

Java ???(Regular Expression)

1. ??

正则表达式 (Regular Expression) 是一种用于匹配字符串中模式的工具。在 Java 中，正则表达式通过 `java.util.regex` 包中的类来实现。

1.1. 正则表达式的基本语法

正则表达式的基本语法包括元字符、转义字符、量词、分组和断言等。

- 元字符：
 - `[]`：匹配括号内的任意一个字符。
 - `[a-z]`：匹配小写字母。
- 量词：
 - `{n,m}`：匹配 `n` 到 `m` 次。
 - `+`：匹配 1 次或多次。
 - `*`：匹配 0 次或多次。
 - `?`：匹配 0 次或 1 次。
- 特殊字符：
 - `^`：匹配字符串的开始。
 - `$`：匹配字符串的结束。
 - `.`：匹配任意单个字符。
 - `|`：匹配左右两侧的任意一个表达式。

1.2. 正则表达式的应用

- `\d`：匹配数字 (0-9)。
- `\w`：匹配字母、数字、下划线 (a-zA-Z0-9_)。
- `\s`：匹配空白字符 (空格、制表符、换行符)。
- `\b`：匹配单词边界。

2. 正则表达式的应用

2.1. Pattern 类

`Pattern` 类用于编译正则表达式，生成可用于匹配的 `Matcher` 对象。

2.1.1. Pattern 类的常用方法

```
Pattern pattern = Pattern.compile("a*b");
```

编译正则表达式 "a*b" 时，'a' 表示 0 个或多个 'a'，'b' 表示 1 个或多个 'b'。

2.1.2. Matcher ??? ??

Matcher 用于匹配 Pattern 编译的正则表达式。Matcher 的 matches() 方法返回 boolean 值。

```
Pattern pattern = Pattern.compile("a*b");
Matcher matcher = pattern.matcher("aaab");
boolean matches = matcher.matches(); // true
```

2.2. Pattern? Matcher? ?? ???

- matches(): 返回 boolean 值，表示是否匹配。
- find(): 返回 boolean 值，表示是否找到下一个匹配项。
- replaceAll(): 返回 String 值，表示替换后的字符串。

3. ?????? ?? ??

3.1. ??? ?? ??

正则表达式用于验证电子邮件地址的格式。

```
Pattern pattern = Pattern.compile("^([a-zA-Z0-9_+&*~]+(?:\\.[a-zA-Z0-9_+&*~]+)*)@(?:[a-zA-Z0-9-]+\\.)+[a-zA-Z]{2,7}$");
Matcher matcher = pattern.matcher("example@domain.com");
boolean isValid = matcher.matches(); // true
```

3.2. ????? ?? ??

正则表达式用于验证电话号码的格式。

```
Pattern pattern = Pattern.compile("^\\d{3}-\\d{3,4}-\\d{4}$");
Matcher matcher = pattern.matcher("010-1234-5678");
boolean isValid = matcher.matches(); // true
```

Revision #1

Created 22 May 2025 05:15:02 by Dain

Updated 22 May 2025 05:16:16 by Dain