```
int[][]mojNiz = { {0,0,0,0,0}, {0,0,0,0,0}, {0,0,0,0,0} };
```

Textbox

radi potpuno isto kao prethodna.

Deklaracija

```
int[][] mojNiz = { {8,1,2,2,9}, {1,9,4,0,3}, {0,3,0,0,7} };
```

kreira niz sa istim dimenzijama (isti broj vrsta i kolona), kao prethodni, ali se elementi inicijalizuju na određene vrednosti.

ODREĐIVANJE BROJA KOLONA

Ako je potrebno dobiti broj kolona matrice, onda je to dužina jednodimenzionalnih nizova od kojih se sastoji svaki red matrice

Zbog prave slike dvodimenzionalnog niza treba imati na umu da vrednost polja **a.length** u prethodnom primeru iznosi tri, odnosno jednaka je broju redova matrice **a**.

Ako je potrebno dobiti broj kolona matrice, onda je to dužina jednodimenzionalnih nizova od kojih se sastoji svaki red matrice, odnosno **a[0].length**, **a[1].length** ili **a[2].length**. Generalno, svi redovi matrice ne moraju biti jednake dužine i u nekim složenijim primenama se koriste matrice sa različitim dužinama redova.

Elementi dvodimenzionalnih nizova se prilikom konstruisanja inicijalizuju odgovarajućim podrazumevanim vrednostima. To se može promeniti pisanjem svih početnih vrednosti iza operatora **new**, slično načinu na koji se to postiže kod jednodimenzionalnih nizova.

Pritom treba voditi računa da se početne vrednosti matrice navode po redovima, odnosno redom za jednodimenzionalne nizove redova matrice u vitičastim zagradama. Na primer:

```
int[][] a = new int[][] {  
    { 1, -1, 0, 0},  
    {10, 17, -2, -3},  
    { 0, 1, 2, 3}};
```

Korišćenje dvodimenzionalnih nizova u programu se, slično jednodimenzionalnom slučaju, zasniva na dvostrukim indeksima pojedinih elemenata matrice. Ovi indeksi mogu biti bilo koji celobrojni izrazi i njihovim izračunavanjem se dobijaju brojevi reda i kolone u kojima se nalazi odgovarajući element matrice.

NIZOVI PROMENLJIVE DUŽINE

Svaka vrsta dvodimenzionalnog niza može imati različit broj elemenata

Svaka vrsta dvodimenzionalnog niza može imati različit broj elemenata. U primeru koji sledi niz **a** ima 3 elementa u prvoj vrsti, 2 u drugoj i 5 u poslednjoj vrsti


```
class Primer{
    public static void main( String[] arg ){
        // deklaracija i inicijalizacija 2D niza
        int[][] a = {{1, 9, 4 },
                     { 0, 2},
                     { 0, 1, 2, 3, 4 } };
        System.out.println("a[0][2] is ", + a[0][2] ) ; // OK
        System.out.println("a[1][1] is ", + a[1][1] ) ; // OK
        System.out.println("a[1][2] is ", + a[1][2] );// GREŠKA!
        a[2][4] = 97; // OK
        a[1][4] = 97; // POGREŠNO!
        int val = a[0][2] ; // OK
        int sum = a[1][2] ; // POGREŠNO!
    }
}
```

Element niza mora postojati da bi se koristio u programu. Ako program pristupa nekom elementu koji ne postoji, onda proverava granica vraća grešku i program se obično prekida.

Kada se govori o veličini dvodimenzionalnog niza, misli se na broj vrsta. Možda ćete pomisliti da bi dužina mogla biti definisana u paru (za vrste i kolone), ali pošto svaka vrsta može imati različit broj elemenata, to se o dužini elemenata ne može govoriti (bar ne o istoj).

```
class Primer2{
    public static void main( String[] arg ){
        // deklaracija i konstrukcija 2D niza
        int[][] a = { { 1, 9, 4 },
                     { 0, 2},
                     { 0, 1, 2, 3, 4 } };
        System.out.println("Dužina je: " + a.length );
    }
}
```

