

Predavanje 3.

OBJEKTNO-ORIJENTISANO PROGRAMIRANJE

2024/25.

Agenda

za danas

this

Poziv metoda iz iste klase i this()

Članovi klasa

this

AGENDA

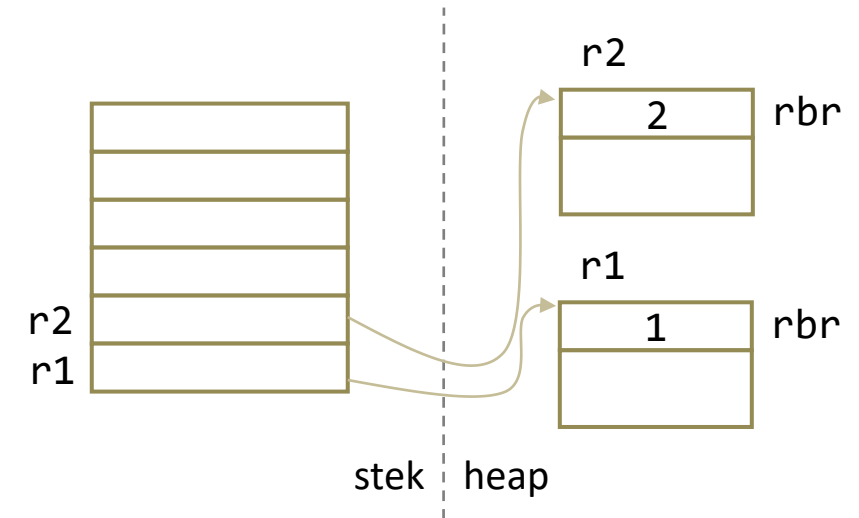
this

Poziv metoda iz iste klase i this()

Članovi klasa

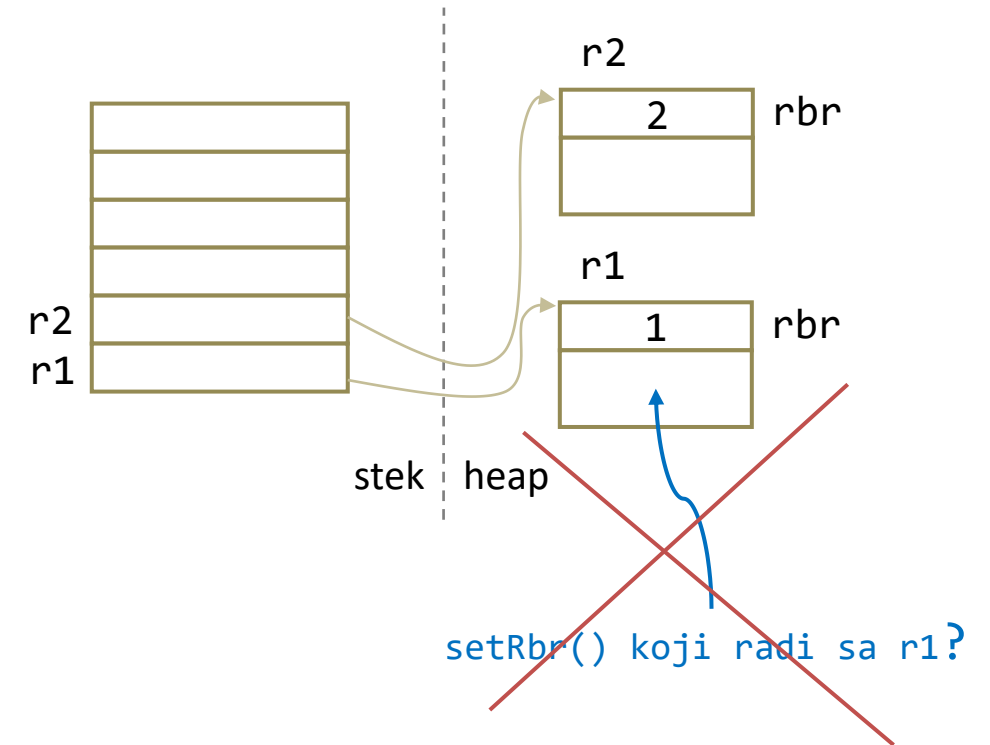
Svaki objekat ima svoje podatke

```
class Robot {  
    int rbr; ...  
    void setRbr(int br) {  
        rbr = br;  
    }  
}  
class Test {  
    public static void main() {  
        Robot r1 = new Robot();  
        Robot r2 = new Robot();  
        r1.setRbr(1);  
        r2.setRbr(2);  
    }  
}
```



