

常用 API 方法简单汇总:

Object 类

Object 类是类层次结构的根类, 每个类都会使用 Object 作为超类, 所有对象 (包括数组) 都实现这个类的方法。

普通方法:

protected void finalize():对象的垃圾回收器调用此方法。

public final Class<?> getClass():返回此 Object 的运行类, 即输出当前类是什么。

public int hashCode()返回该对象的哈希码值。

public String toString():返回该对象的字符串表示,建议所有子类都重写此方法。

public boolean equals(Object obj):用来比较两个对象的地址值是否相等, 需要重写此方法。

Scanner 类

Scanner 类是一个获取键盘输入数据的封装类, 通过使用 Scanner 类中的方法, 可以得到各种类型的数据

构造方法:

public Scanner(InputStream source):构造一个新的 Scanner, 它生成的值是从指定的输入流扫描的。取自该流的字节通过底层平台的默认字符集转换成字符。

public int nextInt():将输入信息的下一个标记扫描为一个 int。返回: 从输入信息扫描的 int

public String nextLine():此扫描器执行当前行, 并返回跳过的输入信息。 此方法返回当前行的其余部分, 不包括结尾处的行分隔符。当前位置移至下一行的行首。

String 类

就是将多个字符串联起来, 其实, 字符串就相当于是一个字符数组。

构造方法:

public String():空参数构造函数。

public String (byte[] bytes):把给定的 byte 数组转换成字符串, 打印出的是字符。

public String (byte[] bytes,int startIndex ,int length):把给定的字节数组的一部分转换成字符串。(包含 startIndex 位置的元素)。

public String (char[] value):把给定的字符数组转换成字符串。

public String (char[] value ,int startIndex ,int length):把给定的字符数组的一部分转换成字符串。

public String (String original):把字符串转换成字符串。

普通方法

String 类的判断功能

boolean equals(Object obj): 比较两个字符串内容是否相同。

boolean equalsIgnoreCase(String str): 比较两个字符串内容是否相同(忽略大小写)。

boolean contains(String str): 判断该字符串是否包含给定的字符串。

boolean startsWith(String str): 判断该字符串是否以 给定的字符串开头。

boolean endsWith(String str): 判断该字符串是否以给定的字符串结尾。

boolean isEmpty(): 判断该字符串是否为空, null 作为空字符串就会报空指针异常的错误, 所以空字符串指“”。

String 类的获取功能

`int length()`: 获取字符串的长度。

`char charAt(int index)`: 把该字符串中给定位置的字符返回。

`int indexOf(int ch)`: `byte short char int` 返回给定字符在字符串中第一次出现的位置。

`int indexOf(String str)`: 返回给定的字符串在该字符串中第一次出现的位置。

`int indexOf(int ch,int fromIndex)`: 在指定位置开始, 返回给定字符在字符串中第一次出现的位置。

`int indexOf(String str,int fromIndex)`在指定位置开始, 返回给定的字符串在该字符串中第一次出现的位置。

`String substring(int start)`: 在指定位置开始, 截取该字符串, 返回一个新字符串。

`String substring(int start,int end)`在指定位置开始, 到指定位置结束, 截取该字符串, 范返回一个新字符串(包左不包右)。

String 类的转换功能

`byte[] getBytes()`: 把该字符串 转换成 字节数组。

`char[] toCharArray()`: 把该字符串 转换成字符数组。

`static String copyValueOf(char[] chs)`: 把给定的字符数组 转换为 字符串。

`static String valueOf(char[] chs)`: 把给定的字符数组 转换为 字符串。

`static String valueOf(int i)`基本类型: 把给定的 `int` 数据 转换成字符串。

`String toLowerCase()`: 把该字符串转换成小写字母。

`String toUpperCase()`: 把该字符串转换成大写字母。

`String concat(String str)`: 字符串连接。

替换功能

`String replace(char old,char new)`: 在该字符串中, 用指定的新字符 替换指定老字符。

`String replace(String old,String new)`: 在该字符串中, 用指定的新字符串 替换 指定的老字符串。

切割功能

`String[] split(String regex)`: 使用给定的分隔符, 将该字符串分解成多个字符串。

`String trim()`: 去除字符串两空格。

按字典顺序比较两个字符串:

`int compareTo(String str)`

`int compareToIgnoreCase(String str)`

Arrays 类

`Arrays`: 数组工具类, 此类包含用来操作数组 (比如排序和搜索) 的各种方法, 里面的方法都是静态修饰的, 可以类名直接调用

普通方法:

`public static String toString(int[] a)`: 把数组转换成字符串

`public static void sort(int[] a)`: 把数组排序

`public static int binarySearch(int[] a,int key)`: 二分查找法

StringBuffer 类

字符串缓冲区, 缓冲区用于存储数据, 所以也称之为容器。字符串组成原理就是通过它实现的。

构造方法:

`public StringBuffer()`:容量为 16 个字符。

`public StringBuffer(int capacity)`: 容量为指定的大小。

`public StringBuffer(String str)`: 容量为 16 字符+字符串的长度。

需要掌握的方法:

`public int capacity()`: 理想使用情况。

`public int length()`: 实际使用情况。

添加功能

`public StringBuffer append(String str)`: 在该字符串后面 追加给定的字符串, 返回此对象的一个引用。

`public StringBuffer insert(int offset,String str)`:在给定位置, 插入给定的字符串到原字符串中。

删除功能

`public StringBuffer delete(int start,int end)`:从指定位置开始到指定位置结束的数据在该字符串中删除, 返回此对象。

`public StringBuffer deleteCharAt(int index)`:把指定位置的字符在字符串中删除, 返回此对象。

替换功能

`public StringBuffer replace(int start,int end, String str)`

从指定的开始位置, 到指定的结束位置, 把原来的数据 替换为 新的字符串数组。

截取子串功能

`public String substring(int start)`:从给定位置开始, 截取该字符串,返回截取后一个新字符串。

`public String substring(int start,int end)`:从给定位置开始, 到给定的位置结束, 截取该字符串, 返回截取后的一个新字符串。

反转功能

`public StringBuffer reverse()`:abc --> cba

StringBuilder 类与 StringBuffer 一致

Integer 类

构造方法

`public Integer (int value)`: 把指定的 int 数据转换成 Integer 对象。

`public Integer (String s)`: 把指定的字符串转换成 Integer 对象。使用要求: 字符中只能有数字字符来组成。

成员变量

最大值 $2^{31}-1$: MAX_VALUE:静态的, 可以直接调用。

最小值 $-(2^{31})$

普通方法:

进制转换

`public static String toBinaryString(int i)`: 将给定的 10 进制数 转换为 2 进制, 以字符串的形式显示。

`public static String toHexString(int i)`: 将给定的 10 进制数 转换为 16 进制, 以字符串的形式显示。

`public static String toOctalString(int i)`: 将给定的 10 进制数 转换为 8 进制, 以字符串的形式显示。

`public static int parseInt(String s)`: 将字符串 转换为 `int` 数据。

`public static int parseInt(String s,int radix)`:把给定进制的字符串 转换为 10 进制。