Java Fundamentals: The Regular Expressions Playbook

INTRODUCING REGULAR EXPRESSIONS

Victor Grazi
ORACLE JAVA CHAMPION, SPEAKER AND GEEK

@vgrazi

Text Processing in Java with RegEx

Victor Grazi
ORACLE JAVA CHAMPION, SPEAKER AND GEEK

@vgrazi

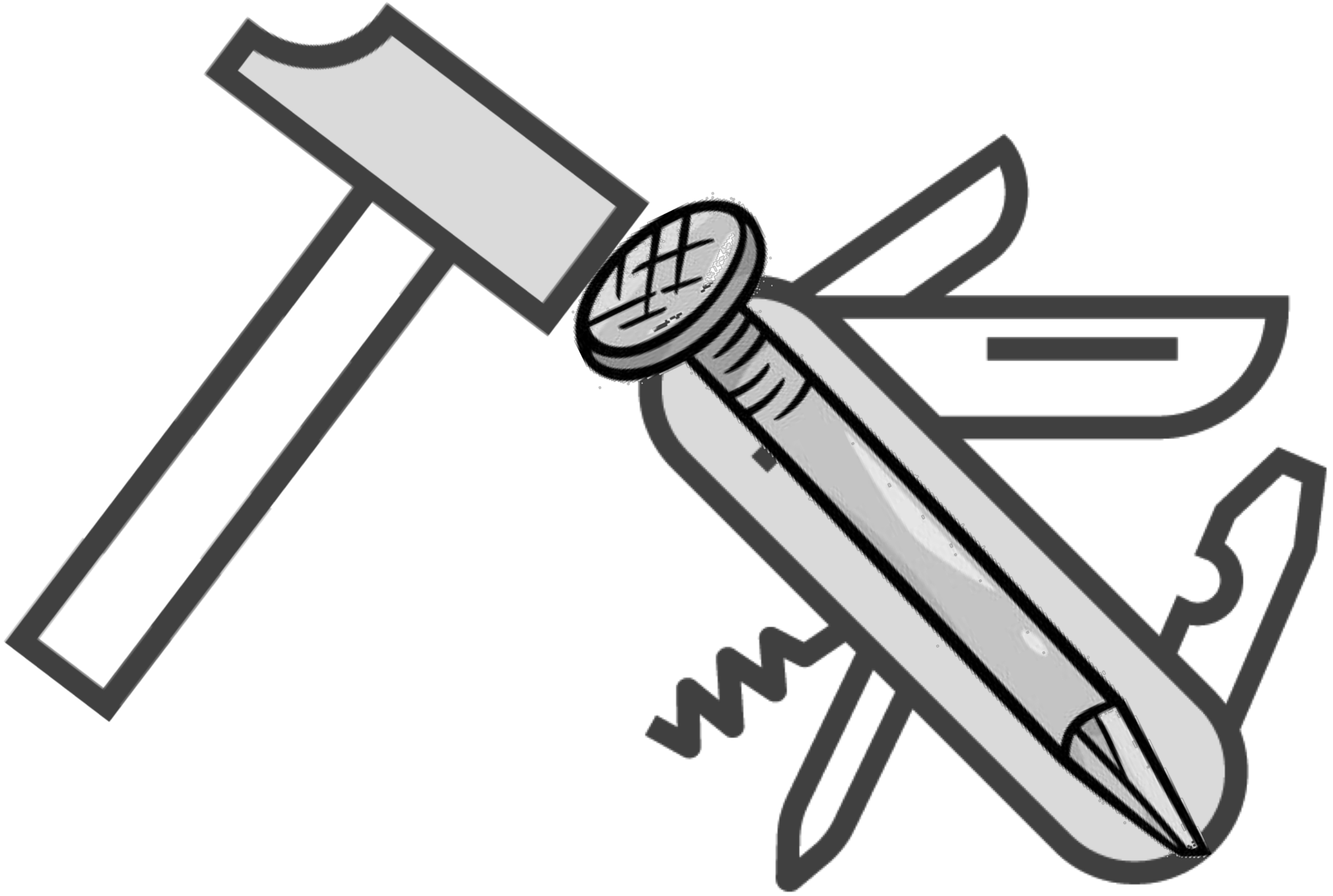
Introduction

Text processing is vital in many modern applications

- Validating form data
- Parsing data files or logs
- Extracting elements from ad-hoc text