Kako metod zna sa čijim podacima radi?

```
class Robot {  
    int rbr; ...  
    void setRbr(int br) {  
        rbr = br;  
    }  
}  
  
class Test {  
    public static void main() {  
        Robot r1 = new Robot();  
        Robot r2 = new Robot();  
        r1.setRbr(1);  
        r2.setRbr(2);  
    }  
}
```



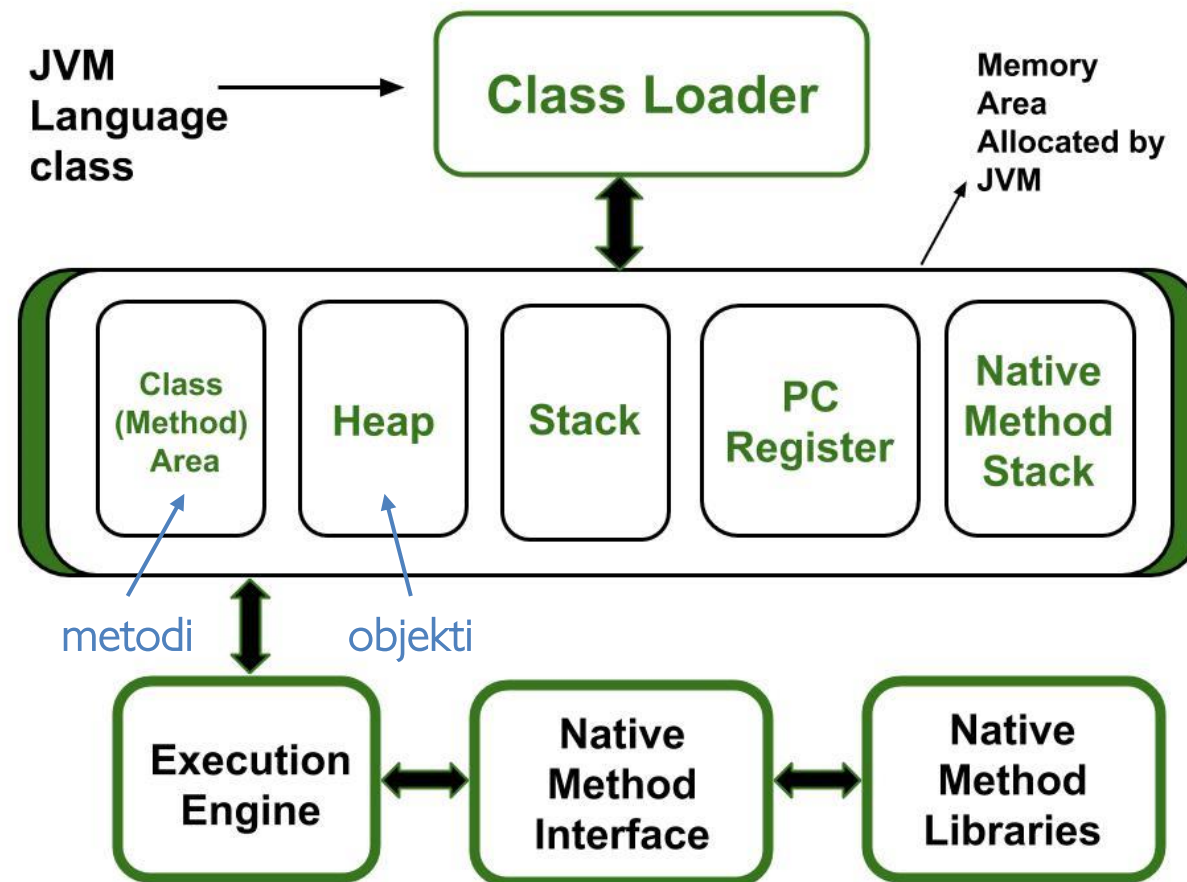
Kako metod zna sa čijim podacima radi?

