JAVA ОБЕКТИ

Ненко Табаков
Пламен Танов
Технологическо училище “Електронни системи”
Технически университет – София

9 октомври 2008

JAVA ОБЕКТИ

Забележка: Тази лекция е адаптация на лекция от курса:

•6.092 Java Preparation for 6.170, Януари 2006
- Lucy Mendel
- Corey McCaffrey
- Rob Toscano
- Justin Mazolla Paluska
- Scott Osler
- Ray He

Интернет адрес:
http://ocw.mit.edu/OcwWeb/Electrical-Engineering-and-Computer-Science/6-092January--IAP--2006/CourseHome/index.htm

ПРЕПРАТКИ

- Препратки (references), също познати като псевдоними, сочат към обекти
- Препратката сочи към инстанция на даден клас (обект)
- Деклариране на препратка

```
Integer x;

тип име_на_променлива
```

ОБЕКТИ

- Класовете описват обекти
- Обектът е инстанция на даден клас
- За да се създаде обект трябва да се извика конструктор

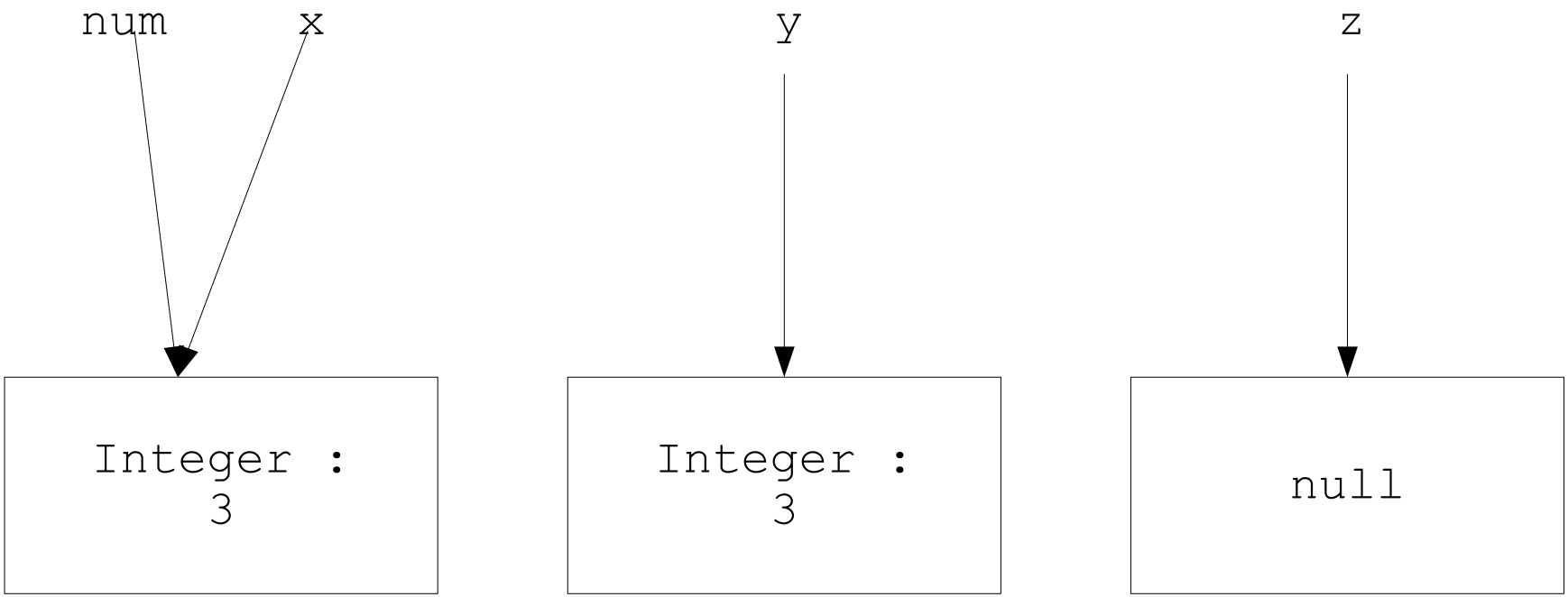
```
new Integer (3);
```

ПРИМЕР

```
public class AssignmentReview {
    public static void main(String[] args) {
        Integer num;
        num = new Integer(3);
        Integer x = num;
        Integer y = new Integer(3);
        Integer z;
    }
}
```

ПРЕДСТАВЯНЕ НА ***JAVA*** ХИЙП

JAVA хийп показва кои референции и обекти съществуват по време на изпълнение на програмата



ПРЕПРАТКИ КЪМ ***NULL***

- Ненасочените препратки сочат към ***null***
- ***null*** не е обект (няма полета, нито методи)
- Деклариране на препратка

```
Integer z;
z.intValue(); //този ред ще генерира изключение
              //(NullPointerException)
```

РАБОТА С ОБЕКТИ И ПРЕПРАТКИ

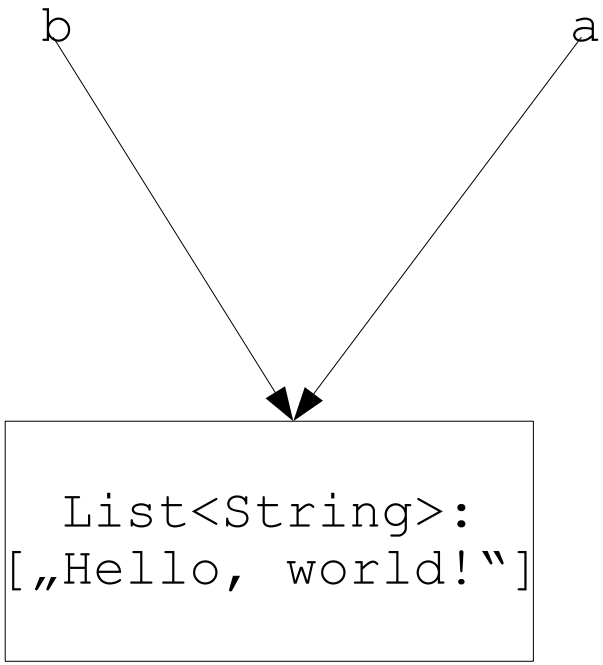
- Използвайте „=“ за да насочите препратка към обект
- Някои методи променят вътрешното състояние на обекта
- Може да имате няколко препратки към един и същ обект, така че се пазете от странични ефекти