In the olden days, we needed to write complicated custom parsers

Regular Expressions – “Regex” greatly reduces the parsing code

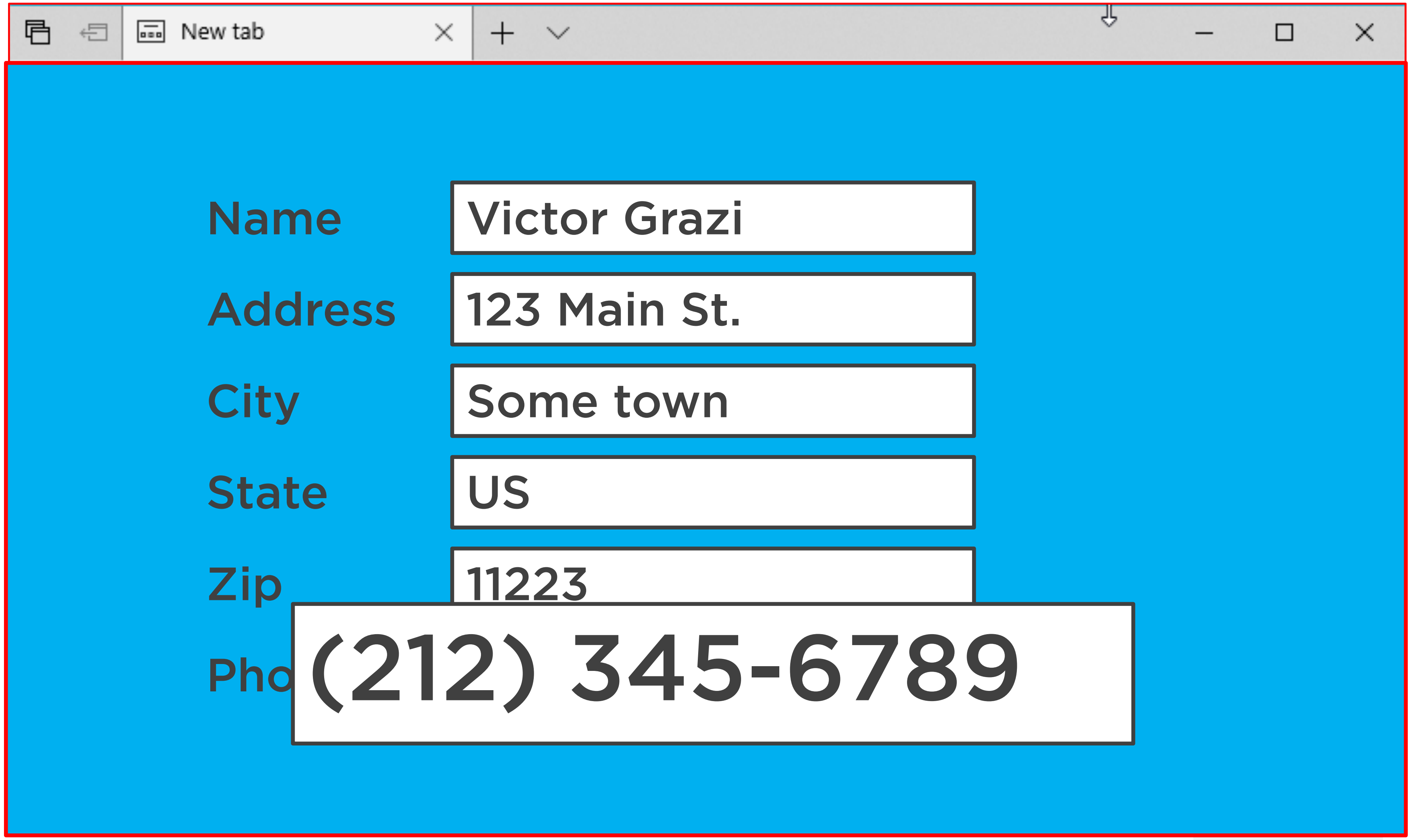


```java
// first, remove the white space, to validate the rest
phone = phone.trim();
boolean valid = phone.length() == 14;
for (int i = 0; i < 14; i++) {
    char c = phone.charAt(i);

    switch (i) {
        case 0:
            valid = c == '(';
```

```java
return phone.matches("\\(\\d{3}\\) ?\\d{3}-\\d{4}");
```

```java
            valid = c == ')';
            break;
        case 5:
            valid = c == ' ';
            break;
        case 9:
            valid = c == '-';
            break;
        default:
            valid = c >= '0' && c <= '9';
    }
    if (!valid) {
        break;
    }
}
return valid;
```



Name	Victor Grazi
Address	123 Main St.
City	Some town
State	US
Zip	11223
Pho	(212) 345-6789