Izlaz iz ovog programa je:

Dužina je: 3

Dužinu svakog podniza koji je element ovog dvodimenzionalnog niza možete pronaći na sledeći način:

```
class Primer3{
    public static void main( String[] arg ) {
        // deklaracija i konstrukcija 2D niza
        int[][] a = { { 1, 9, 4 },
                     { 0, 2},
                     { 0, 1, 2, 3, 4 } };
        // Duzina niza (broj vrsti)
        System.out.println("Duzina niza je: " + a.length );
        // duzina svake vrste (broj njenih kolona)
        System.out.println("Duzina vrste[0] je: " + a[0].length );
        System.out.println("Duzina vrste [1] je: " + a[1].length );
    }
}
```

```
System.out.println("Duzina vrste [2] je: " + a[2].length );  
}
```

IMPLEMENTACIJA DVODIMENZIONALNOG NIZA

Dvodimenzionalni niz se implementira kao niz čiji su elementi jednodimenzionalni nizovi

Dvodimenzionalni niz se implementira kao niz čiji su elementi jednodimenzionalni nizovi. Deklaracija

```
int[][] mojNiz; // 1.
```

deklariše promenljivu **mojNiz**, koja u budućnosti može da ukazuje na objekat niza. U ovom trenutku se ništa ne može reći o broju vrsta i kolona.

Niz sa tri vrste ćete napraviti na sledeći način:

```
mojNiz = new int[3][] ; // 2.
```

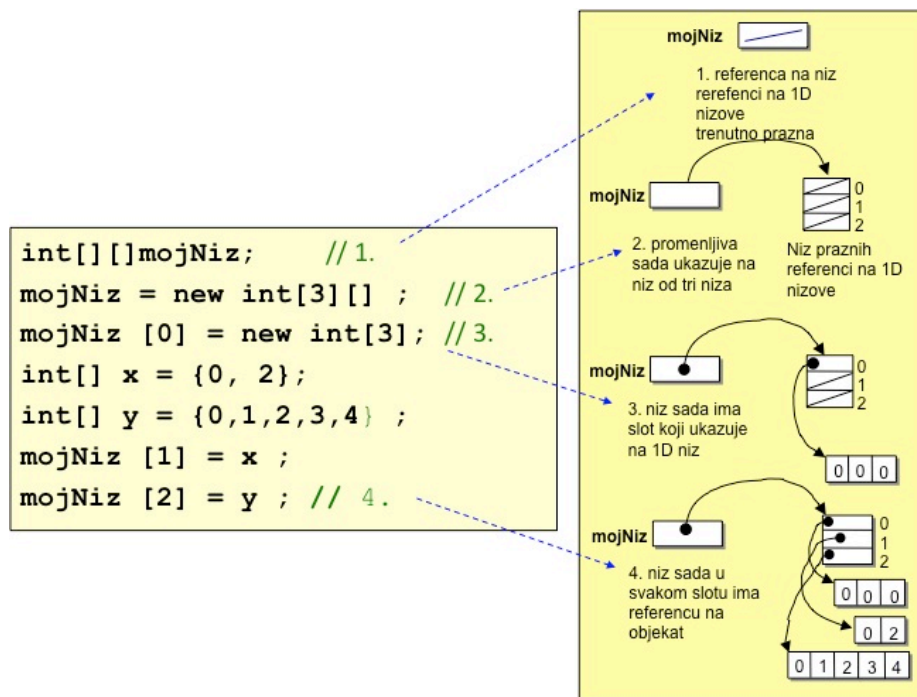
Sada **mojNiz** ukazuje na objekat niza. Ovaj objekat ima tri elementa. Svaki elemenat može (u budućnosti) da ukazuje na niz celih brojeva, odnosno na objekat tipa **int[]**. U ovom trenutku nijedan od tih elemenata ne ukazuje na objekat. Svi su oni inicijalizovani na **null**.

Jedan od načina da se napravi vrsta sa indeksom 0 je:

```
mojNiz [0] = new int[3] ; // 3.
```

Ovim se pravi jednodimenzionalni niz i referenca na njega se stavlja u prvi element niza **mojNiz**. Elementi ovog niza su inicijalizovani na 0.

