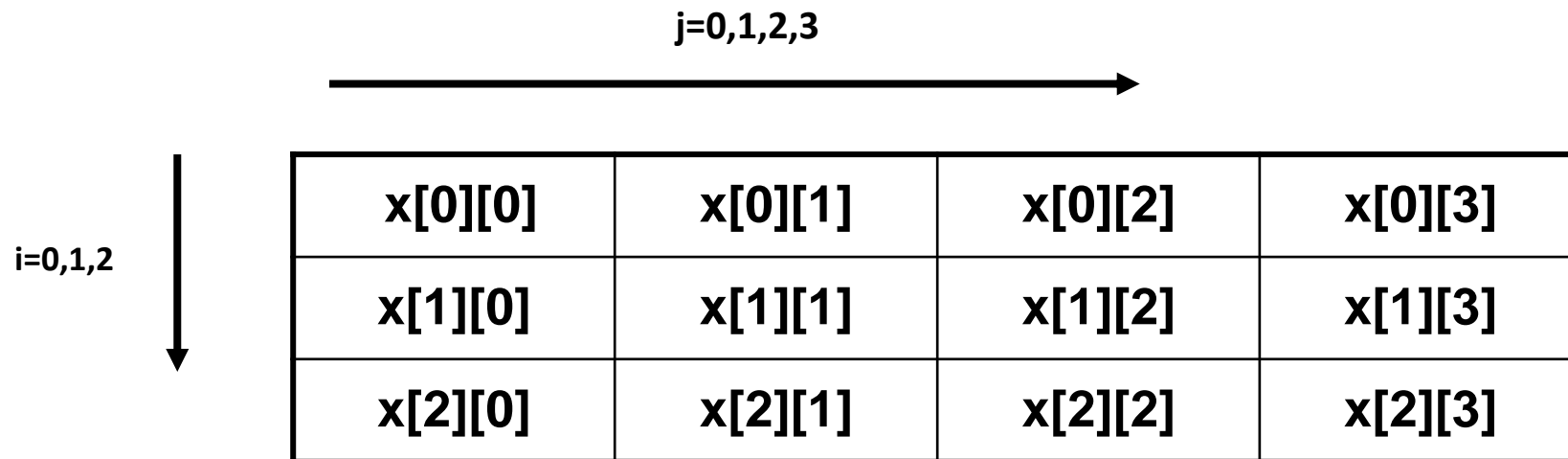


Dvodimenzionalni nizovi

# Dvodimenzionalni nizovi (matrice)

```
int x[][] = new int[3, 4];  
// deklaracija i instanciranje 2D niza od tri vrste i četiri kolone
```



`x[i][j]` – u preseku i-te vrste i j-te kolone matrice

# Dvodimenzionalan niz

```
public static void main(String[] args) {  
  
    int[][] x = new int[3][4];  
  
    for (int i = 0; i < 3; i++) {  
        for (int j = 0; j < 4; j++) {  
            System.out.print(x[i][j] + "\t");  
        }  
        System.out.println();  
    }  
}
```

# Inicijalizacija 2D niza

```
public static void main(String[] args) {  
    int[][] x = {  
        {1,2,3,4},  
        {2,3,4,5},  
        {3,4,5,6}  
    };  
    for (int i = 0; i < 3; i++) {  
        for (int j = 0; j < 4; j++) {  
            System.out.print(x[i][j] + "\t");  
        }  
        System.out.println();  
    }  
}
```

# Dimenzije 2D niza

```
public static void main(String[] args) {  
    int[][] x = {  
        {1,2,3,4},  
        {2,3,4,5},  
        {3,4,5,6}  
    };  
  
    int brojVrsta = x.length;  
    int brojKolona = x[0].length;  
  
    System.out.println("Broj vrsta: " + brojVrsta);  
    System.out.println("Broj kolona: " + brojKolona);  
}
```

# Štampanje članova 2D niza – paket nizovi2D

```
package nizovi2D;

public class nizovi2D {
    public static void pisi2Dniz(int[][] x)
    {
        int brojVrsta = x.length;
        int brojKolona = x[0].length;

        for (int i = 0; i < brojVrsta; i++)
        {
            for (int j = 0; j < brojKolona; j++)
            {
                System.out.print(x[i][j] + "\t");
            }
            System.out.println();
        }
    }
}
```

