

第十二章 String类的概述和使用

12.1 String类的概念（重点）

- java.lang.String类用于描述字符串，Java程序中所有的字符串字面值都可以使用该类的对象加以描述，如："abc"。
- 该类由final关键字修饰，表示该类不能被继承。
- 从jdk1.9开始该类的底层不使用char[]来存储数据，而是改成 byte[]加上编码标记，从而节约了一些空间。
- 该类描述的字符串内容是个常量不可更改，因此可以被共享使用。
如：
String str1 = "abc"; - 其中"abc"这个字符串是个常量不可改变。

str1 = "123"; - 将"123"字符串的地址赋值给变量str1。
 - 改变str1的指向并没有改变指向的内容

12.2 常量池的概念（原理）

由于String类型描述的字符串内容是常量不可改变，因此Java虚拟机将首次出现的字符串放入常量池中，若后续代码中出现了相同字符串内容则直接使用池中已有的字符串对象而无需申请内存及创建对象，从而提高了性能。

12.3 常用的构造方法（练熟、记住）

| 方法声明 | 功能介绍 |
|--|--|
| String() | 使用无参方式构造对象得到空字符序列 |
| String(byte[] bytes, int offset, int length) | 使用bytes数组中下标从offset位置开始的length个字节来构造对象 |
| String(byte[] bytes) | 使用bytes数组中的所有内容构造对象 |
| String(char[] value, int offset, int count) | 使用value数组中下标从offset位置开始的count个字符来构造对象 |
| String(char[] value) | 使用value数组中的所有内容构造对象 |
| String(String original) | 根据参数指定的字符串内容来构造对象，新创建对象为参数对象的副本 |

12.4 常用的成员方法（练熟、记住）

| 方法声明 | 功能介绍 |
|----------------------|------------------------|
| String toString() | 返回字符串本身 |
| byte[] getBytes() | 将当前字符串内容转换为byte数组并返回 |
| char[] toCharArray() | 用于将当前字符串内容转换为char数组并返回 |

| 方法声明 | 功能介绍 |
|------------------------|-------------------------|
| char charAt(int index) | 方法charAt用于返回字符串指定位置的字符。 |
| int length() | 返回字符串字符序列的长度 |
| boolean isEmpty() | 判断字符串是否为空 |

- 案例题目

判断字符串“上海自来水来自海上”是否为回文并打印，所谓回文是指一个字符序列无论从左向右读还是从右向左读都是相同的句子。

| 方法声明 | 功能介绍 |
|-------------------------------------|-------------------------|
| int compareTo(String anotherString) | 用于比较调用对象和参数对象的大小关系 |
| int compareToIgnoreCase(String str) | 不考虑大小写，也就是'a'和'A'是相等的关系 |

- 案例题目

编程实现字符串之间大小的比较并打印。

| 方法声明 | 功能介绍 |
|--|----------------------|
| String concat(String str) | 用于实现字符串的拼接 |
| boolean contains(CharSequence s) | 用于判断当前字符串是否包含参数指定的内容 |
| String toLowerCase() | 返回字符串的小写形式 |
| String toUpperCase() | 返回字符串的大写形式 |
| String trim() | 返回去掉前导和后继空白的字符串 |
| boolean startsWith(String prefix) | 判断字符串是否以参数字符串开头 |
| boolean startsWith(String prefix, int toffset) | 从指定位置开始是否以参数字符串开头 |
| boolean endsWith(String suffix) | 判断字符串是否以参数字符串结尾 |

- 案例题目

编程实现上述方法的使用。

| 方法声明 | 功能介绍 |
|--|--------------------------------------|
| boolean equals(Object anObject) | 用于比较字符串内容是否相等并返回 |
| int hashCode() | 获取调用对象的哈希码值 |
| boolean equalsIgnoreCase(String anotherString) | 用于比较字符串内容是否相等并返回，不考虑大小写，如：'A'和'a'是相等 |

