**StringBuffer类：**

除了可以使用String类表示字符串之外，StringBuffer类也可以表示字符串。但两者有 明显的区别：

1）String类对象使用“+”进行字符串的连接，而StringBuffer类对象使用append( )方 法进行字符串的连接。

2）String类对象的内容一旦声明便不可改变，如果要改变，只能通过修改引用地址改 变。而StringBuffer类对象不需要修改引用地址就可以改变。

3）String类对象引用传递进行修改后原值并不会发生改变。而StringBuffer类对象 引用传递后原值会发生改变。

**StringBuffer类常用的操作：**

1. 使用append( )方法进行字符串的连接，此方法返回一个 StringBuffer类实例。
2. 使用insert( )方法在指定的下标处插入内容。
3. 使用reverse( )方法进行反转。
4. 使用toString( )方法将StringBuffer类对象变为String类对象。
5. 使用replace( )方法对指定范围的内容用其他内容进行替换。
6. 使用substring( )方法截取指定范围的内容。
7. 使用delete( )方法删除指定范围的内容。
8. 使用indexOf( )方法查找指定的内容是否存在，如存在，返回内容开始的下标；如果 不存在，返回-1。

**StringBuffer类的应用场景：**

对于频繁修改字符串内容的地方，最好使用StringBuffer类完成。

**String类、StringBuffer类、StringBuilder类的区别：**

String类、StringBuffer类、StringBuilder都可以表示字符串。

1. String类的内容一旦声明便不可改变，而StringBuffer类和StringBuilder类声明的内 容可以改变。
2. StringBuffer类中提供的方法都是同步方法，属于安全的线程操作；

StringBuilder类中提供的方法都是异步方法，属于非安全的线程操作。

**Runtime类：**

Runtime类表示运行时操作类，利用此类可以取得JVM的内存信息，也可以利用此 类产生新的操作系统进程对象。

**Runtime类对象的实例化：**

要想实例化Runtime类对象，只能通过类名称调用getRuntime( )方法。

**使用Runtime类得到JVM的内存信息：**

使用Runtime类可以得到JVM的内存信息，包括JVM的最大内存、空闲内存等，并且 还可以运行垃圾回收，释放JVM内存空间。

**Runtime类与Process类：**

通过Runtime类中的exec( )方法可以运行本机的可执行程序。与Process类搭配的话还 可以控制可执行程序的运行时间。