

API基础

字符串

String

- java.lang.String使用了final修饰, 不能被继承
- 是一个封装char[]数组的对象
- String有一个常量池
 - char charAt(int index) 根据下标获取相应字符
 - String concat(String str) 在字符串的末尾处拼接自定义字符串
 - boolean contains(CharSequence cs) 判断是否包含指定的字符串
 - boolean endsWith(String suffix) 判断字符串是否以指定后缀 (字符串) 结尾
 - boolean equals(Object anObject) 判断字符串是否于指定字符串相等
 - byte[] getBytes() 把字符串的数据转成byte[]数组
 - int indexOf(String str) 获取指定字符串在字符串中第一次出现时的下标值
 - int lastIndexOf(String str) 获取指定字符串在字符串中最后一次出现时的下标值
 - boolean isEmpty() 判断字符串是否为空
 - int length() 获取字符串的长度
 - String replace(char oldChar, char newChar) 把旧的字符串中的字符替换
 - String[] split(String regex) 按照指定的规则切割字符串
 - boolean startsWith(String prefix) 判断是否以指定字符串开始
 - String substring(int beginIndex) 从制定下标处截取所有的字符串
 - String substring(int beginIndex, int endIndex) 从制定下标处开始, 到指定下标处结束, 截取中间段, 含头不含尾
 - char[] toCharArray() 把字符串转成一个新的字符数组
 - String toLowerCase() 自动转换成小写
 - String toUpperCase() 自动转换成大写
 - String trim() 去除前导空格和后置空格
 - static String valueOf(int n) 用来把各种类型的数据转成String类型
- StringBuilder/StringBuffer
 - 构造方法
 - StringBuffer() 构造一个不带任何字符的字符串生成器, 其初始容量为16个字符。
 - StringBuffer() 构造方法
 - StringBuffer() 构造方法
 - 常用方法
 - StringBuffer.append(String str) 将指定的字符串追加到此字符串序列。
 - StringBuffer.append(String str) 常用方法

Scanner类

Scanner类

- 一个简单的文本扫描器, 可以使用正则表达式解析原始类型和字符串。
- nextLine 从左往右扫描数据, 扫描结束标志: 回车符或换行符
- next 从左往右扫描数据, 在扫描到第一个有效字符之前会自动过滤掉换行和回车符。扫描结束标志: 空格, tab键, 回车符, 不会过滤回车符
- nextInt 从左往右扫描数据的int类型
- nextBoolean 判断输入中是否有其他单位, 返回值是一个boolean类型

Object类型

Object类型

- 所有对象的顶级父类
- protected void finalize() 当垃圾回收器确定不存在对该对象的更多引用时, 会对对象执行该方法而调用此方法。通常在子类中实现此方法, 最后一次执行的方法
- boolean equals(Object obj) 指示其他某个对象是否与此对象“相等”; 当前对象和参数对象比较大小, 默认是比较内存地址
- int hashCode() 返回该对象的哈希码值
- String toString() 返回该对象的字符串表示; 默认返回: 类名@地址 的格式
- 这些方法我们一般会对其行重写操作, 已达到业务需求

java.util.*

Math

- sqrt 开方根, double sqrt(double d)
- pow 次方, Math.pow(a, b) 即a的b次方
- random 随机数

IO

BigDecimal/BigInteger

BigDecimal/BigInteger

- BigDecimal: 常用来解决精确的浮点运算。
- BigInteger: 常用来解决超大的整数运算。
- 常用方法
 - add(BigDecimal bd) 做加法运算
 - subtract(BigDecimal bd) 做减法运算
 - multiply(BigDecimal bd) 做乘法运算
 - divide(BigDecimal bd) 做除法运算
 - setScale(保留位数, 舍入方式)
 - pow(int n) 求数据的几次方
- 除不尽时使用 divide(BigDecimal bd, 保留位数, 舍入方式)

Date

Date

- 存在java.util.Date包。
- 用毫秒封装一个毫秒值表示一个精确的时间点, 从1970-1-1 0点开始的时间。
- 封装的是系统当前时间的毫秒值
 - new Date() 构造方法
 - new Date(long time) 构造方法
- 取内部毫秒值
 - getTime() 常用方法
- 存取内部毫秒值
 - setTime(long time) 常用方法
- 获取当前月份
 - getMonth() 常用方法
- 获取当前小时
 - getHours() 常用方法
- 当前对象与参数对象比较, 当前对象大返回正数, 小返回负数, 相同0。
 - compareTo(Date date) 常用方法
- new SimpleDateFormat(格式) 构造方法
 - 格式: YYYY-MM-dd HH:mm:ss 年4位, 月2位, 日2位, 小时2位, 分钟2位, 秒2位
 - 格式: MM/dd/yyyy 月2位, 日2位, 年4位
- 把Date格式化成字符串
 - format(Date date) 常用方法
- 把String解析成Date
 - parse(String str) 常用方法

包装类

包装类

byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
char	Character
boolean	Boolean

数字包装类的抽象父类

- Number (java.lang)
- AtomicInteger (java.util.concurrent.atomic)
- AtomicLong (java.util.concurrent.atomic)
- BigInteger (java.math)
- BigDecimal (java.math)
- Byte (java.lang)
- Double (java.lang)
- Float (java.lang)
- Integer (java.lang)
- Long (java.lang)
- Short (java.lang)

Integer是int的包装类, 提供更多的方法

- 在Integer类中, 包含256个Integer缓存对象, 范围是-128到127。使用valueOf()时, 如果指定范围内的值, 返回缓存对象, 而不新建; 如果指定范围外的值, 则新建对象。
- Integer i = new Integer(123); 构造方法
- Integer i = Integer.valueOf(123); 构造方法
- 自动装箱 Integer i = 123;

Double是double的包装类, 提供更多的方法

- Double d = new Double(3.14); 构造方法
- Double d = Double.valueOf(3.14); 构造方法
- 自动装箱 Double d = 3.14;

和 new 没有区别