# Paket nizovi2D – funkcija kreiraj2Dniz

```
public static int[][] kreiraj2Dniz(int n, int m) {  
    int[][] x = new int[n][m];  
    Random rnd = new Random();  
    for (int i = 0; i < n; i++)  
    {  
        for (int j = 0; j < m; j++){  
            x[i][j] = rnd.nextInt(10)+1;  
        }  
    }  
    return x;  
}
```

# Metoda za učitavanje 2D niza- paket Nizovi2D

```
public static int[][] citaj2Dniz(int n, int m, Scanner sc){
    int[][] x = new int[n][m];
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            System.out.format("x[%d][%d]=?", i,j);
            x[i][j] = sc.nextInt();
        }
    }
    return x;
}
```



# Paket Nizovi2D funkcija linija

```
public static void linija(int n) {  
    for (int i = 0; i < n; i++) {  
        System.out.print("_");  
    }  
    System.out.println();  
}
```

# Paket nizovi2D funkcija pisiNiz

```
public static void pisiNiz(int[] x)
{
    for (int i : x)
    {
        System.out.print(i + "\t");
    }
    System.out.println();
}
```

# Poziv funkcija iz paketa

```
public static void main(String[] args) {  
    int x1[][] = {  
        { 1, 2, 3, 4, 5 },  
        { 6, 7, 8, 9, 10 }  
    };  
  
    nizovi2D.pisi2Dniz(x1);  
    nizovi2D.linija(50);  
  
    int x2[][] = nizovi2D.kreiraj2Dniz(3, 4);  
    nizovi2D.pisi2Dniz(x2);  
}
```

# Učitavanje i štampanje matrice

```
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
  
    int x[][] = nizovi2D.citaj2Dniz(2, 3, sc);  
    nizovi2D.linija(50);  
    nizovi2D.pisi2Dniz(x);  
}
```

# Kreiranje 1D niza od reda 2D niza

```
public static void main(String[] args) {  
    // matrica od 4 reda i 5 kolone  
    int[][] x = nizovi2D.kreiraj2Dniz(4, 5);  
    nizovi2D.pisi2Dniz(x);  
    nizovi2D.linija(50);  
  
    System.out.println("Drugi red matrice je:");  
  
    int x1[] = new int[5];  
    for (int j = 0; j < 5; j++)  
    {  
        // prepisujemo drugi red u niz  
        x1[j] = x[1][j];  
    }  
  
    nizovi2D.pisiNiz(x1);  
}
```

# Primer

Odreditu sumu elemenata na glavnoj dijagonali matrice dimenzija  $n \times m$ .

```
public static int sumaGlavneDijagonale(int[][] niz) {  
    int duzina = Math.min(niz.length, niz[0].length);  
    int suma = 0;  
    for (int i = 0; i < duzina; i++) {  
        suma += niz[i][i];  
    }  
    return suma;  
}
```

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Unesite broj redova niza: ");
    int n = scanner.nextInt();
    System.out.print("Unesite broj kolona niza: ");
    int m = scanner.nextInt();
    scanner.close();
    nizovi2D.niz = nizovi2D.kreiraj2Dniz(n, m);
    nizovi2D.pisi2Dniz(niz);
    nizovi2D.linija(50);
    int suma = sumaGlavneDijagonale(niz);