Objekti čuvaju samo podatke.

Kod je izdvojen u poseban deo memorije.

Ne čuvaju kod metoda, već samo informaciju o tome gde se nalazi kod u memoriji (tačnije imaju referencu na klasu).

Kako onda izgleda definicija metoda?



Kompajler menja definiciju metoda

Pri prevođenju koda, poziv metoda setRbr, kao i potpis tog metoda u klasi Robot će biti prošireni.

```
class Robot {  
    int rbr; ...  
    void setRbr(Robot this, int br) {  
        this.rbr = br;  
    }  
}
```

```
class Test {  
    public static void main(){  
        Robot r1 = new Robot();  
        Robot r2 = new Robot();  
        r1.setRbr(r1, 1);  
        r2.setRbr(r2, 2);  
    }  
}
```

Intervencija kompajlera

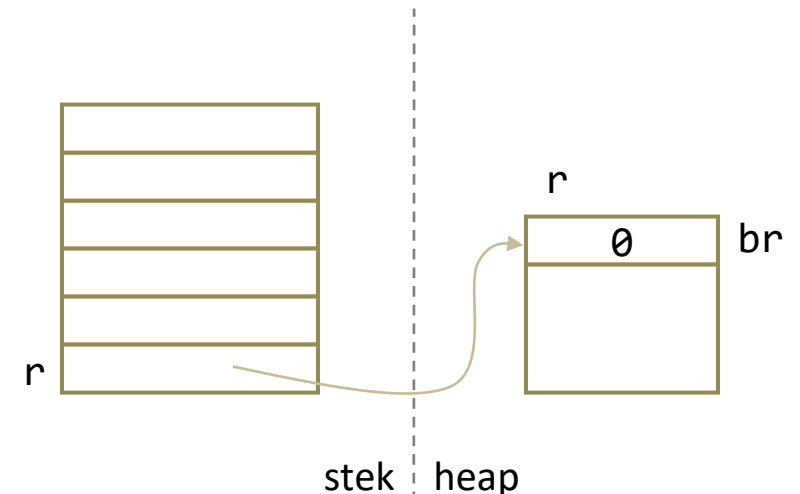
Dodaje ako mi nismo. Mora da se zna čiji je podatak.

Metod setRbr pored argumenta navednog u definiciji poseduje i **implicitni parametar `this` koji sadrži adresu objekta čiji je metod pozvan**, tj. objekta kojem je prosleđena poruka.

Odakle vrednost za this? Kompajler se o tome brine.

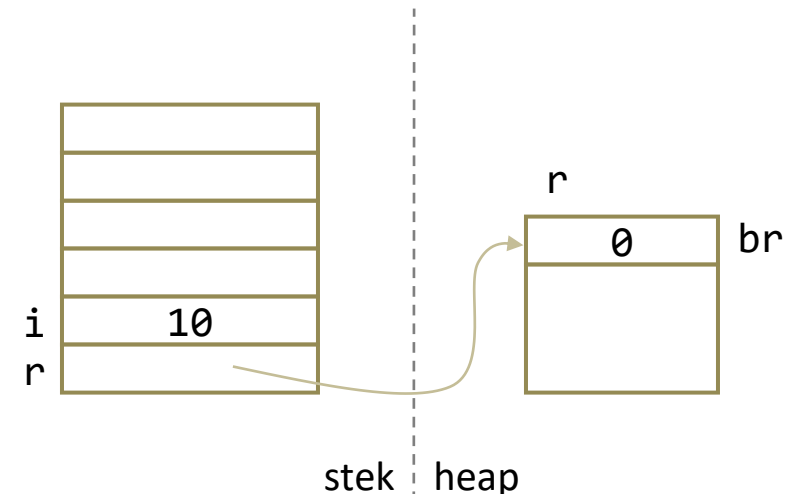
Kakvo je stanje na steku tokom izvršavanja

```
class Robot {  
    int rbr; ...  
    void setRbr(Robot this, int br) { this.rbr = br; } ...  
}  
class Test {  
    public static void main(){  
        Robot r = new Robot();  
        int i = 10;  
        r.setRbr(r,i);  
    }  
}
```