Demo

Validate values from a web form, such as

- Address
- Postal code
- Email
- Phone

Validation algorithms for each field type

Using straight Java, requires many lines of code to parse each section

Text parsing

Validation

Matching

123 Main Street, Oaktown, Ca 12345

212-345-6789 ✓

11223-4567 ✓

"." is the all-purpose wildcard

* means zero or more occurrences of the preceding

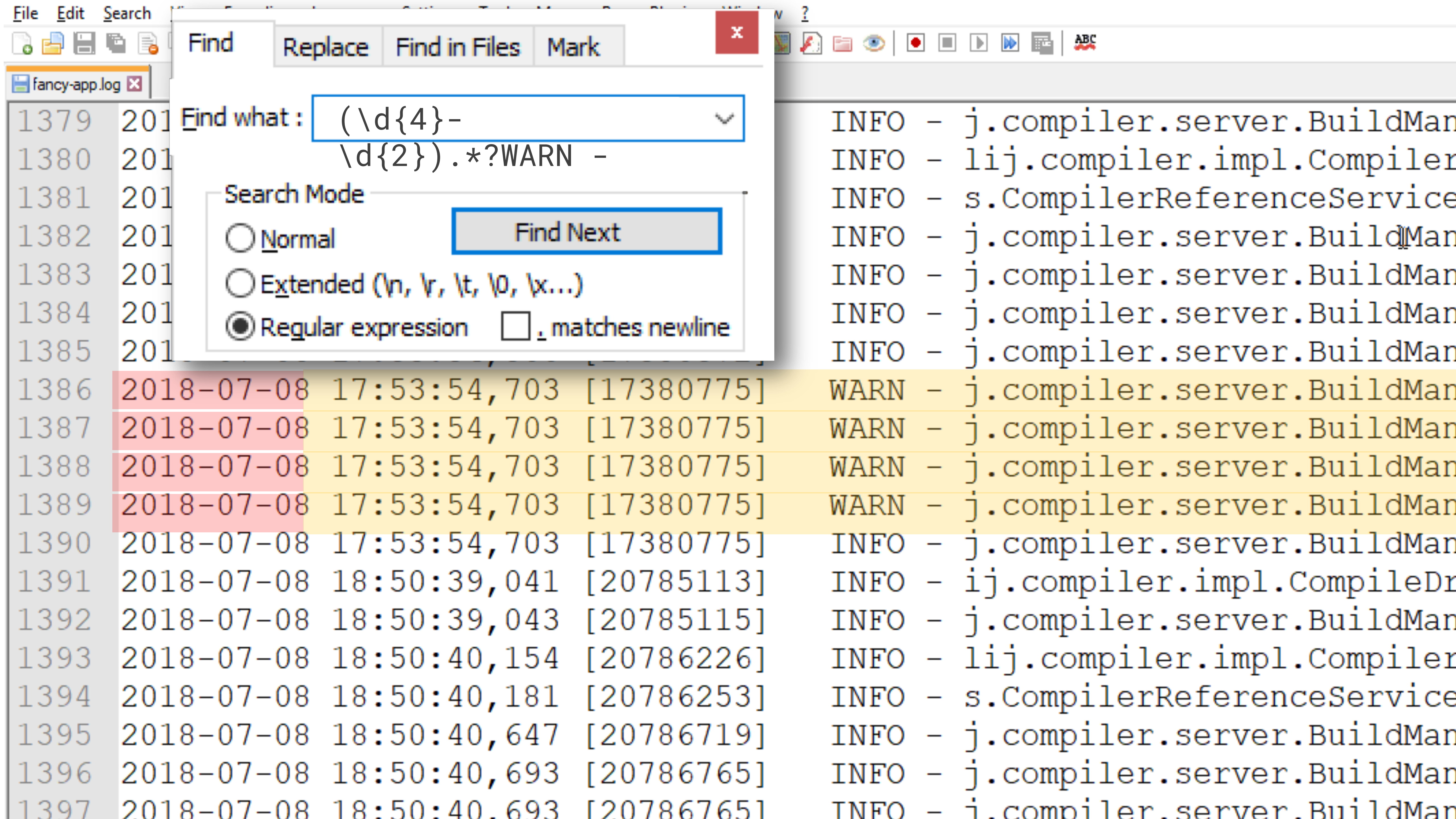
.* means match everything up to the end of line

```
100 bottles of beer on the wall
.............

aaaaaabcdefghij12345
............
a*

aaaaaabcdefghij12345
a*.*
```

```
dir my-file*.xls
```

