**一、字符串String**

String底层用unicode索引数组(既char数组)保存数据

**1构造**

字面量赋值：String str = "abc";

构造器方法：

String(byte[], <Charset>); 用byte构造，可指定字符集

String(char[]) 用char数组构造

String(String|StringBuffer) 用字符串构造

**2遍历**

char charAt(int index) 返回指定索引的字符

int indexOf(String str<, int fromIndex>)  返回指定字符串出现的索引

int lastIndexOf(String str<, int fromIndex>) 返回指定字符串最后出现索引

int hashCode() 返回串的哈希码

int length() 返回串的长度

**3判断比较**

判等比较

int compareTo(String anotherStr) 比较字符串

int compareToignoreCase(String str) 不区分大小写比较字符串

boolean equals(Object anObject) 比较字符串是否相等

boolean equalsIgnoreCase(String anotherString) 不区分大小写比较字符串

包含比较

boolean contains(CharSequence s) 是否包含指定字符串

字符串首尾比较

boolean endsWith(String suffix) 是否以指定字符串结尾

boolean startWith(String prefix) 是否以指定字符串开头

空串检查

boolean isEmpty() 判断字符串是否为空

正则匹配比较

boolean matches(String regex) 是否匹配对应正则

**4字符串操作**

连接

str1 + str2

String concat(String str) 返回拼接的字符串

替换

String replace(str, replacement\_str) 返回替换的串

String replaceAll(String regex, replacement\_str) 返回正则替换的串，正则引用为$1/$2...

分割

String[] split(String regex) 返回正则分割字符串数组

截取

String substring(int beginIndex<, int endIndex>) 返回截取的字符串

整理：大小写和去除空白

String toLowerCase() 将字符串转化为小写

String toUpperCase() 将字符串转化为大写

String trim() 去除字符串首尾空白

**5字符编码**

byte[] getBytes(<charset>) 返回指定编码的字节数组

char[] toCharArray() 返回字符数组

**6字符集(java.nio.charset.Charset)**

静态方法

SortedMap availableCharsets() 返回可用字符编码

Charset defaultCharset() 返回虚拟机默认字符编码

Charset forName(String) 返回指定名称的字符编码

实例方法

CharBuffer decode(ByteBuffer) 返回解码后的字符缓冲

ByteBuffer encode(CharBuffer|String) 返回编码后的字节缓冲

String name() 返回字符集的名称UTF-8

**二、可变字符串StringBuilder(非线程安全)**

与StringBuffer具有相同API。而StringBuffer支持多线程，但同步操作影响速度

**1初始化字符串**

指定字符串容量

StringBuilder() 初始容量为16个字符

StringBuilder(int capacity) 指定字符串初始容量

使用字符串初始化

StringBuilder(CharSequence seq) 用字符串序列初始化

**2字符串遍历**

遍历字符串中的字符

char charAt(int index) 返回指定索引的字符

查询子字符串的位置

int indexOf(String str) 指定字符串最先出现索引

int indexOf(String str, int fromIndex) 指定字符串最先出现索引

int lastIndexOf(String str) 指定字符串最后出现索引

int lastIndexOf(String str, int fromIndex)

字符串信息

int length() 返回字符串的长度

int capacity() 返回字符串的当前容量

**3字符串操作**

追加操作

StringBuilder append(boolean b) 追加布尔字符串描述

如："123".append(false) => "123false"

StringBuilder append(char c) 追加一个字符

StringBuilder append(CharSequence seq) 追加一个字符串序列

StringBuilder append(int|long|float|double) 追加数值型

插入操作

StringBuilder insert(int from, boolean b) 指定位置插入布尔描述

StringBuilder insert(int from, char c) 指定位置插入一个字符

StringBuilder insert(int from, CharSequence s) 指定位置插入字符串

StringBuilder insert(int from, int|long|float|double) 指定位置插入数值型

删除操作

StringBuilder delete(int start, int end) 删除指定的子字符串内容

StringBuilder deleteCharAt(int index) 删除指定索引的字符

字符串替换

StringBuilder replace(int start, int end, String str) 替换指定位置的字符串

颠倒字符串

StringBuilder reverse() 将字符串顺序颠倒

截取子字符串

String substring(int start) 从指定位置截取字符串

String substring(int start, int end) 从指定位置截取字符串

调整字符串容量

void ensureCapacity(ini mininumCapacity) 扩容字符串

void trimToSize() 适当的缩减字符串大小

**三、正则匹配**

**1匹配模式(java.util.regex.Pattern)**

static Pattern compile(String, <flags>) 创建一个匹配对象

flags：匹配模式int

·CASE\_INSENSITIVE：忽略大小写

·MULTILINE：匹配多行，即^$对每一行进行匹配

static boolean matches(String reg, String) 判断是否匹配

Matcher matcher(String) 返回匹配对象

**2匹配对象(java.util,regex.Matcher)**

基于find()迭代寻找匹配子串

int end() 返回匹配串的结尾索引

int end(int) 返回匹配子组串的结尾索引

boolean find() 匹配下一个串，false表遍历结束

String group() 返回匹配的串

String group(int) 返回匹配的子组串，int指定组号

int groupCount() 返回匹配的子组数

int start() 返回匹配串的起始索引

int end(int) 返回匹配子组串的起始索引