

Regex

Intro

Problem

- Что если нам необходимо искать весьма специфические подстроки в тексте?
- Что если нам необходимо проверить соответствие текста, какому-то шаблону?

Solution

- **Regular Expression**

Regular Expression in **String**

split(String): String[]

```
String text = "FIFA will never regret it";  
String[] words = text.split("\\s*(\\s|,|!|\\.)*\\s*");  
for (String word : words) {  
    System.out.println(word);  
}
```

matches(String): boolean`

```
String input = "+12343454556";  
boolean result = input.matches("(\\+*)\\d{11}");  
if (result == true) {  
    System.out.println("It is a phone number");  
} else {  
    System.out.println("It is not a phone number!");  
}
```

replaceAll(String, String): String

```
String input = "Hello Java! Hello JavaScript! JavaSE 8.";
String myStr =input.replaceAll("Java(\\w*)", "HTML");
System.out.println(myStr); // Hello HTML! Hello HTML! HTML 8.
```


Regular Expression with **Pattern** and **Matcher**

Regular Expression with **Pattern** and **Matcher**

- Более мощные средства, для работы с регулярными выражениями предлагают классы **Pattern** и **Matcher** из пакета `java.util.regex`.
- Класс **Pattern** служит для хранения регулярного выражения
- Класс **Matcher** служит для выполнения операций поиска и сравнения.

Pattern: matches(String, CharSequence): boolean

```
import java.util.regex.Pattern;

public class StringsApp {
    public static void main(String[] args) {
        String input = "Hello";
        boolean found = Pattern.matches("Hello", input);
        if (found) {
            System.out.println("Найдено");
        } else {
            System.out.println("Не найдено");
        }
    }
}
```

Pattern: split(CharSequence): String[]

```
import java.util.regex.Pattern;

public class StringsApp {
    public static void main(String[] args) {
        String input = "Hello Java! Hello JavaScript! JavaSE 8.";
        Pattern pattern = Pattern.compile("[ ,.!?]");
        String[] words = pattern.split(input);
        for (String word : words) {
            System.out.println(word);
        }
    }
}
```

Pattern: `split(CharSequence) : String[]`

```
Hello  
Java
```

```
Hello  
JavaScript
```

```
JavaSE  
8
```

Matcher: matches(): boolean

```
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class StringsApp {
    public static void main(String[] args) {
        String input = "Hello";
        Pattern pattern = Pattern.compile("Hello");
        Matcher matcher = pattern.matcher(input);
        boolean found = matcher.matches();
        if (found) {
            System.out.println("Найдено");
        } else {
            System.out.println("Не найдено");
        }
    }
}
```

Matcher: find(): boolean ~~if~~ group(): String

```
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class StringsApp {
    public static void main(String[] args) {
        String input = "Hello Java! Hello JavaScript! JavaSE 8.";
        Pattern pattern = Pattern.compile("Java(\\w*)");
        Matcher matcher = pattern.matcher(input);
        while (matcher.find()) {
            System.out.println(matcher.group());
        }
    }
}
```

Matcher: find(): boolean и group(): String

```
Java  
JavaScript  
JavaSE
```


Matcher: replaceAll(String): String

```
String input = "Hello Java! Hello JavaScript! JavaSE 8.";
Pattern pattern = Pattern.compile("Java(\\w*)");
Matcher matcher = pattern.matcher(input);
String newStr = matcher.replaceAll("HTML");
System.out.println(newStr); // Hello HTML! Hello HTML! HTML 8.
```