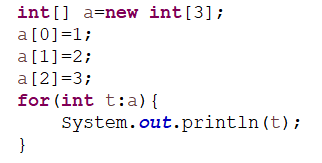
## 无下表的循环

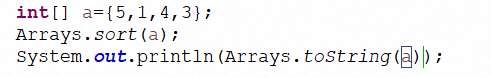


数组可以换成任何集合类型，如果不知道一个集合类型怎样遍历，那就用这个吧

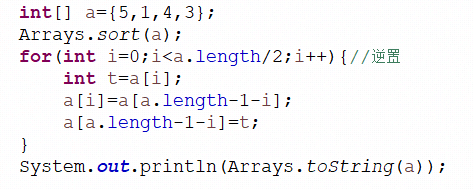
## 自己创建数组的升序与降序

**自己创建的数组有基本类型的数组，也类数组。基本类型的数组不用定义规则，但是，类数组一定要定义排序规则。**

升序

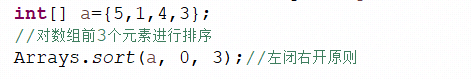


降序



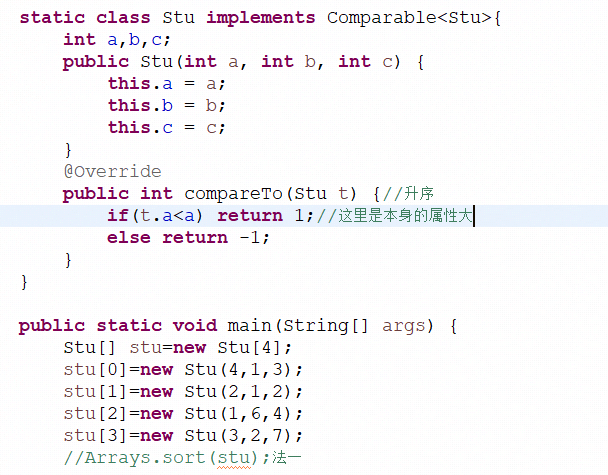
注意，二分查找法必须是数组排完序之后才能进行查找，无需的数组不能用二分查找法。

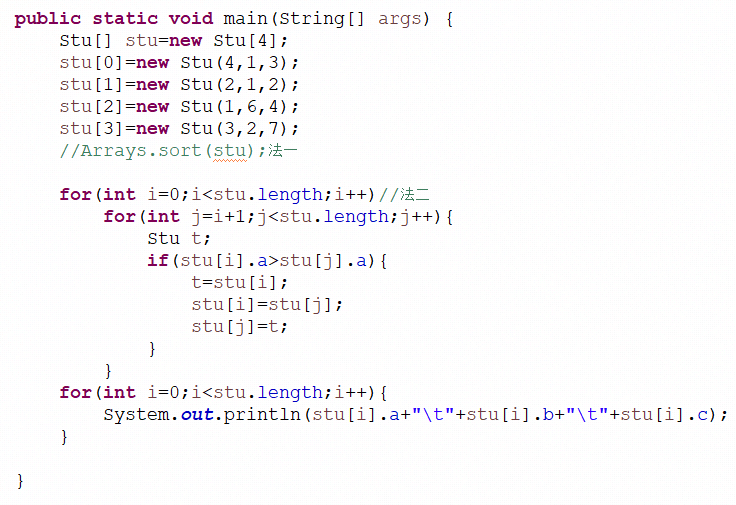
**对数组前3个元素进行排序**



类数组

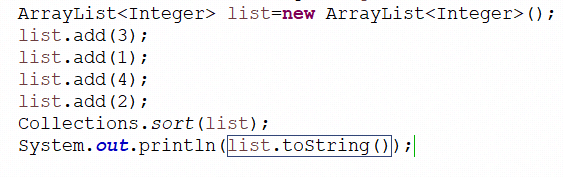
注意：**返回1代表放前面**



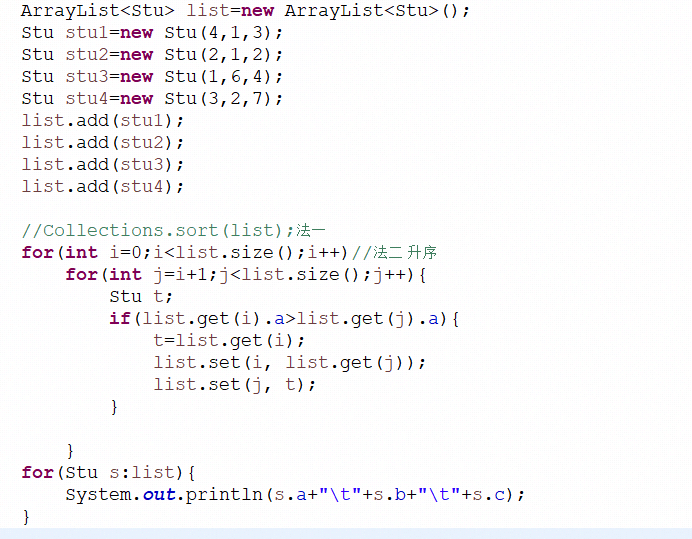


## 3、ArrayList数组的排序

ArrayList的整数排序，不需要定义规则

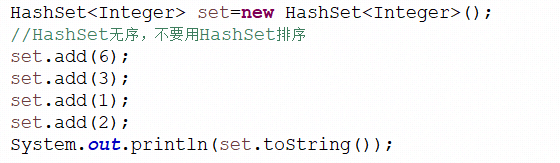


类排序要定义Comparable规则，这里省略