Vrsti se da se može dodeliti i jednodimenzionalni niz koji je prethodno napravljen (slika 3)



Slika 1.1.3 Proces kreiranja i inicijalizacije jednodimenzionalnog niza

PRIMER 1 - UPOTREBA DVODIMENZIONALNOG NIZA

U radu sa dvodimenzionalnim nizovima se često koriste ugnježdene (unutrašnje) for petlje

U radu sa dvodimenzionalnim nizovima se često koriste ugnježdene **for** petlje tako da se u spoljašnjoj petlji prate redovi matrice, a u unutrašnjoj petlji se prate kolone matrice. Na taj način se za svaki aktuelni red matrice obrađuju svi elementi tog reda, a kako se ovo ponavlja za sve redove matrice, time se obrađuju baš svi elementi matrice red po red.

Na primer, ukoliko za matricu:

```
double[][] a = new double[10][10];
```

treba elementima te matrice na glavnoj dijagonali dodeliti vrednost 1 i ostalim elementima vrednost 0, to se može uraditi na sledeći način:

```
// za svaki red u matrici
for (int i = 0; i < a.length; i++) {
    // i za svaku kolonu u aktuelnom redu
    for (int j = 0; j < a[i].length; j++) {
        // da li je element na glavnoj dijagonali?
        if (i == j)
            a[i][j] = 1;
        else
            a[i][j] = 0;
    }
}
```

Na sličan način se mogu sabrati svi elementi matrice **a**:

```
double zbir = 0;
for (int i = 0; i < a.length; i++)
    for (int j = 0; j < a[i].length; j++)
        zbir = zbir + a[i][j];
```

Sabiranje svih elemenata matrice **a** može se postići i ugnježđenim for-each petljama:

```
double zbir = 0;
for (double[] red : a)
    for (double elem : red)
        zbir = zbir + elem;
```

PRIMER 2

Zbir i aritmetička sredina članova dvodimenzionalnog niza.

Tekst zadatka:

Napisati program koji učitava brojeve **n** i **m** sa standardnog ulaza, zatim učitava **n** redova i **m** kolona dvodimenzionalnog niza i ispisuje sumu i aritmetičku sredinu svih elemenata dvodimenzionalnog niza. Program treba da bude deo NetBeans projekta pod nazivom KI103-L07. Naziv klase treba da bude Primer2. Za učitavanje podataka od korisnika treba koristiti objekat klase **Scanner**, a za ispis podataka objekat **System.out**.

Analiza problema:

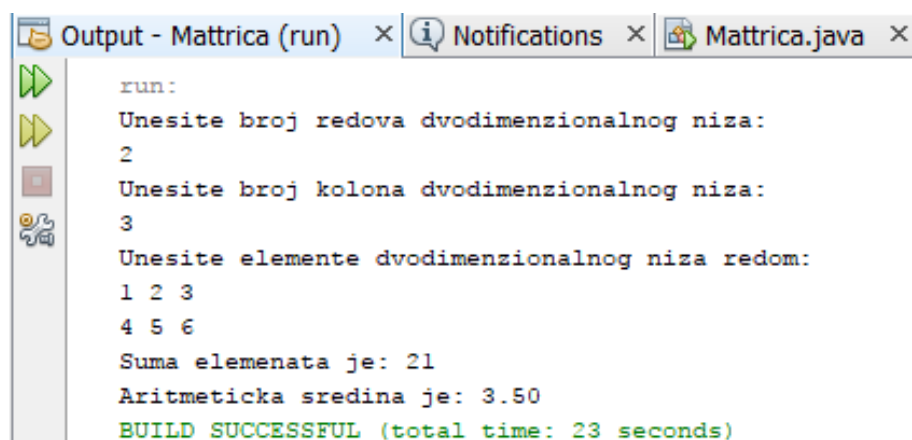
Korišćenjem objekta ulaz klase Scanner ćemo učitati broj redova i kolona dvodimenzionalnog niza u celobrojne promenljive **n** i **m**, a zatim ćemo konstruisati celobrojni dvodimenzionalni niz pod nazivom **matrica** od **n** redova i **m** kolona. Celobrojnu promenljivu **suma** ćemo inicijalno postaviti na vrednost 0, a zatim ćemo je uvećavati za **j**-ti element **i**-tog reda dvodimenzionalnog niza koristeći petlju **for**. Na kraju ćemo izračunati i ispisati aritmetičku sredinu i sumu svih elemenata dvodimenzionalnog niza.

