Академија струковних
студија Шумадија
одсек у Крагујевцу

Увод у програмирање

Презентација 9

Академија струковних студија Шумадија

Одсек у Крагујевцу

Студијски програм Информатика

Крагујевац, 2020. година

Објектно оријентисано програмирање

- Подсетићемо како се дефинише класа, њени атрибути и методе посматрајући појам из реалног света – Кутија.
- Свака кутија има атрибуте: дужина, ширина и висина.
- Креираћемо два објекта у главној класи и задати вредности атрибутима.
- Запремину кутије рачунаћемо у одговарајућем методу класе.

```
public class Kutija {

    public int dužina;
    public int širina;
    public int visina;

    public int zapremina() {
        return dužina * širina * visina;
    }
}
```

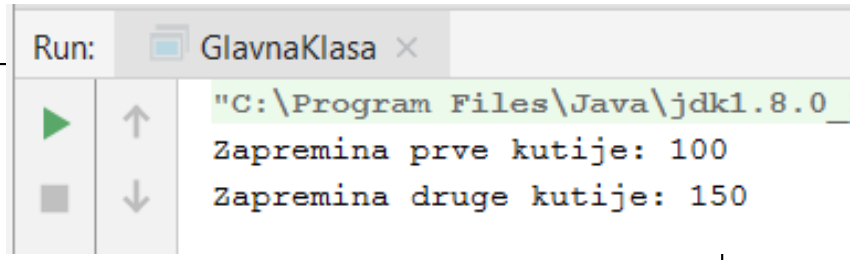
Објектно оријентисано програмирање

- У главној класи креирамо објекте помоћу којих задајемо вредности одговарајућим атрибутима и позивамо метод за израчунавање запремине.

```
public class GlavnaKlasa {
    public static void main(String[] args) {
        Kutija k1 = new Kutija();
        Kutija k2 = new Kutija();

        k1.dužina = 5;        k2.dužina = 10;
        k1.širina = 10;       k2.širina = 3;
        k1.visina = 2;        k2.visina = 5;

        System.out.println("Zapremina prve kutije: " + k1.zapremina());
        System.out.println("Zapremina druge kutije: " + k2.zapremina());
    }
}
```



