



# Java 核心技术(高阶)

第一章 Java语法糖

第二节 语法糖(1)

华东师范大学 陈良育



# for/for-each(1)

- for vs for-each

```
double[] nums = new double[5];
for(int i=0;i<nums.length;i++)
{
    nums[i] = Math.random();
}
for(double item : nums)
{
    System.out.println(item);
}
```

```
Set<String> hs = new HashSet<String>();
hs.add("I");
hs.add("Love");
hs.add("Java");

for(Iterator<String> i = hs.iterator(); i.hasNext(); )
{
    System.out.println(i.next());
}

for(String i:hs)
{
    System.out.println(i);
}
```



# for/for-each(2)

- for vs for-each
  - for-each 从JDK5.0开始引入
  - for-each 语法更简洁
  - for-each 避免越界错误
  - for 可以删除元素，for-each不可以删除/替换元素
  - for-each遍历的时候，是不知道当前元素的具体位置索引
  - for-each只能正向遍历，不能反向遍历
  - for-each不能同时遍历2个集合
  - for和for-each性能接近



# 枚举类型(1)

- 枚举变量：变量的取值只在一个有限的集合内
  - 如性别、星期几、颜色等
- Java5推出enum类型
  - enum关键字声明枚举类，且都是Enum的子类(但不需写extends)
  - enum内部有多少个值，就有多少个实例对象
  - 不能直接new枚举类对象

```
public enum Size {  
    SMALL, MEDIUM, LARGE, EXTRA_LARGE;  
}
```

```
Size s1 = Size.SMALL;  
Size s2 = Size.SMALL;  
Size s3 = Size.LARGE;
```

```
System.out.println(s1 == s2); //true  
System.out.println(s1 == s3); //false
```





# 枚举类型(2)

- Java的enum类型

- 除了枚举的内容，还可以添加属性/构造函数/方法
- 构造函数只能是package-private(default)或者private，内部调用

```
enum Fruit
{
    APPLE(10), ORANGE(8);
    private int price;

    Fruit(int price) {
        this.price = price;
    }

    public int getPrice() {
        return this.price;
    }
}
```

```
Fruit a1 = Fruit.APPLE;
System.out.println("Price is " + a1.getPrice());
```



# 枚举类型(3)

- Java的enum类型

- 所有的enum类型都是Enum的子类, 也继承了相应方法
- ordinal()返回枚举值所在的索引位置, 从0开始
- compareTo()比较两个枚举值的索引位置大小
- toString()返回枚举值的字符串表示
- valueOf()将字符串初始化为枚举对象
- values()返回所有的枚举值

# 总结



- for-each循环(5): 避免数组越界, 提倡使用
- 枚举(5): 限定取值集合, 减少实例对象, 提倡使用



谢谢!