**Java API**

* String类和StringBuffer类
  + String类
    - 一 . String两种初始化方式1 . String str1= “abc”;//String类特有的创建字符对象的方式，更高效在字符串缓冲区中检测”abc”是否存在 若存在则不重复创建，将地址赋值给str1. 若不存在，则在字符串缓冲区中创建对象并赋地址给str1.2 . 利用String的构造方法初始化字符串对象String() Initializes a newly created String object so that it represents an empty character sequence.String(byte[] bytes) Constructs a new String by decoding the specified array of bytes using the platform's default charset.String(String original) Initializes a newly created String object so that it represents the same sequence of characters as the argument; in other words, the newly created string is a copy of the argument string.
  + String类的常见操作
    - 1 . indexOfpublic int indexOf(int ch)Returns:the index of the first occurrence of the character in the character sequence represented by this object, or -1 if the character does not occur.2 . lastIndexOfpublic int lastIndexOf(int ch)Returns:the index of the last occurrence of the character in the character sequence represented by this object, or -1 if the character does not occur.3 . indexOfpublic int indexOf(String str)Returns:if the string argument occurs as a substring within this object, then the index of the first character of the first such substring is returned; if it does not occur as a substring, -1 is returned.4 . lastIndexOfpublic int lastIndexOf(String str)Returns:if the string argument occurs one or more times as a substring within this object, then the index of the first character of the last such substring is returned. If it does not occur as a substring, -1 is returned.5 . charAtpublic char charAt(int index)Returns:the char value at the specified index of this string. The first char value is at index 0.6 . endsWithpublic boolean endsWith(String suffix)Returns:true if the character sequence represented by the argument is a suffix of the character sequence represented by this object; false otherwise. Note that the result will be true if the argument is the empty string or is equal to this String object as determined by the equals(Object) method.7 . lengthpublic int length()Returns:the length of the sequence of characters represented by this object.8 . equalspublic boolean equals(Object anObject)/\*\*Compares this string to the specified object. The result is true if and only if the argument is not null and is a String object that represents the same sequence of characters as this object.\*/Returns:true if the String are equal; false otherwise.9 . isEmptypublic boolean isEmpty()Returns true if, and only if, length() is 0.10 . startsWithpublic boolean startsWith(String prefix)Returns:true if the character sequence represented by the argument is a prefix of the character sequence represented by this string; false otherwise. Note also that true will be returned if the argument is an empty string or is equal to this String object as determined by the equals(Object) method.11 . containspublic boolean contains(CharSequence s)Returns true if and only if this string contains the specified sequence of char values.12 . toLowerCasepublic String toLowerCase()Converts all of the characters in this String to lower case using the rules of the default locale. This is equivalent to calling toLowerCase(Locale.getDefault()).13 . toUpperCasepublic String toUpperCase()Converts all of the characters in this String to upper case using the rules of the default locale. This method is equivalent to toUpperCase(Locale.getDefault()).14 . valueOfpublic static String valueOf(Object obj)Returns the string representation of the Object argument.15 . toCharArraypublic char[] toCharArray()Converts this string to a new character array.16 . replacepublic String replace(char oldChar, char newChar)Returns:a string derived from this string by replacing every occurrence of oldChar with newChar.17 . splitpublic String[] split(String regex, int limit)Returns:the array of strings computed by splitting this string around matches of the given regular expression18 . substringpublic String substring(int beginIndex)Returns a new string that is a substring of this string. The substring begins with the character at the specified index and extends to the end of this string.19 . substringpublic String substring(int beginIndex, int endIndex)Returns a new string that is a substring of this string. The substring begins at the specified beginIndex and extends to the character at index endIndex - 1. Thus the length of the substring is endIndex-beginIndex.20 . trimpublic String trim()Returns a copy of the string, with leading and trailing whitespace omitted.
  + StringBuffer类
    - 当对字符串进行修改的时候，需要使用 StringBuffer 和 StringBuilder 类。和 String 类不同的是，StringBuffer 和 StringBuilder 类的对象能够被多次的修改，并且不产生新的未使用对象。
    - StringBuilder 类在 Java 5 中被提出，它和 StringBuffer 之间的最大不同在于 StringBuilder 的方法不是线程安全的（不能同步访问）。由于 StringBuilder 相较于 StringBuffer 有速度优势，所以多数情况下建议使用 StringBuilder 类。
    - 针对添加和删除字符串的操作，StringBuffer类提供了一系列的方法。
    - 1 . appendpublic StringBuffer append(char c)Appends the string representation of the char argument to this sequence.
    - 2 . insertpublic StringBuffer insert(int index, char[] str, int offset, int len)Inserts the string representation of a subarray of the str array argument into this sequence. The subarray begins at the specified offset and extends len chars. The characters of the subarray are inserted into this sequence at the position indicated by index. The length of this sequence increases by len chars.
    - 3 . deleteCharAtpublic StringBuffer deleteCharAt(int index)Removes the char at the specified position in this sequence. This sequence is shortened by one char.
    - 4 . deletepublic StringBuffer delete(int start, int end)Removes the characters in a substring of this sequence. The substring begins at the specified start and extends to the character at index end - 1 or to the end of the sequence if no such character exists. If start is equal to end, no changes are made.
