JAVA

基本数据类型的封装类

本章导读

基本数据类型封装类概述

基本数据类型封装类的变量和方法

基本数据类型封装类举例

本章导读

基本数据类型封装类概述

基本数据类型封装类的变量和方法

基本数据类型封装类举例

- Java是一门完全面向对象的语言，倡导“一切都是类和对象”的理念

- 但是，基本数据类型不是类，不能产生对象

- Java引进基本数据类型的封装类，以统一的机制处理所有的数据类型

整数类型的封装类

数据类型	封装类
byte	Byte
short	Short
int	Integer
long	Long

浮点类型的封装类

数据类型	封装类
float	Float
double	Double

字符类型的封装类

数据类型	封装类
char	Character

布尔类型的封装类

数据类型	封装类
boolean	Boolean

- 封装类体现了Java的纯面向对象的特点

- 封装类对于完成基本数据类型数据的变换操作提供了方便

例 — 将数字字符串转换为整数或者浮点数

本章导读

基本数据类型封装类概述

基本数据类型封装类的变量和方法

基本数据类型封装类举例

Integer类的类变量

MAX_VALUE和MIN_VALUE

在长度为100的整数数组array中找出最大值

```
int max=Integer.MIN_VALUE;
for(int i=0;i<100;i++)
    if(max<array[i])
            max=array[i];
```

Integer类的构造方法

- public Integer(int value)

  用一个int类型的量生成一个Integer对象

```
Integer i=new Integer(5);
```

- 自动装箱（AutoBoxing）

  Integer i=5;

Integer类的构造方法

public Integer(String s)

用一个字符串生成一个Integer对象

```
Integer i=new Integer("05");
```

Integer类的非静态方法

- public int intValue()

将Integer对象转换成int类型的值

```
Integer i=5;
int value=i.intValue();
```

- 自动拆箱（AutoUnboxing）

```
Integer i=5;
int value=i;
```

Integer类的非静态方法

public String toString()

将Integer对象转换成一个字符串

```
Integer i=5;
String s=i.toString();
```

Integer类的静态方法

public static int parseInt(String s)

把字符串转换成int类型的值

```
int i=Integer.parseInt("-05");
```

Integer类的静态方法

public static Integer valueOf(String s)

把字符串转换成Integer对象

```
Integer i=Integer.valueOf("2007");
```

Character类

构造方法

public Character(char c)

用一个char类型的量生成一个Character对象

Character类中的常用类方法

public static boolean isDigit(char ch)

public static boolean isLetter(char ch)

public static boolean isLetterOrDigit(char ch)

Character类

Character类中的常用类方法

```
public static boolean isLowerCase(char ch)

public static boolean isUpperCase(char ch)

public static char toLowerCase(char ch)

public static char toUpperCase(char ch)

public static boolean isSpaceChar(char ch)
```

Character类举例

```
public class Example{
   public static void main(String args[ ])  {
      char a[ ]={'a','b','c','D','E','F'};
      for(int i=0;i<a.length;i++)
      {
         if(Character.isLowerCase(a[i]))
              a[i]=Character.toUpperCase(a[i]);
         else if(Character.isUpperCase(a[i]))
              a[i]=Character.toLowerCase(a[i]);
       }
     for(int i=0;i<a.length;i++)
              System.out.printf("%6c",a[i]);
    }
}
```

本章导读

- 基本数据类型封装类概述
- 基本数据类型封装类的变量和方法
- 基本数据类型封装类举例

基本数据类型的封装类举例

```
public class TestCapsulation{
    public static void main(String[] args){
            Boolean b = new Boolean("5");
            Byte by = new Byte((byte)5);
            Short s = new Short("5");
            Integer i = new Integer("5");
            Long l = new Long(5);
            Float f = new Float("5");
            Double d = new Double("5");
            Character c = new Character('5');
            System.out.println("Boolean: " + b+ " Byte: " + by);
            System.out.println("Short: " + s+ "  Integer: " + i);
            System.out.println(" Long: " + l+ "  Float: " + f);
            System.out.println(" Double: " + d+ "  Character: " + c);
    }
}
```

b.toString()

基本数据类型的封装类举例

```
public class TestCapsulation{
    public static void main(String[] args){
             Boolean b =true;
             Byte by = 5;
             Short s = 5;
             Integer i =5;
             Long l = 5L;
             Float f=5.0f;
             Double d = 5.0;
             Character c = '5';
             System.out.println("Boolean: " + b+ " Byte: " + by);
            System.out.println("Short: " + s+ "  Integer: " + i);
            System.out.println(" Long: " + l+ "  Float: " + f);
            System.out.println(" Double: " + d+ "  Character: " + c);
    }
}
```