PRIMER 2 – PROGRAMSKI KOD

Ovde se prikazuje programski kod u Javi koji rešavanja dati problem.

```
package KI103-L07;
import java.util.Scanner;
public class Primer2 {
    public static void main(String[] args) {
        Scanner ulaz = new Scanner(System.in);
        System.out.println("Unesite broj redova dvodimenzionalnog niza: ");
        int n = ulaz.nextInt();
        System.out.println("Unesite broj kolona dvodimenzionalnog niza: ");
        int m = ulaz.nextInt();
        int[][] matrica = new int[n][m];
```

```
System.out.println("Unesite elemente dvodimenzionalnog niza redom: ");
int suma = 0;
for (int i = 0; i < n; i++) {
    for (int j = 0; j < m; j++) {
        matrica[i][j] = ulaz.nextInt();
        suma += matrica[i][j];
    }
}
double aritmSredina = (double) suma / (n * m);
System.out.println("Suma elemenata je: " + suma);
System.out.printf("Aritmeticka sredina je: %.2f\n", aritmSredina);
}
```



```
run:
Unesite broj redova dvodimenzionalnog niza:
2
Unesite broj kolona dvodimenzionalnog niza:
3
Unesite elemente dvodimenzionalnog niza redom:
1 2 3
4 5 6
Suma elemenata je: 21
Aritmeticka sredina je: 3.50
BUILD SUCCESSFUL (total time: 23 seconds)
```

Slika 1.1.4 Rezultat izvršavanja programa Primer2A

PRIMER 3

Primena metoda `public int[] ucitajMatricu(int n, int m)` za učitavanje dvodimenzionalnog niza sa n redova i m kolona.

Tekst zadatka:

Definisati sledeće metode i izmeniti prethodni program tako da poziva ove metode:

`public int[] ucitajMatricu(int n, int m)` - na osnovu parametar n i m učitati dvodimenzionalni niz od n redova i m kolona i vratiti učitani dvodimenzionalni niz
`public int sumaMatrice(int[][] matrica)` - na osnovu parametra `matrica` tipa dvodimenzionalni celobrojni niz, vratiti sumu svih elemenata ovog dvodimenzionalnog niza

Program treba da bude deo NetBeans projekta pod nazivom KI103-L07. Naziv klase treba da bude Primer3. Za učitavanje podataka od korisnika treba koristiti objekat klase **Scanner**, a za ispis podataka objekat **System.out**.

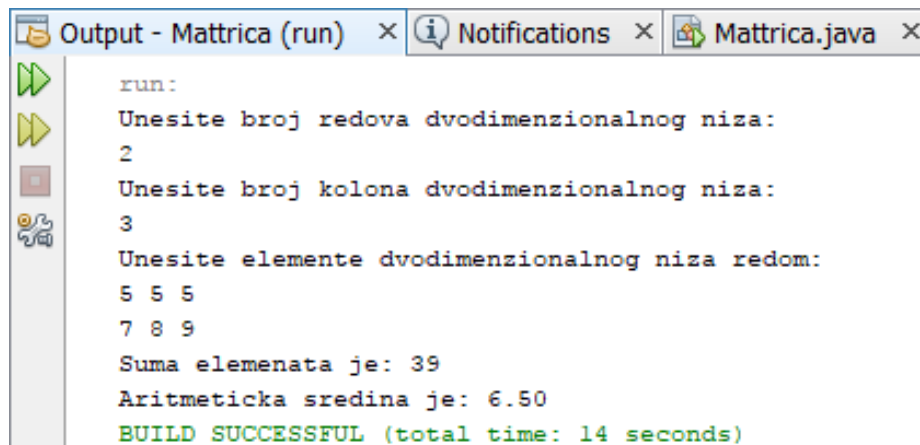
Analiza problema:

Postupak je sličan kao u prethodnom primeru.

