

[Java] Regex como Java

Transcrição

No mundo Java, a classe principal para trabalhar com expressões regulares se chama `Pattern` e faz parte do pacote `java.util.regex`.

Para criar uma regex, devemos usar o método estático `compile(regexString)`, da classe `Pattern`, por exemplo:

```
Pattern pattern = Pattern.compile("(\\d\\d)(\\w)");
```

Repare que foi necessário escapar o símbolo `\`. Outra coisa para notar é que o Java não possui uma sintaxe literal para a definição de regex.

Uma vez declarado o *pattern*, devemos criar um objeto `Matcher`, baseado na string de entrada:

```
Matcher matcher = pattern.matcher("11a22b33c");
```

Esse `matcher` possui, por sua vez, os métodos para acessar o *match*, o *grupo*, *index*, etc, mas antes de tudo devemos chamar o método `find`, que verifica se realmente há um *match* devolvendo `true` ou `false`:

```
boolean encontrou = matcher.find();
```

A partir daí podemos pegar os valores do *match*:

```
if(encontrou) {  
    String match = matcher.group();  
    String group1 = matcher.group(1);  
    String group2 = matcher.group(2);  
  
    int start = matcher.start();  
    int end = matcher.end();  
  
    System.out.printf("%s | %s | %s [%d,%d] %n", match, group1, group2, start, end);  
}
```

Sabendo disso, podemos colocar o nosso código dentro de um laço `while` para mostrar todos os resultados:

```
public class TesteRegex {  
  
    private static Pattern pattern = Pattern.compile("(\\d\\d)(\\w)");  
  
    public static void main(String[] args) {  
  
        Matcher matcher = pattern.matcher("11a22b33c");  
        while(matcher.find()) {
```

```
String match = matcher.group();
String group1 = matcher.group(1);
String group2 = matcher.group(2);

int start = matcher.start();
int end = matcher.end();

System.out.printf("%s | %s | %s [%d,%d] %n", match, group1, group2, start, end);
    }
}
```

As linguagens funcionam bem parecidas trabalhando com expressões regulares e o Java não é diferente.