ПРИМЕР

```
public  class MutationExample {
    public static void main(String[] args) {
    List<String> a = new ArrayList<String>();
    List<String> b = a; // b & a share the List
    a.add("Hello, world!");
    System.out.println(b);
    // Prints “Hello, world!”
    }
}
```

ПРЕПРАТКИ КЪМ ЕДИН И СЪЩ ОБЕКТ

JAVA хийп:



СТАТИЧНИ И НЕСТАТИЧНИ МЕТОДИ/ПОЛЕТА

- Методите и полетата могат да се декларират като статични
- Статичните методи/полета принадлежат на класа
- Нестатичните методи/полета принадлежат на обекта

ПРИМЕР
нестатично поле

```
public class Bean {
    public int beanCounter = 0;

    public Bean() {
        beanCounter++;
    }

    public static void main(String[] args) {
        new Bean();
        new Bean();
        Bean bean = new Bean();
        System.out.println(bean.beanCounter);
        // Prints "1"
    }
}
```

ПРИМЕР
статично поле

```
public class Bean {
    public static int beanCounter = 0;

    public Bean() {
        beanCounter++;
    }

    public static void main(String[] args) {
        new Bean();
        new Bean();
        Bean bean = new Bean();
        System.out.println(bean.beanCounter);
        // Prints "3"
    }
}
```

ПРИМЕР
нестатичен метод

```
public class Bean {
    private boolean planted = false;

    public void plantBean() {
        this.planted = true;
    }

    public static void main(String[] args) {
        Bean bean = new Bean();
        bean.plantBean(); // Invoked on instance
    }
}
```

ПРИМЕР
статичен метод

```
public class Bean {
   private boolean planted = false;

   public static void plantBean(Bean bean) {
      bean.planted = true;
   }

   public static void main(String[] args) {
      Bean bean = new Bean();
      Bean.plantBean(bean); // Invoked on class
      // bean.plantBean(bean); legal but inadvisable!
   }
}
```

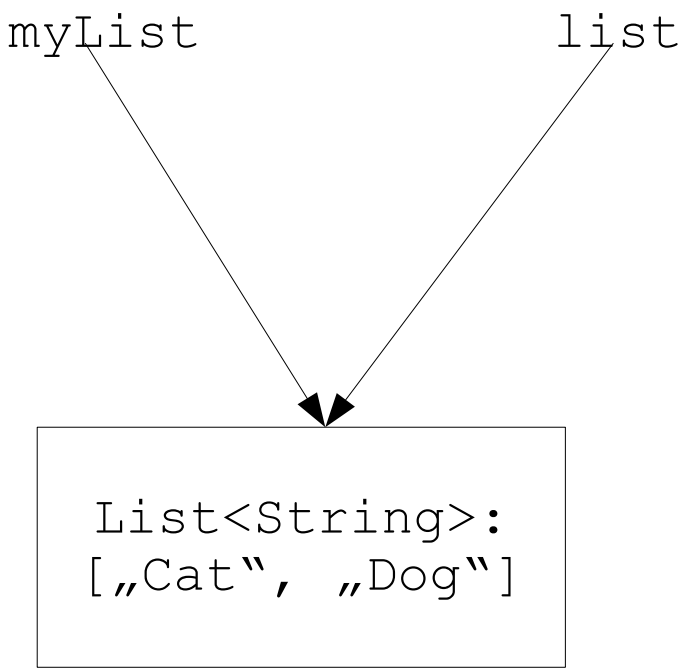
ПРИМЕР
обекти предавани по адрес

```
public static <T> void removeFirst(List<T> list) {
      list.remove(0);
}

public static void main(String[] args) {
   List<String> myList= new ArrayList<String>();
   myList.add("Cat");
   myList.add("Dog");
   removeFirst(myList);
   System.out.println(myList); // Prints "[Dog]"
}
```

ОБЕКТИ ПРЕДАВАНИ ПО АДРЕС

JAVA хийп:



ВИДИМОСТ

- {} - определят региона на видимост
- Препратките съществуват от момента на създаването си докато не излязат от региона на видимост
- Полетата могат да се използват в рамките на класа
- Параметрите могат да се използват в рамките на метода

ПРИМЕР
видимост на променливи

```
public class ScopeExample {
    private int globalField;

    public int method(int parameter) {
        int localVar1;
        if (globalField > 0) {
            int x;
        }
        int localVar2;
    }
}
```

ПРИМЕР
видимост на променливи

```
public class ScopeExample {
   private int globalField;

   public int method(int parameter) {
      int globalField; // Legal, but hides field!
      int localVar;
      if (this.globalField > 0) { // Accesses field
         int x;
      }
      int localVar; // Illegal: same scope
   }
}
```

СЪВЕТИ

- Препратките само сочат към обекти. Пазете се от препратки, които сочат към ***null***
- Не извиквайте статични методи чрез инстанции
- Когато предавате обект по адрес, понякога е добре да се прави копие, в случай, че не искаме обектът да се променя
- Минимизирайте видимостта на променливите колкото се може повече (т.е. не правете глобални променливи)