    - 5 . replacepublic StringBuffer replace(int start, int end, String str)Replaces the characters in a substring of this sequence with characters in the specified String.
    - 6 . setCharAtpublic void setCharAt(int index, char ch)The character at the specified index is set to ch. This sequence is altered to represent a new character sequence that is identical to the old character sequence, except that it contains the character ch at position index.
    - 7 . toStringpublic String toString()
    - Returns:a string representation of this sequence of characters.
    - 8 . reversepublic StringBuffer reverse()
    - Returns:a string representation of this sequence of characters.
* System类和Runtime类
  + System类
    - System类定义了一些与系统相关的属性和方法，它所提供的属性和方法都是静态的，因此想要调用这些属性和方法，直接使用System类调用即可。
    - System类的常用方法表：
    - static void exit(int status) : 该方法用于中止当前正在运行的Java虚拟机，其中，参数status表示状态码，若状态码非0，则表示异常终止。
    - static void gc() : 运行垃圾回收器，并对垃圾进行回收。
    - static native long currentTimeMillis() : 返回以毫秒为单位的当前时间。（时间戳）
    - static void arraycopy(Object src, int srcPos（表示原数组中拷贝元素的起始位置）, Object dest, int destPos（表示拷贝到目标数组的起始位置）, int length（表示拷贝元素的个数）) : 从src引用的指定源数组复制到dest引用的数组，复制从指定的位置开始，到目标数组的指定位置结束。
    - static Properties getProperties() : 取得当前的系统属性。
    - static String getProperty(String key) : 获取指定键描述的系统属性。
  + Runtime类
    - 每次使用Java命令启动虚拟机都对应一个Runtime实例，并且只有一个实例，因此该类采用单例模式进行设计，对象不可以直接实例化。
    - Runtime类中提供了一个exec（）方法，该方法用于执行一个dos命令，此方法返回一个Process对象，通过该对象可以对产生的新进程进行管理，例如，通过调用Process对象的destroy（）方法，就可以关闭此程序。（如果这里的dos命令不是关于打开某个系统软件的话，可能就不同了。）
* Math类和Random类
  + Math类
    - Math类是数学操作类，提供了一系列用于数学运算的静态方法，包括求绝对值、三角函数等。Math类中有两个静态常量PI和E，分别代表数学常量π和e。
    - Math.abs() : 求绝对值。
    - Math.ceil() : 求大于参数的最小整数。
    - Math.floor() : 求小于参数的最大整数。
    - Math.round() : 对小数进行四舍五入。（对传入整形参数的可行性存疑）
    - Math.max() : 求两个数的较大值。
    - Math.min() : 求两个数的
  + Random类
    - 在JDK的Java.util包中有一个Random类，它可以在指定的取值范围内随机产生数字。在Random类中提供了两个构造方法：
    - Random() : 构造方法，用于创建一个伪随机数生成器。
    - Random(long seed) : 构造方法， 使用一个long型的seed种子创建伪随机数生成器。
  + 包装类
    - 在Java中，很多类的方法都需要接收引用类型的对象，此时就无法将一个基本类型的值传入。为了解决这样的问题，JDK中提供了一系列的包装类，通过这些包装类可以将基本数据类型的值包装为引用数据类型的对象。在Java中，每种基本数据类型都有对应的包装类，如下所示(其中，除了Integer和Character类，其他包装类的名称和基本数据类型的名称一致，只是类名的首字母需要大写)：
      * byte Byte
      * char Character
      * int Integer
      * short Short
      * long Long
      * float Float
      * double Double
      * boolean Boolean
    - Integer类除了具有Object类的所有方法外，还有一些特有的方法：
      * static String toBinaryString(int i) : 以二进制无符号整形形式返回一个整数参数的字符串。
      * static String toHexString(int i) : 以十六进制无符号整数形式返回一个整数参数的字符串。
      * static String toOctalString(int i) : 以八进制无符号整数形式返回一个整数参数的字符串。
      * static Integer valueOf(int i) : 返回一个表示指定的int值的Integer实例。
      * static Integer valueOf(String s) : 返回保存指定的String的值的Integer对象。
      * static int parseInt(String s) : 将字符串参数作为有符号的十进制整数进行解析。
      * intValue() : 将Integer类型的值以int类型返回。
    - 其中intValue方法可以用来进行拆箱操作，parseInt（）方法在程序中很常用，它是一个静态方法，用于将一个字符串形式的数值转换成int类型。
    - 在使用包装类时，需要注意以下几点：
      * 包装类都重写了Object类中的toString（）方法，以字符串形式返回被包装的基本数据类型的值。
      * 除了character外，包装类都有valueOf（String s）方法，可以根据String类型的参数创建包装类对象，但参数字符串不能为null，而且字符串必须是可以解析为相应基本类型的数据，否则虽然编译通过，但运行时会报错。
      * 除了character外，包装类都有parse...（String s）的静态方法，将字符串转换为对应的基本数据类型。参数s不能为null，而且同样必须是可以解析为响应基本数据类型的数据，否则运行时会报错。
    - 在JDK5.0的版本中提供了自动拆箱和自动装箱技术，也就是可以自动进行基本数据类型和包装类之间的转换。
  + Lambda表达式和Stream