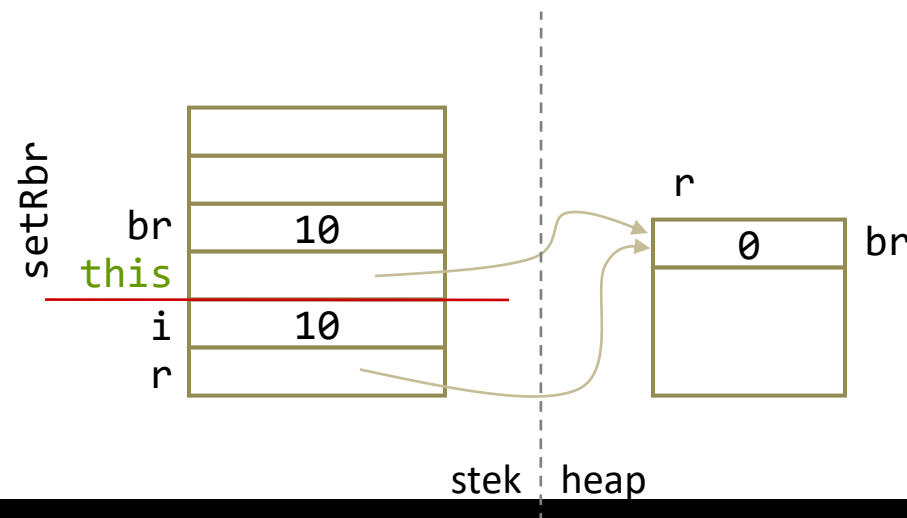
Kakvo je stanje na steku tokom izvršavanja

```
class Robot {  
    int rbr; ...  
    void setRbr(Robot this, int br) { this.rbr = br; } ...  
}  
  
class Test {  
    public static void main(){  
        Robot r = new Robot();  
        int i = 10;  
        r.setRbr(r,i);  
    }  
}
```



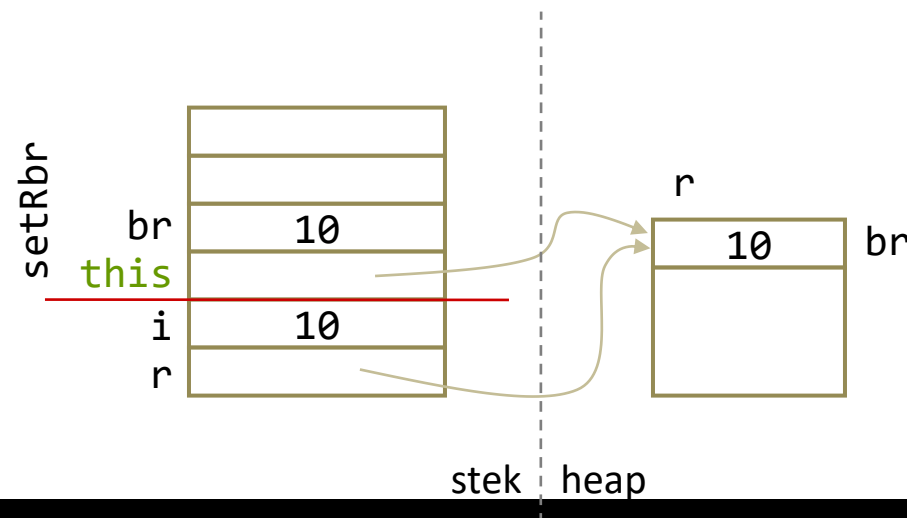
Kakvo je stanje na steku tokom izvršavanja

```
class Robot {  
    int rbr; ...  
    void setRbr(Robot this, int br) { this.rbr = br; } ...  
}  
class Test {  
    public static void main(){  
        Robot r = new Robot();  
        int i = 10;  
        r.setRbr(r,i);  
    }  
}
```



