Theory: Regexes in programs

27 minutes 0 / 4 problems solved

Regular expressions have a very wide scope of application. They're used in text editors and in implementations of programming languages, for parsing and syntax highlighting, for extraction of useful information from files and web sites. In this lesson, we would like to show you an example of a simple yet powerful program using a regular expression for string processing.

§1. Program with a regular expression

Let's consider a program that checks whether the input string is a valid login. We assume that login can contain any Latin letter, a number, the underscore character `_`, and the dollar sign `$`. The length must be no less than 5 characters and no more than 12 characters.

There is one additional feature: we should ignore all whitespace characters in the beginning and the end of the input string.

```
import java.util.Scanner;

class CheckLoginProblem {

    public static void main(String[] args) {

        /* The scanner object to read data from the standard input */
        Scanner scanner = new Scanner(System.in);


        /* The common pattern for valid logins */

        String loginRegex = "\\s*[a-zA-Z0-9_$]{5,12}\\s*";



        /* The read string which may be a login */

        String mayBeLogin = scanner.nextLine();



        boolean isLogin = mayBeLogin.matches(loginRegex);



        System.out.println(isLogin);

    }

}
```

Remember, the regex `"\\s*"` is a very useful practical tool for finding whitespace characters. It's preferable to use this shorthand instead of directly typing `" "`.

Let's try different inputs and test our program.

The input-output pair 1:

```
  testuser7
true
```

The input-output pair 2:

```
 test
false
```

The input-output pair 3:

Current topic:

Regexes in programs

Topic depends on:

- ✕ Shorthands
- ✕ Quantifiers

Topic is required for:

- Patterns and Matcher
- Bean Validation

Table of contents:

```
      user!!!
false
```

The input-output pair 4:

```
$test_user
true
```

As you can see, by using a simple regular expression we can write a quite powerful program for a real-life problem. If you wish, you can rewrite this regex using other special characters.

§2. Conclusion

In this lesson, we've seen a small example of how regular expressions can help us in string processing. You can experiment further and implement other regular expressions to check whether the login meets more strict requirements.

Report a typo

200 users liked this theory. **4** didn't like it. **What about you?**

😍 🙂 😐 ☹️ 😡

Start practicing

Comments (9) Hints (0) Useful links (1) Show discussion