PRIMER 3 – PROGRAMSKI KOD

Programski kod za prethodno opisan problem.

```
package KI103-L07;
import java.util.Scanner;
public class Primer3 {
    static Scanner ulaz = new Scanner(System.in);
    public static void main(String[] args) {
        System.out.println("Unesite broj redova dvodimenzionalnog niza: ");
        int n = ulaz.nextInt();
        System.out.println("Unesite broj kolona dvodimenzionalnog niza: ");
        int m = ulaz.nextInt();
        int[][] matrica = učitajMatricu(n, m);
        double aritmSredina = (double) sumaMatrice(matrica) / (n * m);
        System.out.println("Suma elemenata je: " + sumaMatrice(matrica));
        System.out.printf("Aritmeticka sredina je: %.2f\n", aritmSredina);
    }
    public static int[][] učitajMatricu(int n, int m) {
        int[][] matrica = new int[n][m];
        System.out.println("Unesite elemente dvodimenzionalnog niza redom: ");
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                matrica[i][j] = ulaz.nextInt();
            }
        }
        return matrica;
    }
    public static int sumaMatrice(int[][] matrica) {
        int suma = 0;
        int n = matrica.length;
        int m = matrica[0].length;
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                suma += matrica[i][j];
            }
        }
        return suma;
    }
}
```



```
run:
Unesite broj redova dvodimenzionalnog niza:
2
Unesite broj kolona dvodimenzionalnog niza:
3
Unesite elemente dvodimenzionalnog niza redom:
5 5 5
7 8 9
Suma elemenata je: 39
Aritmeticka sredina je: 6.50
BUILD SUCCESSFUL (total time: 14 seconds)
```

Slika 1.1.5 Rezultat izvršavanja programa Primer2B

PRIMER 4

Ispis elemenata na glavnoj dijagonali matrice.

Učitati i ispisati dvodimenzionalni niz - matricu $n \times n$.

Zatim ispisati članove niza koji su na glavnoj dijagonali.

Pojašnjenje:

Glavna dijagonala - Elementi čiji su indeksi isti (iste indekse reda i kolone odnosno $i=j$) su elementi glavne dijagonale.

Elementi iznad glavne dijagonale glavne dijagonale (indeks reda manji od indeksa kolone $i < j$), a elementi ispod glavne dijagonale (indeks reda veći od indeksa kolone $i > j$).

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package primer4;

import java.util.Scanner;

/**
 *
 * @author Aleksandra
 */
public class Primer4 {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
```

```

System.out.print("Unesite dimenzije kvadratne matrice ");
int n = in.nextInt();

int niz[][] = new int[n][n];

System.out.print("unesi članove ");
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        System.out.print("X[" + i + ", " + j + "]" + "=");
        niz[i][j] = in.nextInt();
    }
}

System.out.println();
System.out.print("Članovi na glavnoj dijagonali su: ");
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        if (i == j) {
            System.out.print(niz[i][j] + " ");
        }
    }
}
}
}

```

PRIMER 4 - DODATNO POJAŠNJENJE

dodatno pojašnjenje

Trenutno se u zadatku 2 koristi sledeći pristup za ispisivanje elmeneta na glavnoj dijagonali:

```

System.out.println();
System.out.print("Članovi na glavnoj dijagonali su: ");
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        if (i == j) {
            System.out.print(niz[i][j] + " ");
        }
    }
}
}

```

Ideja je da se iskoristi deo koda koji služi za ispisivanje dvodimenzionalnog niza i modifikuje tako da se ne ispisuju svi elementi niza već samo oni koji ispunjavaju uslov:

if (i == j)

Drugo rešenje koje je evkivalentno prethodnom delu koda jeste da se prodje jednom petljom kroz niz

-koristimo brojač i

štampano elemente koji su u redu i i u koloni i

tj elemente niz[i][i]