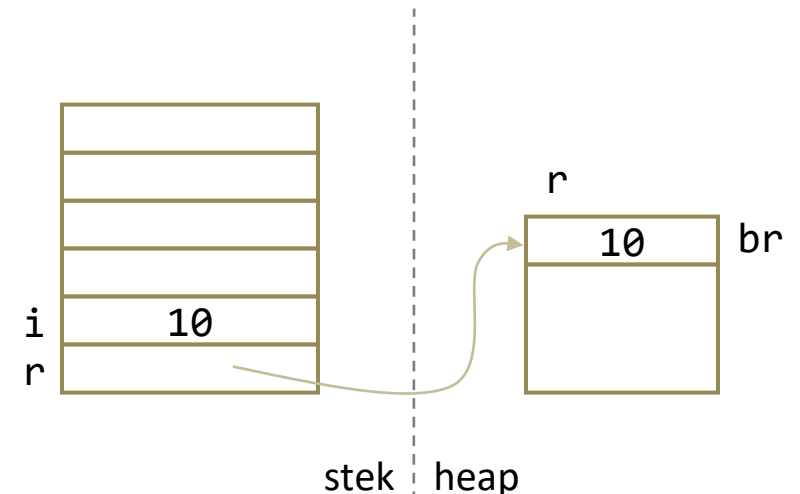
Kakvo je stanje na steku tokom izvršavanja

```
class Robot {  
    int rbr; ...  
    void setRbr(Robot this, int br) { this.rbr = br; } ...  
}  
  
class Test {  
    public static void main(){  
        Robot r = new Robot();  
        int i = 10;  
        r.setRbr(r,i);  
    }  
}
```



Kakvo je stanje na steku tokom izvršavanja

```
class Robot {  
    int rbr; ...  
    void setRbr(Robot this, int br) { this.rbr = br; } ...  
}  
class Test {  
    public static void main(){  
        Robot r = new Robot();  
        int i = 10;  
        r.setRbr(r,i);  
    }  
}
```



Primer

Napisati tip kojim se opisuje **Muzicar**. Svaki muzičar ima:

- ime,
- prezime,
- definisanu vrstu muzike koju izvodi i
- cenu jednog sata izvođenja

i ume da:

- se prijavi na dati festival
- da svoju string reprezentaciju u formi
ime prezime – vrsta muzike

Pri kreiranju muzičara se zadaju ime, prezime, vrsta muzike i cena jednog sata izvođenja.

Napisati tip kojim se opisuje **Festival**. Svaki festival ima:

- naziv
- spisak izvođača

I ume da:

- doda datog muzičara u spisak izvođača
- vrati ukupnu sumu novca koju treba isplatiti (podrazumeva se da svaki izvođač ima nastup od 60 minuta)
- da svoju string reprezentaciju u formi

Naziv

ime prezime – vrsta muzike

ime prezime – vrsta muzike

prijavljenih izvođača

Pri kreiranju festivala se zadaje naziv.

Napisati testnu klasu u kojoj se:

- Kreiraju objekti muzičara Tom Morelo (rok, 5000), Gucci Mane (rep, 3500), LAYLA BENITEZ (elektro, 4500)
- Kreira objekat Exit festivala.
- Kreirani muzičari se prijavljuju na Exit.
- Ispisuje string reprezentacija festivala.