    System.out.println("Suma elemenata na glavnoj dijagonali niza je: " + suma);
}
```

Definisanje klase i kreiranje objekata



# Apstraktni tipovi podataka

- Apstrakcija je proces kojim se objekti iz realnog sveta modeliraju uprošćenom "slikom"
- Apstrakcijom se ističu samo **bitne karakteristike** objekta a zanemaruju se manje bitne karakteristike objekta
- Postoje ugrađeni (osnovni, primitivni) tipovi podataka npr. float, int, double, ...
- Ravnopravno se definišu korisnički definisani tipovi podataka: TekuciRacun, Osoba, Student, ...
- Klasa je novi tip podataka koji se uvodi u objektnom programiranju

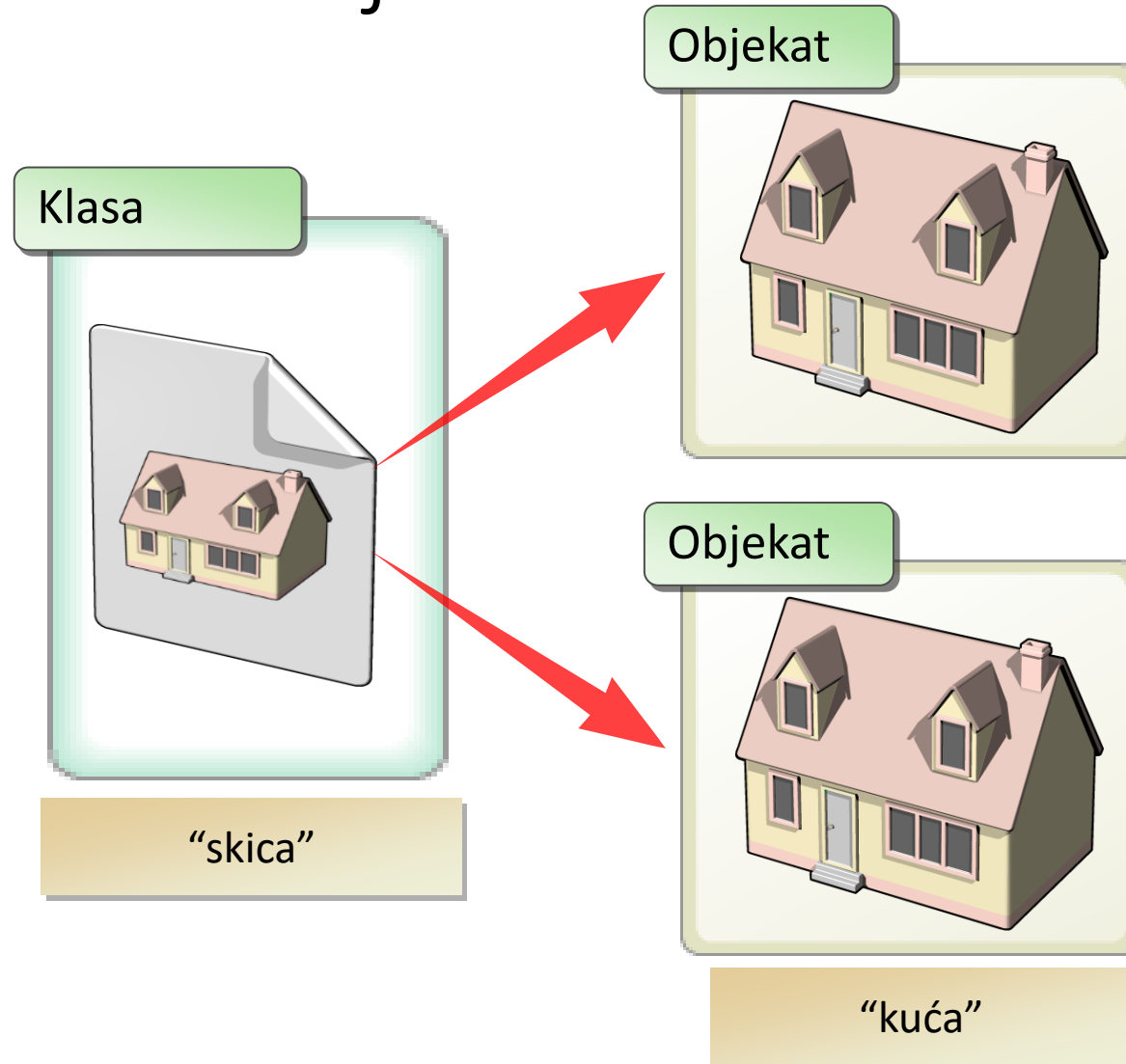
# Enkapsulacija

- Enkapsulacija je sposobnost objekta da skriva svoje unutrašnje podatke i implementacione detalje
- Grupisanje podataka i koda koji manipuliše podacima
- Enkapsulacija se ostvaruje korišćenjem klase kao novog tipa podataka
- Realizija nekog tipa podataka može i treba da se sakrije od ostatka sistema tj. onih koji ga koriste
- Korisnicima se definiše šta se sa tipom može uraditi a način na koji se to radi se skriva

# Odnos klase i objekata

- Klasa predstavlja apstraktnu predstavu skupa objekata koji imaju iste osobine
- Klasa :
  - model koji opisuje kako kreirati objekat
  - je kao "šematski plan", skica
  - Sadrži podatke (polja) i metode
- Objekti:
  - objekat je predstava nekog entiteta iz realnog sveta
  - instance klase
  - može biti više objekata (instanci) klase

# Primer klase i objekata



# Definisanje klase

```
package klase01;  
  
public class TekuciRacun {  
    public String ime;  
    public String prezime;  
    public double stanje;  
}
```

# Kreiranje objekata

- Objekti su inicijalno neoznačeni
- Podrazumevana vrednost objekta je null
- Da bi se koristila promenljiva tipa klase klasa se mora instancirati
- Nova instanca klase kreira se korišćenjem operatora **new**

# Instanciranje klase (kreiranje objekata)