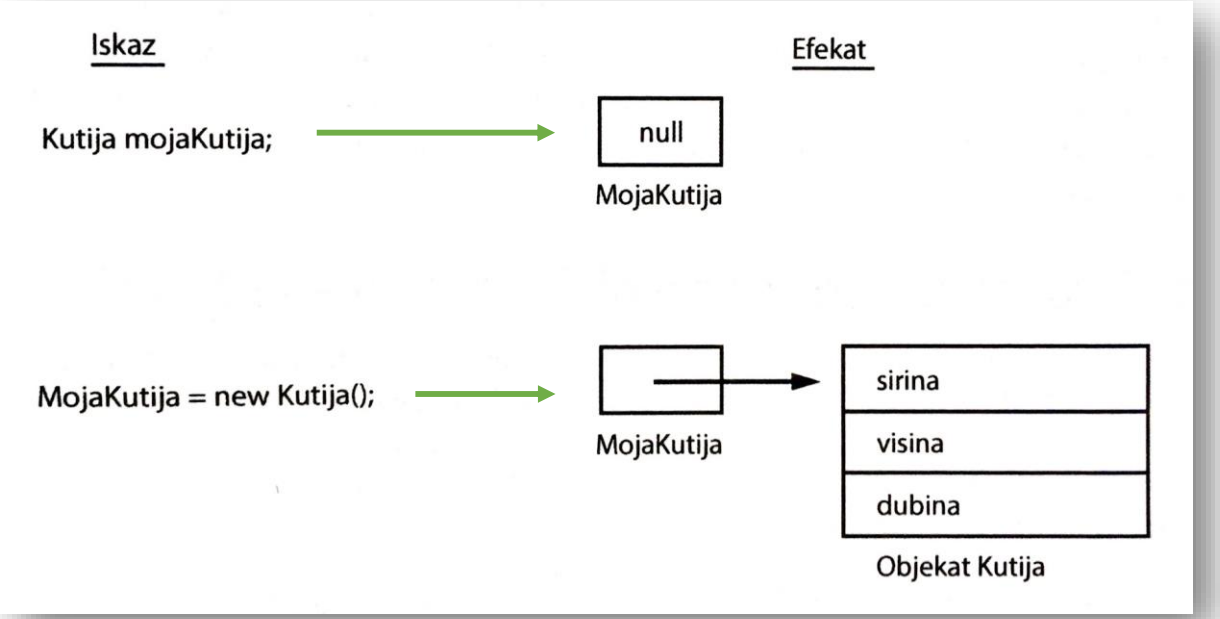
Објектно оријентисано програмирање

- Пошто смо направили класу "Кутија" ми смо тиме направили и нови тип података – класни тип.
- Овај тип података можемо да употребимо, као у примеру, за декларисање објеката тог типа – објекти "к1" и "к2".
- Објекти неке класе се праве у **две фазе**:
  - Прво се декларише променљива чији тип одговара класи. Ова променљива **не дефинише објекат**, већ упућује **референцу** на њега.
  - Други корак подразумева прављење стварног објекта **који се додељује** декларисаној променљивој.
- Прављење објекта се изводи помоћу оператора **`new`**.
- На претходном предавању је објашњено како се "понаша" овај оператор.
- Он виши алоцирање меморијског простора за креирани објекат и програму враћа референцу на тај блок меморије (на његову адресу).

Објектно оријентисано програмирање

- Референца се на крају тог процеса смешта у променљиву.
- Ово значи да се у Јави свим објектима меморија динамички додељује.
- Најчешће, приликом програмирање, обе фазе се спајају у један исказ.
- То, наравно, не мора да буде подразумевано правило:

```
Kutija mojaKutija; // декларисање референце на објекат
mojaKutija = new Kutija(); // додељивање објекта типа Kutija
```



Објектно оријентисано програмирање

- У првој фази, променљива **`mojaKutija`** још не упућује ни на један постојећи објекат.
- У другој фази креира се објекат коме се у меморији додељује простор, а променљивој **`mojaKutija`** се додељује референца на тај објекат.
- Након завршетка друге фазе, променљиву **`mojaKutija`** можете користити као да је објекат типа **`Kutija`**.
- У стварности, **променљива садржи само референцу** (меморијску адресу) стварног објекта типа **`Kutija`**.
- Ово је могуће управо због оператора **`new`** који динамички додељује меморију новокреираном објекту.
- Као што смо то већ помињали у претходном предавању, након оператора **`new`** се позива **конструктор класе**.