- 案例题目

提示用户从键盘输入用户名和密码信息，若输入“admin”和“123456”则提示“登录成功，欢迎使用”，否则提示“用户名或密码错误，您还有n次机会”，若用户输入三次后依然错误则提示“账户已冻结，请联系客服人员！”

| 方法声明 | 功能介绍 |
|--|-----------------------------------|
| int indexOf(int ch) | 用于返回当前字符串中参数ch指定的字符第一次出现的下标 |
| int indexOf(int ch, int fromIndex) | 用于从fromIndex位置开始查找ch指定的字符 |
| int indexOf(String str) | 在字符串中检索str返回其第一次出现的位置，若找不到返回-1 |
| int indexOf(String str, int fromIndex) | 表示从字符串的fromIndex位置开始检索str第一次出现的位置 |
| int lastIndexOf(int ch) | 用于返回参数ch指定的字符最后一次出现的下标 |
| int lastIndexOf(int ch, int fromIndex) | 用于从fromIndex位置开始查找ch指定字符出现的下标 |
| int lastIndexOf(String str) | 返回str指定字符串最后一次出现的下标 |
| int lastIndexOf(String str, int fromIndex) | 用于从fromIndex位置开始反向搜索的第一次出现的下标。 |

- 案例题目

编写通用的代码可以查询字符串"Good Good Study, Day Day Up!"中所有"Day"出现的索引位置并打印出来。

| 方法声明 | 功能介绍 |
|--|--|
| String substring(int beginIndex, int endIndex) | 返回字符串中从下标beginIndex（包括）开始到endIndex（不包括）结束的子字符串 |
| String substring(int beginIndex) | 返回字符串中从下标beginIndex（包括）开始到字符串结尾的子字符串 |

- 案例题目

提示用户从键盘输入一个字符串和一个字符，输出该字符(不含)后面的所有子字符串。

12.5 正则表达式的概念（了解）

正则表达式本质就是一个“规则字符串”，可以用于对字符串数据的格式进行验证，以及匹配、查找、替换等操作。该字符串通常使用^运算符作为开头标志，使用\$运算符作为结尾标志，当然也可以省略。

12.6 正则表达式的规则（了解）

| 正则表达式 | 说明 |
|-------------|---------------------------|
| [abc] | 可以出现a、b、c中任意一个字符 |
| [^abc] | 可以出现任何字符，除了a、b、c的任意字符 |
| [a-z] | 可以出现a、b、c、.....、z中的任意一个字符 |
| [a-zA-Z0-9] | 可以出现a~z、A~Z、0~9中任意一个字符 |

| 正则表达式 | 说明 |
|-------|--------------------------|
| . | 任意一个字符（通常不包含换行符） |
| \d | 任意一个数字字符，相当于[0-9] |
| \D | 任意一个非数字字符 |
| \s | 空白字符，相当于[\t\n\x0B\f\r] |
| \S | 非空白字符 |
| \w | 任意一个单词字符，相当于[a-zA-Z_0-9] |
| \W | 任意一个非单词字符 |

| 正则表达式 | 说明 |
|---------|--------------------------------------|
| X? | 表示X可以出现一次或一次也没有，也就是0 ~ 1次 |
| X* | 表示X可以出现零次或多次，也就是0 ~ n次 |
| X+ | 表示X可以出现一次或多次，也就是1 ~ n次 |
| X{n} | 表示X可以出现恰好 n 次 |
| X{n, } | 表示X可以出现至少 n 次，也就是>=n次 |
| X{n, m} | 表示X可以出现至少 n 次，但是不超过 m 次，也就是>=n并且<=m次 |

12.7 正则表达式相关的方法（熟悉）

| 方法名称 | 方法说明 |
|-------------------------------|------------------------------|
| boolean matches(String regex) | 判断当前正在调用的字符串是否匹配参数指定的正则表达式规则 |

- 案例题目

使用正则表达式描述一下银行卡密码的规则：要求是由6位数字组成。

使用正则表达式描述一下QQ号码的规则：要求是由非0开头的5~15位数组成。

使用正则表达式描述一下手机号码的规则：要求是由1开头，第二位数是3、4、5、7、8中的一位，总共11位

描述身份证号码的规则：总共18位，6位数字代表地区，4位数字代表年，2位数字代表月，2位数字代表日期，3位数字代表个人，最后一位可能数字也可能是X。

| 方法名称 | 方法说明 |
|---|--|
| String[] split(String regex) | 参数regex为正则表达式，以regex所表示的字符串为分隔符，将字符串拆分成字符串数组 |
| String replace(char oldChar, char newChar) | 使用参数newChar替换此字符串中出现的所有参数oldChar |
| String replaceFirst(String regex, String replacement) | 替换此字符串匹配给定的正则表达式的第一个子字符串 |
| String replaceAll(String regex, String replacement) | 将字符串中匹配正则表达式regex的字符串替换成replacement |