```
package klase01;

public class Klase01 {
    public static void main(String[] args) {
        TekuciRacun tr1 = new TekuciRacun();
        tr1.ime = "Pera";
        tr1.prezime = "Peric";
        tr1.stanje = 123456.2;

        System.out.format("Korisnik %s %s ima %.2f dinara na racunu%n",
            tr1.ime, tr1.prezime, tr1.stanje);
    }
}
```

# Pokretanje java aplikacije

```
C:\Users\goran\Desktop\java08>cd Klase01
```

```
C:\Users\goran\Desktop\java08\klase01>javac *.java
```

```
C:\Users\goran\Desktop\java08\klase01>cd..
```

```
C:\Users\goran\Desktop\java08>java klase01.Klase01
```



# Pokretanje java aplikacije

```
C:\Users\goran\Desktop\java08>cd Klase01  
  
C:\Users\goran\Desktop\java08\klase01>javac *.java  
  
C:\Users\goran\Desktop\java08\klase01>cd..  
  
C:\Users\goran\Desktop\java08>java klase01.Klase01  
Korisnik Pera Peric ima 123456,20 dinara na racunu  
  
C:\Users\goran\Desktop\java08>|
```

# Statička metoda Stampaj klase Klase01

```
public static void stampaj(TekuciRacun tr)
{
    System.out.format("Korisnik %s %s ima %f dinara na racunu%n",
        tr.ime, tr.prezime, tr.stanje);
}
```

# Metoda Main klase Klase01

```
public static void main(String[] args) {
    TekuciRacun tr1 = new TekuciRacun();
    tr1.ime = "Pera";
    tr1.prezime = "Peric";
    tr1.stanje = 123456.2;

    //System.out.format("Korisnik %s %s ima %.2f dinara na racunu%n", tr1.ime,
tr1.prezime, tr1.stanje);
    stampaj(tr1);
    tr1 = null;
    stampaj(tr1);
}
```

# Generisanje izuzetka od strane okruženja

```
Korisnik Pera Peric ima 123456,200000 dinara na racunu  
Exception in thread "main" java.lang.NullPointerException: Cannot read field "ime"  
because "<parameter1>" is null  
    at klase01.Klase01.stampaj(Klase01.java:18)  
    at klase01.Klase01.main(Klase01.java:13)  
  
C:\Users\goran\Desktop\java08>
```

# Metode klase

- Metode klase predstavljaju funkcije – članice klase.
- Svaka metoda sadrži :
  - tip podataka koga metoda vraća (ili void ukoliko ne vraća podatke)
  - naziv(ime metode)
  - listu parametara
  - telo metode
- Ukoliko metoda vraća neku vrednost, onda se unutar tela metode mora pozvati naredba **return**.

```
public T nazivMetode(T1 param1, ..., TN paramN)
{
    // telo metode
    return rezultat;
}
```

# Primeri metoda

```
public double uplata(double iznos) {  
    stanje += iznos;  
    return stanje;  
}
```

```
public void uplata(double iznos) {  
    stanje += iznos;  
}
```

```
public void isplata(double iznos) {  
    stanje -= iznos;  
}
```

# Metoda Stampaj klase TekuciRacun

```
public void stampaj(){  
    System.out.format("Korisnik %s %s ima %f dinara na racunu%n",  
        ime, prezime, stanje);  
}
```

# Klasa sa poljima i metodama

```
package klase01;

public class TekuciRacun {
    public String ime;
    public String prezime;
    public double stanje;

    public void uplata(double iznos){

        stanje += iznos;
    }

    public void isplata(double iznos) {

        stanje -= iznos;
    }

    public void stampaj(){
        System.out.format("Korisnik %s %s ima %f dinara na racunu%n", ime, prezime, stanje);
    }
}
```



# Pozivanje metoda klase

```
public static void main(String[] args) {
    TekuciRacun tr1 = new TekuciRacun();
    tr1.ime = "Pera";
    tr1.prezime = "Peric";
    tr1.stanje = 123456.2;

    //System.out.format("Korisnik %s %s ima %.2f dinara na racunu%n", tr1.ime,
    //tr1.prezime, tr1.stanje);
    //stampaj(tr1);
    //tr1 = null;
    //stampaj(tr1);
    tr1.stampaj();
    tr1.uplata(45678.12);
    tr1.stampaj();
    tr1.isplata(12345.45);
    tr1.stampaj();
}
```