```
System.out.print("Članovi na glavnoj dijagonali su: ");  
    for (int i = 0; i < n; i++) {  
        System.out.print(niz[i][i] + " ");
```

ZADATAK 1

Ispis parnih elemenata matrice

- Kreirajte nov NetBeans projekat pod nazivom Matrica, u paketu L07;
- Učitajte broj vrsta i kolona n i m;
- Učitajte članove matrice;
- Prođite petljama kroz matricu i ispišite na konzoli sve parne članove matrice.

ZADATAK 2

Određivanje najvećeg elementa matrice.

- Kreirajte nov NetBeans projekat pod nazivom Matrica2, u paketu L07;
- Učitajte broj vrsta i kolona n i m;
- Učitajte članove matrice;
- Pronađite najveći element u matrici;
- Rezultat pretrage prikazati na konzoli

✓ 1.2 Korišćenje dvodimenzionalnih nizova

PREDSTAVLJANJE TABELA POMOĆU DVODIMENZIONALNIH NIZOVA

Svaki element 2D niza je definisan preko broja vrste i broja kolone – i to baš tim redom.

Podaci koje koristimo u programima su često dvodimenzionalni. Na primer, mape imaju dve dimenzije; strana na kojoj će se štampati neki tekst, takođe itd. U takvim situacijama nam je potreban dvodimenzionalni niz. On se može posmatrati kao skup elemenata koji su raspoređeni u obliku tabele. Svaki element sadrži vrednost, ali su za pristup pojedinom elementu potrebna dva indeksa.

Pretpostavimo da imamo razred sa sedam učenika koji su tokom 5 nedelja, svake nedelje imali test. Predavač je broj poena zapisivao u tabeli. Konkretna ćelija u tabeli se određuje preko id oznake studenta i broja nedelje. Na primer:

- Broj poena za studenta sa brojem 0 u prvoj nedelji je 42.
- Broj poena za studenta sa brojem 3 u četvrtoj nedelji je 93.

Ćelija se u tabeli najčešće predstavlja preko indeksa vrste i kolone:

```
tabelaPoena[ vrsta ][ kolona ]
```

Na primer:

tabelaPoena [0][1] je 42

tabelaPoena [3][4] je 93

Izgled cele tabele dat je na slici 1.

Student	Nedelja				
	0	1	2	3	4
0	99	42	74	83	100
1	90	91	72	88	95
2	88	61	74	89	96
3	61	89	82	98	93
4	93	73	75	78	99
5	50	65	92	87	94
6	43	98	78	56	99

Slika 1.2.1 Prikaz tabele predstavljene dvodimenzionalnim nizom

Tabela se u Javi može implementirati kao [dvodimenzionalni niz](#). Svaki elemenat niza je promenljiva koja može sadržati vrednost i ponaša se kao i bilo koja druga promenljiva. Kao i kod jednodimenzionalnih nizova i ovde su svi elementi istog tipa. Tip može biti primitivan tip ili referenca na objekat.

Svaki elemenat 2D niza je definisan preko broja vrste i broja kolone – i to baš tim redom.

Iako smo u prethodnoj tabeli prikazali i zaglavlje tabele, zaglavlje nije deo niza.

GRANICE NIZA

Ako se pokuša pristup elementu koji ne postoji, jer se izlazi van granica niza, javlja se greška (izuzetak) tipa.

Za 2D nizove u Javi treba upamtiti sledeće:

- Vrste se označavaju brojevima od 0 do N-1, gde je N broj vrsti.
- Kolone se označavaju brojevima od 0 do M-1, gde je M broj kolona.
- 2D niz sa N vrsta i M kolona će imati N x M elemenata.
- Moguće je da 2D niz ima različit broj elemenata u svakoj vrsti.

Provera granica niza

Kada program u toku rada pokuša da pristupi nekom elementu niza, Java virtuelna mašina proverava da li taj element zaista postoji. Tom prilikom se vrši provera granica niza. Ako se pokuša pristup elementu koji ne postoji, javlja se greška (izuzetak) tipa.

ArrayIndexOutOfBoundsException

Ovo će obično prekinuti rad programa. Indeksi niza moraju biti celi brojevi. Nema smisla da napišete iskaz:

tabelaPoena[3.5][2].

Kao i kod jednodimenzionalnog niza, indeks niza može biti literal, promenljiva celobrojnog tipa, metod koji vraća ceo broj ili neki aritmetički izraz. Svi naredni iskazi su ispravni:

