

自动装箱和枚举

自动装箱

有时需要将int这样的基本数据类型转换成引用数据类型，在JDK1.5之后可以自动完成转换

自动装箱

由基本数据类型自动转换为引用数据类型

比如：

```
Integer i = 10;
```

等价于：

```
Integer i = new Integer(10);
```

自动拆箱

由引用数据类型自动转换为基本数据类型

比如：

```
Double d = 10.5;  
double num = d;
```

相当于

```
double num = d.doubleValue();
```

基本数据类型的包装类

基本类型	包装类
boolean	Boolean
byte	Byte
char	Character
short	Short
int	Integer
long	Long
float	Float
double	Double

注意：基本数据类型没有父类

枚举

概念

enum 像一个特殊的 class，用 enum 声明的其实就相当于一个类

只是这个类提前创建了一系列对象，叫做枚举值，用逗号隔开

枚举值

枚举值只能是 `public static final` 修饰，所有枚举值都应该大写

比如：

```
public enum Color{
    RED, GREEN, BLUE;
    private Color(){
        System.out.println("执行了构造方法");
    }
}
```

事先创建好三个枚举值，分别是：RED，GREEN，BLUE

相当于：

```
public static final Color RED = new Color();
public static final Color GREEN = new Color();
public static final Color BLUE = new Color();
```

枚举中的构造方法

1. 只能是 `private` 的
2. 只有在创建枚举值的时候被执行

获取所有枚举值

```
enum Person{
    P1("张三",12),P2("李四",13),P3("王五",14);
    private String name;
    private int age;
    private Person(String name,int age){
        this.name = name;
        this.age = age;
    }

    @Override
    public String toString() {
        return name+"---"+age;
    }
}
```

上面枚举类中获取所有枚举值：

```
Person[] persons = Person.values();
```

ForEach循环

```
for(Person p : persons){
    System.out.println(p);
}
```

1. 必须全部遍历
2. 不可以指定循环次数
3. 不可以追踪下标