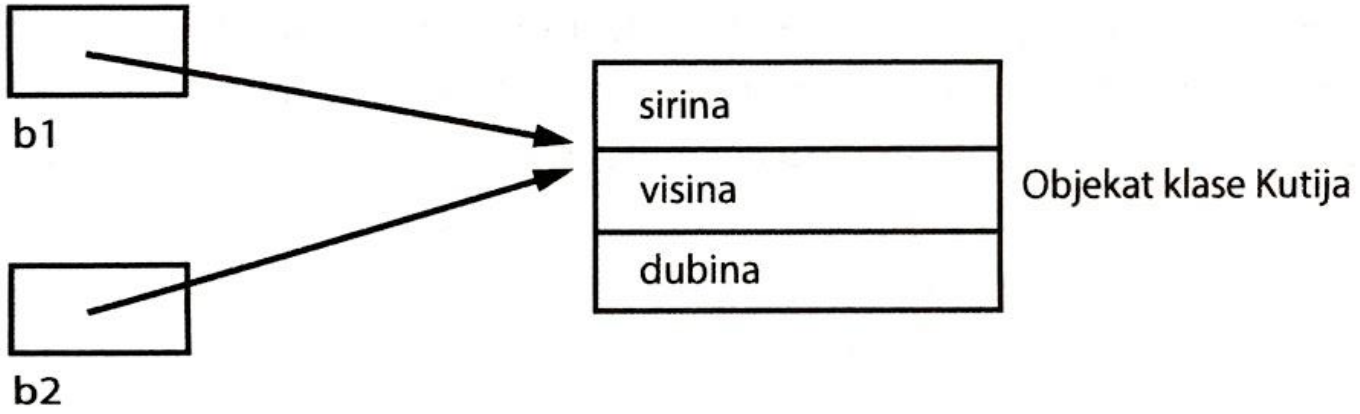
Објектно оријентисано програмирање

- Важно је разумети концепт да оператор **new** резервише меморију у тренутку извршавања програма.
- Предност овог приступа је то што програм током извршавања прави тачно онолико објеката колико их је у том тренутку потребно.
- Пошто је количина слободне меморије коначна, може се догодити (у теорији) да оператор **new** не може да додели меморију одређеном објекту и у том случају ће се генерисати изузетак током извршавања програма.
- За сада, док учимо програмирање поменута ситуација се скоро па никада неће догодити – тј. готово је немогуће изазвати је.
- Промењљиве које дају референце на објекте не понашају се приликом додељивања вредности онако како се понашају прости типови података.

```
Kutija b1 = new Kutija();
Kutija b2 = b1;
```

Објектно оријентисано програмирање

- У наведеном примеру, промењљивој "b2" се не додељује копија објекта "b1", већ обе промењљиве референцирају на исти објекат класе "Kutija".



- Иако промењљиве "b1" и "b2" садрже референце на исти објекат, оне нису ни на који начин међусобно повезане.
- Нпр. уколико промењљивој "b1" накнадно доделите неку другу вредност, тиме сте промењљиву "b1" само "откачили" од првобитног објекта, при чему се не мања ни објекат, нити промењљива "b2".

Објектно оријентисано програмирање

```
public static void main(String[] args) {

    Kutija k1 = new Kutija();

    k1.dužina = 5;
    k1.širina = 10;
    k1.visina = 2;

    Kutija k2 = k1;

    System.out.println(k1.zapremina());
    System.out.println(k2.zapremina());
}
```

```
Run:  GlavnaKlasa
"C:\Program Files\Java\jdk1
100
100
```

```
public class GlavnaKlasa {
    public static void main(String[] args) {
        Kutija k1 = new Kutija();
        k1.dužina = 5;
        k1.širina = 10;
        k1.visina = 2;

        Kutija k2 = k1;

        k1 = null;

        //System.out.println(k1.zapremina());
        System.out.println(k2.zapremina());
    }
}
```

```
Run:  GlavnaKlasa
"C:\Program Files\Java\jdk1.
100
```

Објектно оријентисано програмирање

- Шта су конструктори смо учили на претходном предавању.
- Сада ћемо проширити наш пример класе "Кутија" додавањем конструктора.
- Улога конструктора је да иницијализује објекат одмах приликом стварања објекта.
- Он иницијализује интерно стање објекта тако да објекат може одмах да се употреби.
- За разлику од примера са претходног предавања, овога пута ћемо направити конструктор класе "Кутија" како се он то иначе прави (није правило али је пожељно) у било којој класи.
- Овога пута ћемо употребити резервисану реч **`this`**.
- Резервисана реч **`this`** се може позвати унутар сваке методе ради референцирања текућег објекта.

Објектно оријентисано програмирање

- Приликом декларације промењљиве и креирања објекта класе морамо да унесемо све вредности за параметре који су наведени у конструктору.

```
public class Kutija {

    public int dužina;
    public int širina;
    public int visina;

    public Kutija(int dužina, int širina, int visina){
        this.dužina = dužina;
        this.širina = širina;
        this.visina = visina;
    }

    public int zapremina() {
        return dužina * širina * visina;
    }
}
```

```
public class GlavnaKlasa {
    public static void main(String[] args) {

        Kutija k1 = new Kutija();

        System.
        System.
    }
}
```