Poziv metoda iz iste klase i this()

AGENDA

this

Poziv metoda iz iste klase i this()

Članovi klase

Poziv metoda iz iste klase

```
class Test {
public static void main(String[] args) {
    Kasica branko = new Kasica();
    int brojDana = 15;
    if (branko.ocekivano(brojDana) > 2000) branko.uzmiDzeparac(brojDana);
}}
class Kasica
{
    int stanje=100;
    int dzeparacDeda = 500;                // na dve nedelje
    int dzeparacRoditelji = 1000;          // na nedelju dana
    void uzmiDzeparac(int brojDana) {
        stanje += ocekivano(brojDana);
    }
    int ocekivano(int brojDana) {
        return brojDana%7*dzeparacDeda + brojDana%14*dzeparacRoditelji;
    }
}
```

Poziv metoda iz iste klase

```
class Test {
public static void main(String[] args) {
    Kasica branko = new Kasica();
    int brojDana = 15;
    if (branko.ocekivano(brojDana) > 2000) branko.uzmiDzeparac(brojDana);
}}
class Kasica
{
    int stanje=100;
    int dzeparacDeda = 500;                // na dve nedelje
    int dzeparacRoditelji = 1000;          // na nedelju dana
    void uzmiDzeparac(int brojDana) {
        this.stanje += this.ocekivano(brojDana);
    }
    int ocekivano(int brojDana) {
        return brojDana%7*dzeparacDeda + brojDana%14*dzeparacRoditelji;
    }
}
```


this()

- **this(<lista argumenata>)** poziva odgovarajući vlastiti konstruktor

```
class Robot {  
    int rbr;  
    Robot() {  
        this(1);  
    }  
    Robot(int rbr) {  
        this.rbr=rbr;  
    }  
}
```

- Ako postoji, **this(<lista argumenata>)** ona mora biti prva naredba u konstruktoru

Članovi klasa

AGENDA

this

Poziv metoda iz iste klase i this()

Članovi klasa

Članovi objekata i članovi klase

- Ispred deklaracije promenljivih i metoda se može navesti **static** modifikator.
- Static-om se obeležavaju članovi zajednički svim instancama date klase.

Statički podaci i metodi	članovi klase
Nestatički podaci i metodi	članovi objekta

Deklarisanje statičkih članova

```
class Test {  
    int x;                                // nestatička varijabla  
    static int xStatic;                   // statička varijabla  
  
    public void print() {                  // nestatička metoda  
        System.out.println("Hello, World");  
    }  
    public static void printStatic() {     // statička metoda  
        System.out.println("Hello, World");  
    }  
}
```

