

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(char 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(char 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。

Math 类

MATH 类包含用于执行基本数学运算的方法, 如初等指数、对数、平方根和三角函数, MATH 类的所有方法都是静态的, 所以可以直接通过类名进行调用。

成员变量 (常量):

E: 自然对数的底数 PI: 圆周率

普通方法:

`public static double abs(double a)`:返回 `double` 的绝对值。

`public static double ceil(double a)`:返回接近这个数的最小整数。

`public static double exp(double a)`: 返回欧拉数的 `double` 次幂的值。

`public static double floor(double a)`:返回接近这个数的最大整数。

`public static double log(double a)`: 返回 `double` 值的以 `e` 为底的自然对数。

`public static double log10(double a)`:返回 `double` 值的以 10 为底的自然对数。

`public static double max(double a,double b)`:返回两个 `double` 值中较大的一个。

`public static double min(double a ,double b)`:返回两个 `double` 值中较小的一个。

`public static double pow(double a,double b)`:返回第一个参数的第二个参数次幂的值。

`public static double random()`:返回带正号的 `double` 值, 该值大于等于 0.0 且小于 1.0。

`public static double round(double a)`:获取参数的近似值。

`public static double sqrt(double a)`:返回参数的平方根。

Random 类: 用于产生随机数的类, 用于生成伪随机数的流。

构造方法:

`public Random()`:创建一个随机数 (使用的是默认种子, 其实就是当前时间的毫秒值)。

`public Random(long seed)`:使用单个 `long` 种子创建一个新的随机数生成器。

普通方法:

`public int nextInt()`:返回 `int` 数据类型范围内的伪随机数。

`public int nextInt(int n)`:返回包含[0,n)的 `int` 的数据 (包左不包右)。

Date 类 util

Date 类是日期类, 用于日期数据的操作。表示特定的时间, 精确到毫秒。java 重新了 Date 的 `toString()` 方法。

构造方法

`public Date()`:分配 `Date` 对象并初始化此对象，通过当前时间创建对象。

`public Date(long date)`:通过指定的时间类创建对象，按指定的时间为基准，将基准时间 1970 年 1 月 1 日 00:00:00 GMT 加上相对应的毫秒值 `date`。

普通方法

`public void setTime(long time)`:设置此对象的时间，按指定的时间为基准，加上 `long` 类型的毫秒值，得到对应的时间。

`public long getTime()`:获取时间的对应毫秒值。

日期到毫秒的转换

使用 `getTime` 方法：`public long getTime()`:获取时间的对应毫秒值。

毫秒到日期的转换

使用构造函数：`public Date(long date)`。

`DateFormat` 类：是对日期进行格式化的类，是抽象类，不是静态的，子类 `SimpleDateFormat` 实现，需要通过它来使用 `DateFormat`。

构造方法

`public SimpleDateFormat()`:用默认的模式和默认的语言环境的日期格式。

`public SimpleDateFormat(String pattern)`:用给定的模式和给定的语言环境来创建日期。

普通方法

`public final String format(Date date)`:将时间转化成字符串。

`public Date parse(String source)`:把指定的时间转换成字符串。

`Calendar` 类：`util` 是工具包，先看工具包，是一个抽象类，可以获取年、月、日、时、分、秒等字段信息。其子类是 `GregorianCalendar`

成员变量，字段(常量)计算机中的月份是从 0—11，全是大写表示常量。

`YEAR`, `MONTH`, `DAY_OF_MONTH`, `HOURL_OF_DAY`, `MINUTE`, `SECOND`

常量都由 `public static final` 来修饰。以上分别表示年、月、日、时、分、秒，都是静态的，可以直接类名调用。

普通方法

`public int get(int field)`:获取字段的值。

`public static Calendar getInstance()`:获取日历对象，通过本类方法获取本类对象。

`Calendar` 的方法：

`public final void set(int year, int month, int date)`:指定年月日。

`public abstract void add (int field, int amount)`: 为给定的日历字段添加或减去指定的时间量。

`System` 类，在 `lang` 包中，不可以被继承，提供一些有用的字段和方法,系统类，提供标准输入、输出和错误输出流。

普通方法

`public static void gc()`:运行垃圾回收器，jvm 的垃圾回收机制中，用来释放内存中的垃圾。

`public static void exit(int status)`:终止当前正在运行的 Java、虚拟机，参数用作状态码，根据惯例，非 0 的状态码表示异常终止。

`public static long currentTimeMillis()`:返回以毫秒为单位的当前时间，通过此时间可以计算程

序运行的时间。

`public static void arraycopy(Object src,int srcPos,Object dest,int destPos,int length):`从指定源数组中复制一个数组，复制从指定位置开始，到目标数组的指定位置结束。

参数说明：

`src` : 源数组 `srcPos` 源数组中的起始位置 `dest` 目标数组 `destPos` : 目标数据中的起始位置 `length` 要复制数组元素的数量，体现了多态。