

BACKEND - РАЗРАБОТКА НА ЯЗЫКЕ JAVA

РЭУ ИМ. Г.В. ПЛЕХАНОВА МПТ



<https://ucmpt.ru/>

<https://do.rea.ru/>

РЕГУЛЯРНОЕ ВЫРАЖЕНИЕ



Регулярное выражение – это шаблон, который обработчик регулярных выражений пытается сопоставить с введенным текстом. Шаблон состоит из односимвольных или многосимвольных литералов, операторов или конструкций

`String - matches()` принимает регулярное выражение и возвращает `true`, если строка соответствует этому выражению. Иначе возвращает `false`.

РЕГУЛЯРНОЕ ВЫРАЖЕНИЕ



```
public static void main(String[] args) {
    System.out.println("Введите одну цифру: ");
    Scanner input = new Scanner(System.in);
    String number = input.nextLine();
    if(number.matches(regex: "\\d+"))
    {
        System.out.println("Вы ввели одну или более цифр");
    }
    else {
        System.out.println("Вы ввели количество цифр < 1");
    }
}
```

Regular Expression Basics

.	Any character except newline
a	The character a
ab	The string ab
a b	a or b
a*	0 or more a's
\	Escapes a special character

Regular Expression Quantifiers

*	0 or more
+	1 or more
?	0 or 1
{2}	Exactly 2
{2, 5}	Between 2 and 5
{2,}	2 or more
(,5)	Up to 5

Default is greedy. Append ? for reluctant.

Regular Expression Groups

(...)	Capturing group
(?P<Y>...)	Capturing group named Y
(?:...)	Non-capturing group
\Y	Match the Y'th captured group
(?P=Y)	Match the named group Y
(?#...)	Comment

Regular Expression Character Classes

[ab-d]	One character of: a, b, c, d
[^ab-d]	One character except: a, b, c, d
[\b]	Backspace character
\d	One digit
\D	One non-digit
\s	One whitespace
\S	One non-whitespace
\w	One word character
\W	One non-word character

Regular Expression Assertions

^	Start of string
\A	Start of string, ignores m flag
\$	End of string
\Z	End of string, ignores m flag
\b	Word boundary
\B	Non-word boundary
(?=...)	Positive lookahead
(?!...)	Negative lookahead
(?<=...)	Positive lookbehind
(?<!...)	Negative lookbehind
(?())	Conditional

Regular Expression Flags

i	Ignore case
m	^ and \$ match start and end of line
s	. matches newline as well
x	Allow spaces and comments
L	Locale character classes
u	Unicode character classes
(?iLmsux)	Set flags within regex

Regular Expression Special Characters

\n	Newline
\r	Carriage return
\t	Tab
\YYY	Octal character YYY
\xYY	Hexadecimal character YY

Regular Expression Replacement

\g<0>	Insert entire match
\g<Y>	Insert match Y (name or number)
\Y	Insert group numbered Y

PATTERN MATHER



- Pattern представляет само регулярное выражение
- Matcher позволяет проводить различные операции над данными

```
public class REGULAR_EXPRESSIONS {  
    public static void main(String[] args) {  
        String text = "Иван: 89167048931\n Николай: 89178904725\n Дмитрий: 89153215763\n Сергей: 89164546831";  
  
        Pattern telephone_number = Pattern.compile( regex: "^((\\+7|7|8)+([0-9]){10})$");  
  
        Matcher matcher = telephone_number.matcher(text);  
    }  
}
```

ПАТТЕРНЫ ПРОЕКТИРОВАНИЯ



Шаблон проектирования или паттерн (англ. design pattern) в разработке программного обеспечения — повторяемая архитектурная конструкция, представляющая собой решение проблемы проектирования в рамках некоторого часто возникающего контекста. Обычно шаблон не является законченным образцом, который может быть прямо преобразован в код; это лишь пример решения задачи, который можно использовать в различных ситуациях.

Виды:

- Порождающие
- Структурные
- Поведенческие