

正则表达式

正则表达式

知识点

- 正则表达式语法
 - 预定义字符类
 - Greedy 数量词
 - 范围词
 - 边界匹配器

应用

实例

- 判断字符串是否为邮箱号
- 偶数的模式

知识点

正则表达式：正则表达式其实就是用于操作字符串的一个规则，正则表达式的规则使用了特殊的符号表示规则

反斜杠是一个特殊的字符，在字符串中开始转义序列。因此Java中需要使用 `\\` 来表示 `\`。

正则表达式语法

预定义字符类

.	任何字符(与行结束符可能匹配可能不匹配)
\d	数字[0-9]
\D	非数字[^0-9]
\s	空白字符[\t\n\f\r]
\w	单词字符[a-zA-Z_0-9]
\W	非单词字符[^\w]

tips：任何预定义字符没有加上数量词之前都只能匹配一个字符而已

Greedy 数量词

X?	X，一次或一次没有
X*	X，零次或多次
X+	X，一次或多次
X{n}	X，恰好n次
X{n,}	X，至少n次
X{n,m}	X，至少n次，但是不超过m次

范围词

[abc]	a、b或c(简单类)
[^abc]	任何字符，除了a、b或c(否定)
[a-zA-Z]	a到z 或 A-Z，两头字母包括在内(范围)
[a-d[m-p]]	a到d 或 m到p：[a-dm-p](并集)
[a-z&&[def]]	d、e或f(交集)

tips：范围词里面不管内容有多长，没有数量词的配合都只能匹配一个字符而已

边界匹配器

^	行的开头
\$	行的结尾
\b	单词的边界
\B	非单词边界
\A	输入的开始
\G	上一个匹配的结尾
\Z	输入的结尾，仅用于最后的结束符(如果有的话)
\z	输入的结尾

应用

应用	
匹配	matches()
切割	split()
替换	replace()

实例

判断字符串是否为邮箱号

```
"schhznu@dingtalk.com".matches("\\w{3,20}@\\w+\\. (com|org|cn|net|gov)")
```

偶数的模式

```
"[\\d]*[02468]"
```