```
tabelaPoena[ 3 ][ j ] = 34;  
sum = tabelaPoena [ i ][ j ] + tabelaPoena [ i ][ j+1 ] ;  
value = tabelaPoena [ 2 ][ nekaFunkcija ( ) ] ;  
tabelaPoena [ 1 ][ 0 ]=tabelaPoena[ i+3 ][ nekaFunkcija()-2 ] ;
```

VIŠEDIMENZIONALNI NIZOVI (VIDEO)

Video pokazuje formiranje i korišćenje višedimenzionalnih nizova

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

KREIRANJE VIŠEDIMENZIONALNIH NIZOVA (VIDEO)

Video pokazuje kako od nekoliko jednodimezionalnih nizova možete napraviti jedan višedimenzionalni niz.

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

SORTIRANJE VIŠEDIMENZIONALNIH NIZOVA (VIDEO)

Video pokazuje kako možete da sortirate višedimenzionalne nizove.

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

PRIMER 5

Kreiranje klase Array2DUtility sa pomoćnim metodama za rad sa nizovima, uključujući i metode ucitaMatricu() i sumaMatrice().

Tekst zadatka:

Možemo kreirati i novu klasu **Array2DUtility** koja će sadržati sve pomoćne metode koje se tiču nizova, a definicije metoda **sumaMatrice** i **ucitajMatricu** prebaciti u ovu klasu.

Prethodni program ćemo promeniti tako da poziva statičke metode klase **ArrayUtility** po nazivu klase i naziv klase promeniti u Primer5. Prednost ovog pristupa je u tome što više različitih klasa može pozivati statičke metode klase **Array2DUtility**.

PROBLEM 5 – PROGRAMSKI KOD*Listing programa opisan primer.*

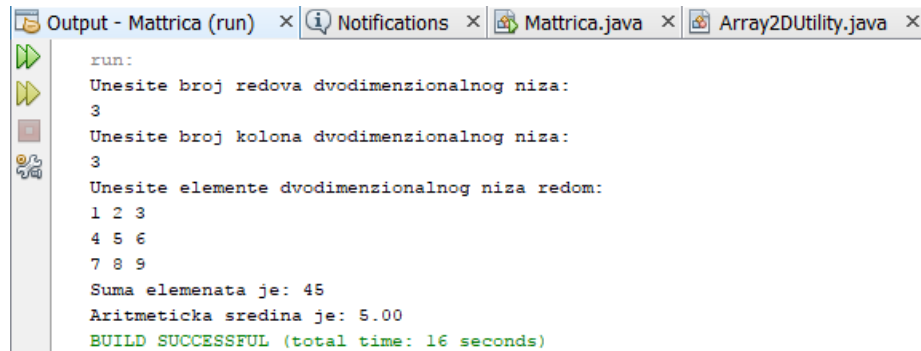
```
package KI103.l07;
import java.util.Scanner;
public class Primer5 {
    static Scanner ulaz = new Scanner(System.in);
    public static void main(String[] args) {
        System.out.println("Unesite broj redova dvodimenzionalnog niza: ");
        int n = ulaz.nextInt();
        System.out.println("Unesite broj kolona dvodimenzionalnog niza: ");
        int m = ulaz.nextInt();
        int[][] matrica = Array2DUtility.ucitajMatricu(n, m);
        double aritmSredina = (double) Array2DUtility.sumaMatrice(matrica) / (n *
m);
        System.out.println("Suma elemenata je: " +
Array2DUtility.sumaMatrice(matrica));
        System.out.printf("Aritmeticka sredina je: %.2f\n", aritmSredina);
    }
}

package KI103.l07;
import static KI103.l07.Primer5.ulaz;
public class Array2DUtility {
    public static int[][] ucitajMatricu(int n, int m) {
        int[][] matrica = new int[n][m];
        System.out.println("Unesite elemente dvodimenzionalnog niza redom: ");
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                matrica[i][j] = ulaz.nextInt();
            }
        }
        return matrica;
    }
    public static int sumaMatrice(int[][] matrica) {
        int suma = 0;
        int n = matrica.length;
        int m = matrica[0].length;
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                suma += matrica[i][j];
            }
        }
    }
}
```

