

# 泛型接口

## 接口

```
package com.southwind.demo;

public interface MyInterface<T> {
    public T getValue();
}
```

实现泛型接口有两种方式：

- 实现类在定义时继续使用泛型标识

```
package com.southwind.demo;

public class MyInterfaceImpl<T> implements
MyInterface {

    private T obj;

    public MyInterfaceImpl(T obj) {
        this.obj = obj;
    }

    @Override
    public T getValue() {
        return this.obj;
    }
}
```

- 实现类在定义时直接给出具体的数据类型

```
package com.southwind.demo;

public class MyInterfaceImpl2 implements
MyInterface<String> {

    private String obj;

    public MyInterfaceImpl2(String obj) {
        this.obj = obj;
    }

    @Override
    public String getValue() {
        return this.obj;
    }
}
```

```
package com.southwind.demo;

public class Test {
    public static void main(String[] args) {
        MyInterfaceImpl myInterface = new
MyInterfaceImpl<String>("接口");
        String val = (String)
myInterface.getValue();

        MyInterfaceImpl2 myInterface1 = new
MyInterfaceImpl2("接口");
        val = myInterface1.getValue();
    }
}
```

# Java 实用类

- 枚举

枚举 Enum，是一种有确定值区间的数据类型，本质上就是一个类，具有简洁、安全、方便等特点。

枚举的值被约束到了一个特定的范围内，只能从这个范围以内取值。

为什么要有枚举？

因为在描述某些对象的属性时，该属性的值不能随便定义，必须在某个特定的区间内取值。

出于对数据的安全性考虑，类似这种有特定取值范围的数据我们就可以使用枚举来描述。

枚举指由一组常量组成的类型，指定一个取值区间，我们只能从该区间中取值。

```
final class Week extends Enum{
    public static final Week MONDAY;
    public static final Week TUESDAY;
    public static final Week WEDNSDAY;
    public static final Week THURSDAY;
    public static final Week FRIDAY;
    public static final Week SATURDAY;
    public static final Week SUNDAY;
    private static final Week $VALUES[];

    static{
        MONDAY = new Week("MONDAY",0);
        TUESDAY = new Week("TUESDAY",1);
```

```

    WEDNSDAY = new Week( "WEDNSDAY", 2 );
    THURSDAY = new Week( "THURSDAY", 3 );
    FRIDAY = new Week( "FRIDAY", 4 );
    SATURDAY = new Week( "SATURDAY", 5 );
    SUNDAY = new Week( "SUNDAY", 6 );
    $VALUES[ ] = (new Week[ ]{

        MONDAY, TUESDAY, WEDNSDAY, THURSDAY, FRIDAY, SATURDAY, SU
        NDAY
        })
    }

    public static Week[ ] values(){
        return $VALUES.clone();
    }

    public static Week valueOf( String s ){
        return Enum.valueOf(s);
    }

    private Week( String s, int i ){
        super(s, i);
    }

}

```

## Math

Math 类为开发者提供了一系列的数学方法，同时还提供了两个静态常量 E（自然对数的底数）和 PI（圆周率）。

```
package com.southwind.demo4;

public class Test {
    public static void main(String[] args) {
        System.out.println("常量E"+Math.E);
        System.out.println("常量PI"+Math.PI);
        System.out.println("9的平方根"+Math.sqrt(9));
        System.out.println("8的立方根"+Math.cbrt(8));
        System.out.println("2的3次方"+Math.pow(2,3));
        System.out.println("较大值"+Math.max(6.5,1));
        System.out.println("-10.3的绝对
值"+Math.abs(-10.3));
        System.out.println(Math.ceil(10.000001));
        System.out.println(Math.floor(10.999999));
        System.out.println((int)(Math.random()*10));
        System.out.println(Math rint(5.4));
    }
}
```

## Random

用来产生随机数的类，并且可以任意指定一个区间，在此区间范围内产生一个随机数。

方法	描述
public Random()	创建一个无参的随机数构造器，使用系统时间作为默认种子
public Random(long seed)	使用 long 数据类型的种子创建一个随机数构造器
public boolean nextBoolean()	返回一个 boolean 类型的随机数
public double nextDouble()	返回一个 double 类型的随机数，0.0 - 1.0 之间
public float nextFloat()	返回一个 float 类型的随机数，0.0 - 1.0 之间
public int nextInt()	返回一个 int 类型的随机数
public int nextInt(n)	返回一个 int 类型的随机数，0-n之间
public long nextLong	返回一个 long 类型的随机数，0-1 之间

```

package com.southwind.demo5;

import java.util.Random;

public class Test {
    public static void main(String[] args) {
        Random random = new Random();
        //生成订单编号（时间戳+随机数）
        for (int i = 1; i <= 10000; i++) {
            //随机生成一个六位数

```

```
        System.out.println("订单"+i+"的编号  
是："+System.currentTimeMillis()+random.nextInt(10000  
0)+100000);  
    }  
  
}  
  
}
```