## 日期和时间处理：

### 获取当前时间：

Calendar c1 = Calendar.*getInstance*();

System.*out*.println(c1.get(Calendar.*YEAR*)+"年"+(c1.get(Calendar.*MONTH*)+1)+

"月"+c1.get(Calendar.*DAY\_OF\_MONTH*)+"日"+c1.get(Calendar.*HOUR*)+

":"+c1.get(Calendar.*MINUTE*)+":"+c1.get(Calendar.*SECOND*));

### 日期增加

// 增加天数为230

c1.add(Calendar.*DAY\_OF\_YEAR*,230);

### 日期格式互相转换

SimpleDateFormat format1 = **new** SimpleDateFormat("yyyy 年 MM月 dd日 hh时 mm分 ss秒");

SimpleDateFormat format2 = **new** SimpleDateFormat("yyyy/MM/dd hh:mm:ss");

System.*out*.println(format1.format(**new** ~~Date~~(2014,12,11,23,45,20)));

System.*out*.println(format2.format(**new** ~~Date~~(2014,12,11,23,45,20)))

## 数值处理

### 随机数字

// nextInt中的数字是0到int的单位，后面加上的数字可以确定开始的范围

Random r = new Random();

show(Integer.toString(r.nextInt(90)+10));

## 数组

### 常用数组

ArrayList：基于数组结构

LinkedList: 基于链表结构，一般情况下，用ArrayList就可以了，如果涉及到频繁的插入和删除元素，这个时候考虑使用LinkedList.

HashSet：基于set的实现,拒绝重复对象

TreeSet：检索和存储更快速

Hashtable: 基于陈旧的Dictionary类的

Hashmap: hashtable的替代品

遍历：

HashMap<String, String> map = **new** HashMap<String, String>();

map.put("a","123");

map.put("b","456");

map.put("c","178");

Iterator iter = map.entrySet().iterator();

**while**(iter.hasNext())

{

Map.Entry<String, String> str = (Map.Entry<String, String> )iter.next();

*show*("k:" + str.getKey() + "\*\*\*v:"+str.getValue());

}

TreeMap:

## 语法

### 遍历:

一般遍历

**for** (Object item : al) {

*show*(item);

}

使用Iterator遍历

LinkedList al = **new** LinkedList();

al.add("123");

al.add("456");

al.add("178");

Iterator iter = al.iterator();

**while**(iter.hasNext())

{

String str = (String)iter.next();

*show*(str);

}