```

    }
    }
    return suma;
}
}

```



```

run:
Unesite broj redova dvodimenzionalnog niza:
3
Unesite broj kolona dvodimenzionalnog niza:
3
Unesite elemente dvodimenzionalnog niza redom:
1 2 3
4 5 6
7 8 9
Suma elemenata je: 45
Aritmeticka sredina je: 5.00
BUILD SUCCESSFUL (total time: 16 seconds)

```

Slika 1.2.2 Rezultat izvršavanja programa Primer2C

ZADATAK 3

Nova matrica kreirana iz početne.

- Napisati program koji od korisnika zahteva da unese broj redova i broj kolona.
- Napraviti matricu $n \times m$ sa random elementima koristeći dvodimenzionalni niz.
- Napraviti novi dvodimenzionalni niz sa elementima koji su kvadrati elemenata prvog niza.

ZADATAK 4

Prebrojavanje članova matrice većih od aritmetičke sredine njenih članova.

- Napisati program koji od korisnika zahteva da unese broj redova i broj kolona.
- Napraviti matricu $n \times m$ sa random elementima koristeći dvodimenzionalni niz.
- Odrediti zbir i aritmetičku sredinu članova matrice;
- Prebrojati broj članova matrice koji su veći od aritmetičke sredine;
- Štampati rezultate na konzoli.

▼ Poglavlje 2

Domaći zadaci

ZADACI 1 - 3

Zadaci za individualnu vežbu

Zadatak 1:

Napisati program koji od korisnika zahteva da unese veličine dvodimenzionalnog niza n i m . Na osnovu unetih vrednosti kreirati niz random vrednosti. Napisati metodu koja izračunava sumu elemenata na sporednoj dijagonali i dobijeni rezultat odštampati na konzoli.

Zadatak 2:

Napisati program koji od korisnika zahteva da unese veličine dvodimenzionalnog niza n i m . Generisati random vrednosti za svaki element tog niza. U tako kreiranom nizu potrebno je pronaći maksimalan i minimalan element tog niza i zameniti im mesta.

Zadatak 3:

Napisati program koji od korisnika zahteva da unese veličine dvodimenzionalnog niza n i m . Na osnovu unetih vrednosti kreirati niz random vrednosti. Prebrojati parne i neparne članove niza i kao rezultat štampati veću vrednost od ove dve,

▼ Zaključak

REZIME

Nizovi su uređeni skupovi podataka, kojim se može manipulirati kao sa objektima.

U predavanju smo zaokružili rad sa nizovima u Javi. Sada ste stekli kompletnu predstavu u vezi sa:

- Pojmom niza kao strukture podataka
- Jednodimenzionalnim nizovima
- Klasom Arrays
- Naredbom for-each
- Nedostacima nizova
- Dvodimenzionalnim nizovima

Ova i prethodna lekcija obuhvataju vežbanja:

- Kreiranje programa za učitavanje i obradu jednodimenzionalnog niza
- Kreiranje posebne klase sa statičkim metodama za rad sa jednodimenzionalnim nizovima
- Kreiranje metoda za rad sa jednodimenzionalnim nizovima bez korišćenja petlji
- Kreiranje metoda za rad sa jednodimenzionalnim nizovima sa korišćenjem petlji
- Kreiranje programa za učitavanje i obradu dvodimenzionalnog niza
- Kreiranje metoda za rad sa stringovima bez korišćenja petlji
- Kreiranje metoda za rad sa stringovima sa korišćenjem petlji