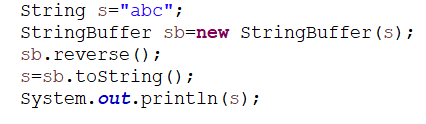
## 4、HashSet简单介绍

HashSet无序，不要用HashSet排序，用到时只要用HashSet不重复的特性就可以了



**不要用map，用的时候自己构建数据结构**

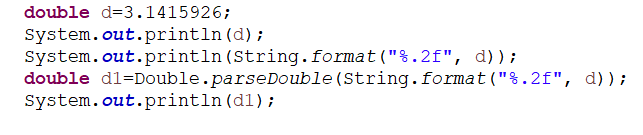
## 6、字符串的逆置



## 7、保留几位小数

**要注意两个，一个是 % ，另一个是 .2f ，注意没有0**

**这个是四射五入，哈哈，多好**

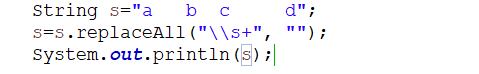


## 8、字符串分隔(重要)

Replace和replaceAll效果一样，replaceAll只是多了一只支持正则表达式的功能，**所以以后要用replaceAll**

**一定要记住\\s+，这是替换所有空格的表达式**

（1）把所有的空格替换为一个空格



（2）split的用法

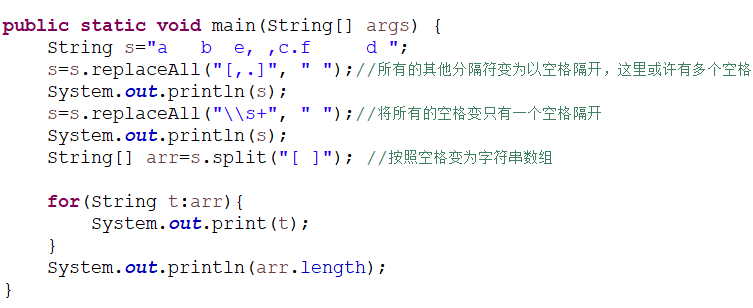
**所有的split里面用[ ]**，空格里面就添加一个空格

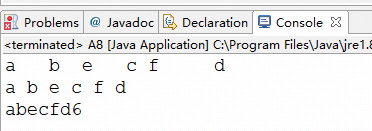
多个分隔字符都写在[ ]里面，**不需要加逗号，不需要空格隔开**

如s.split("[,.]");

如果分隔符单个分布还好直接可以用s.split("[,.]")，如果是多个并且是无规则分布，首先得用replaceAll替换为有规则的（一般替换为一个空格隔开），再使用split