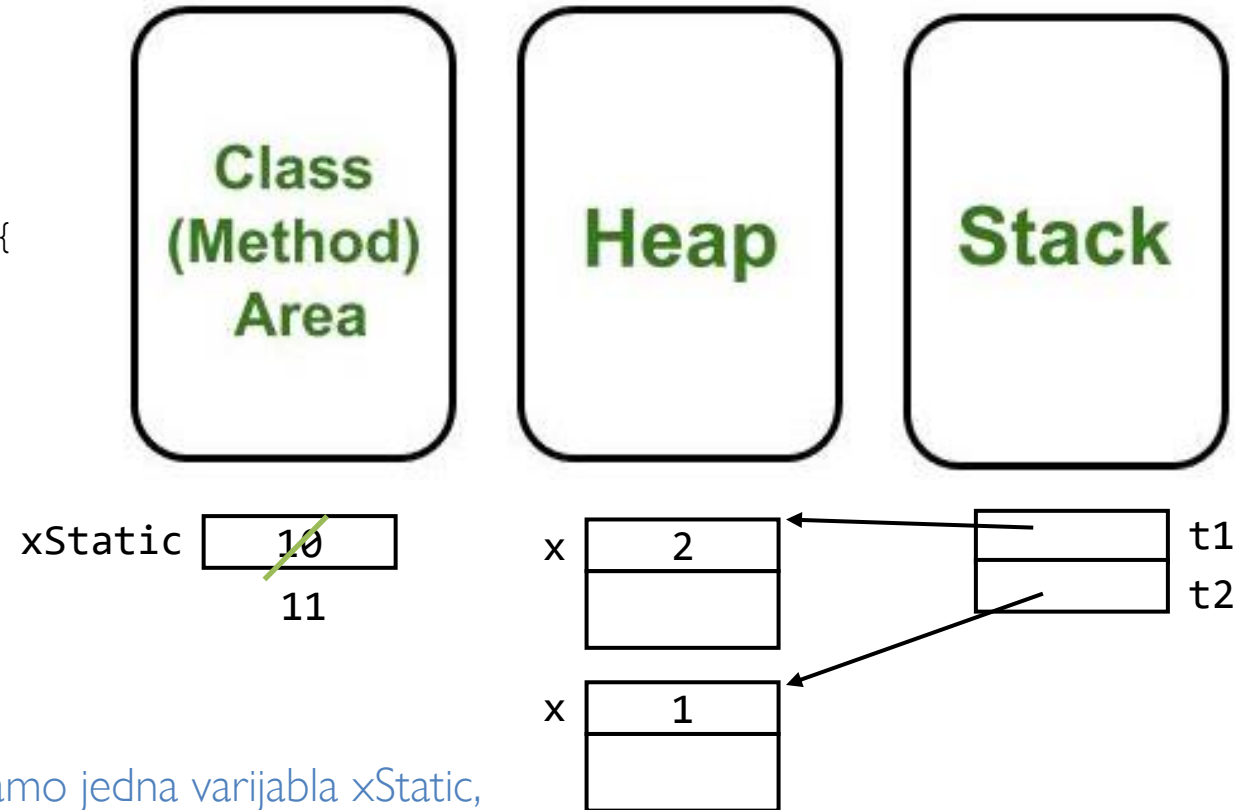
Da bismo koristili statičku varijablu (ili metod) neke klase **nije potrebno imati instancu te klase**.
Statičkim varijablama (i metodama) se pristupa sa:

`(ime_klase).(ime_statičke_varijable)`

* Za razliku od jezika kao što je C#, u Javi je dozvoljen i pristup statičkoj varijabli i preko reference objekta.

Statičke varijable

```
class Test {  
    int x = 0;  
    static int xStatic = 0;  
}  
  
public class TestStatika {  
    public static void main(String[] args) {  
        Test t1 = new Test();  
        Test t2 = new Test();  
        t1.x = 1;  
        t2.x = 2;  
        Test.xStatic = 10;  
        // ili t1.xStatic = 10;  
        // ili t2.xStatic = 10;  
        t2.xStatic = 11;  
    }  
}
```



Postoji samo jedna varijabla xStatic,
zajednička za sve objekte klase.
Svaka promena vidljiva svim objektima.

Statički metodi

- Metode deklarisanе sa `static` se tretiraju slično.
- Da bismo koristili statičku metodu neke klase nije potrebno imati instancu te klase. Statičkim metodama se pristupa sa:

`(ime_klase).(ime_statičke_metode)`

npr. `double x=Math.sqrt(2);` // `Math` je klasa iz paketa `java.lang` u kojoj je `sqrt()` statički metod

Statički metodi

```
class Test {  
    int x;                // nestatička javna varijabla  
    static int y;  
  
    public void print() { // nestatička metoda  
        System.out.println("Hello, World");  
    }  
    public static void print_static() { // statička metoda  
        System.out.println("Hello, World");  
    }  
    public static void main(String[] args) {  
        x=3;              //error  
        print();          //error  
        print_static();   //OK  
        new Test().x=5;    //OK  
        Test test=new Test(); //OK  
        test.print();      //OK  
    }  
}
```

Statički metodi ne dobijaju **this**!

Statičke metode **ne mogu koristiti nestatičke** metode iste klase direktno, niti pristupati nestatičkim varijablama.

Primer

Napisati definiciju tipa Predmet.

Predmet ima:

- string oznaku vrste predmeta
- jedinstven, automatski generisan identifikacioni broj
- metode kojim se dobija tip predmeta, kao i njegov ID.