Kutija() in **Kutija** cannot be applied to:

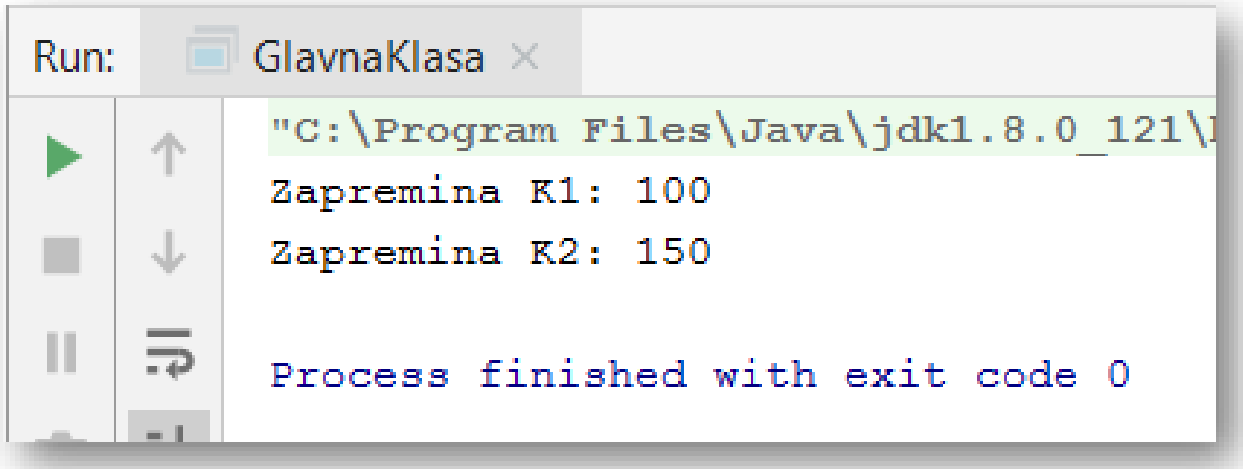
Expected Parameters:	Actual Arguments:
dužina:	**int**
širina:	**int**
visina:	**int**

Објектно оријентисано програмирање

```
public class GlavnaKlasa {
    public static void main(String[] args) {

        Kutija k1 = new Kutija(5,10,2);
        Kutija k2 = new Kutija(10, 3, 5);

        System.out.println("Zapremina K1: " + k1.zapremina());
        System.out.println("Zapremina K2: " + k2.zapremina());
    }
}
```



Објектно оријентисано програмирање

- Применом резервисане речи **`this`** можете имати **локалне променљиве**, укључујући и параметре метода, **са истим именом** као атрибути класе.
- Резервисану реч **`this`** овога пута користимо да би смо разрешили "сукобе" у именском простору који могу да настану између променљивих инстанце и локалних променљивих.
- По овом питању **не постоји јасан и коначан консензус** између програмера.
- Једна група инсистира на коришћењу различитих имена за формалне параметре, док друга група има потпуно супротан став и сматра да је употреба истих имена добра конвенција која доприноси разумљивости.
- Које ћете решење ви одабрати је потпуно на вама, ми ћемо на даље, у предавањима, користити принцип истих имена и употребу резервисане речи **`this`**.

Објектно оријентисано програмирање

Прва провера знања - предавања

Литература

Градиво осмог предавања :

- Поглавље 7:

https://singipedia.singidunum.ac.rs/izdanje/40716-osnove-java-programiranja

- Ако пратите препоручене видео лекције, градиво које смо обрадили у овом предавању се односи на видео лекцију 13.

https://www.youtube.com/playlist?list=PL-UTrxF0y8kK49N01V5ttb2Xaua7JfyXu

- Одабрана поглавља из књиге: Јава JDK9: Комплетан приручник
  - Аутор: *Herbert Schildt (може и старије издање JDK7).*
- Књига: Објектно оријентисани начин мишљења
  - Аутор: *Matt Weisfeld*
- Не морате да купујете наведене књиге.