**split("[,.]")里面的内容和replaceAll通用**



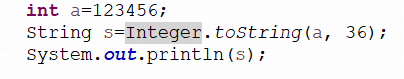


## 9、十进制与任意进制互转

**注意：所有的转换都是Integer开头的**

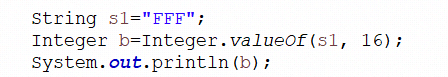
**十进制转化为任意进制**

**Integer.toString**，第一个参数是一个十进制数，第二个参数是要转化进制的基数



**任意进制转化为十进制**

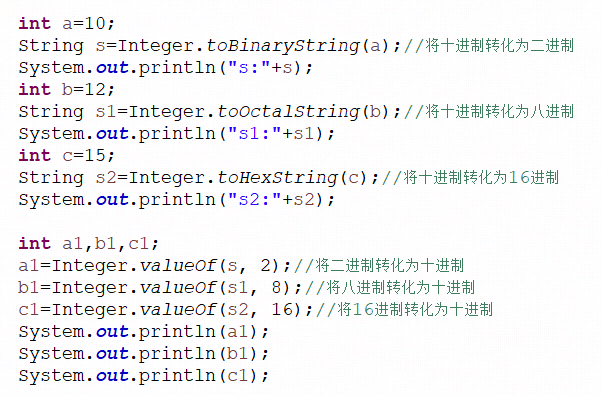
**Integer.valueOf，**第一个参数是一个带转化的字符串，第二个参数是第一个参数所属于的进制



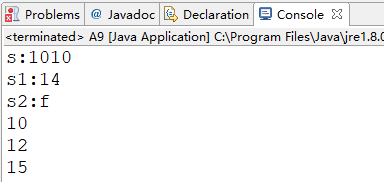
**另外，其他任意进制互转可以用十进制为中介**

**Integer.valueOf和Integer.parseInt功能一样，都是讲其他进制转化为十进制，基数都在后面**

下面可以不看



输出结果



## 11、先接收单个字符，再接收nextLine时，无法输入

**无论是nextInt()还是next()，后面跟一个nextLine，都会输入空行。**

原因是输入回车时证明一行结束，而在我们输入一个整数并敲击回车后，nextLine()就读取到了这一个回车符。解决方法：

我们可以专门再读取这一个回车，即用sc. nextLine()，不接收就行了。

## 12、java输入char类型

//java输入char类型

String s=*sc*.next();

**char** c=s.charAt(0);

System.***out***.println(c);

## 13、java将字符转化为对应的ASCII

注意，**java和c在char和int转换之间完全一样**

**‘\0’的Ascii是0**

如下完全正确

**char** c='a';

**int** i=c;//i为97

**char** b=(**char**) i;//b为'a'

字符转化为对应的ASCII也可以这样转化

**char** c='a';

**int** a=Integer.*valueOf*(c);

System.***out***.println(a);

## 14、关于next与nextLine及hasNext与hasNextLine(重要)

next()与nextLine有以下区别：

next（）一定要读取到有效字符后才可以结束输入，对输入有效字符之前遇到的空格键、Tab键或Enter键等结束符，next（）方法会自动将其去掉，只有在输入有效字符之后，next（）方法才将其后输入的空格键、Tab键或Enter键等视为分隔符或结束符。简单地说，next方法不能得到带空格的字符串。

而nextLine（）方法的结束符只是Enter键，即nextLine（）方法返回的是Enter键之前的所有字符，它是可以得到带空格的字符串的,并且，**nextLine（）也可以读入换行符。**

所以，**如果要读入换行符，就用hasNextLine。**

## 15、判断是否为瑞年

((year%4==0&&year%100!=0)||year%400==0)

## 16、StringBuffer的getChars方法

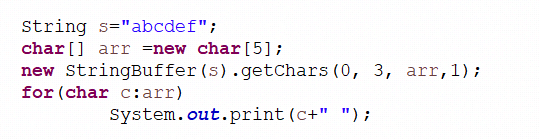
public void **getChars**(int srcBegin,

int srcEnd,

char[] dst,

int dstBegin)

dstBegin表示从这一位置开始填充



输出结果为