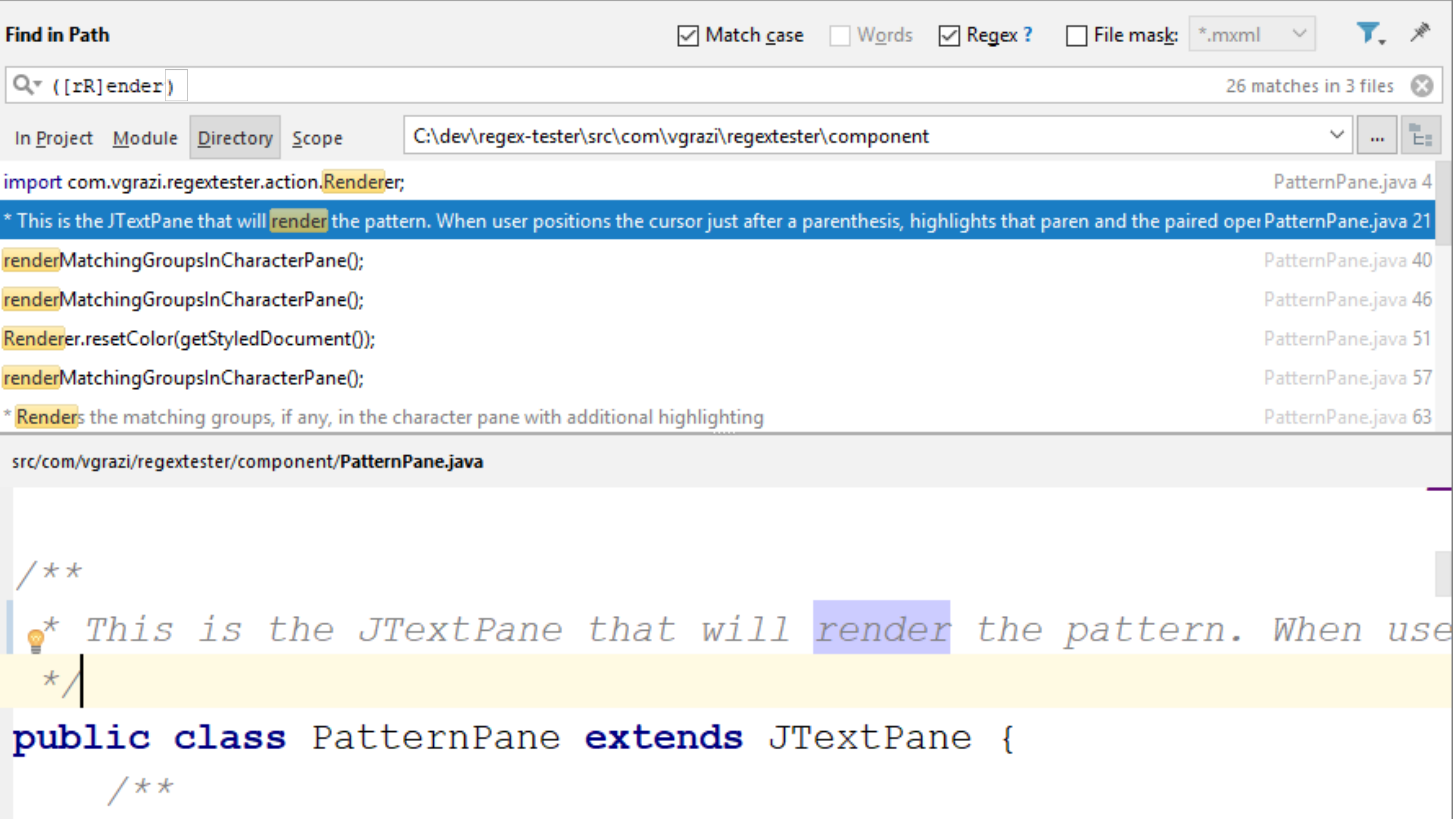


Find what : (\d{4}-\d{2}).*?WARN -

Search Mode
Normal [✓ Regular expression] Extended (\n, \r, \t, \0, \x...)
☐ . matches newline

```
201                                   INFO - j.compiler.server.BuildMan
201                                   INFO - lij.compiler.impl.Compiler
201                                   INFO - s.CompilerReferenceService
201                                   INFO - j.compiler.server.BuildMan
201                                   INFO - j.compiler.server.BuildMan
201                                   INFO - j.compiler.server.BuildMan
201                                   INFO - j.compiler.server.BuildMan
2018-07-08 17:53:54,703 [17380775]    WARN - j.compiler.server.BuildMan
2018-07-08 17:53:54,703 [17380775]    WARN - j.compiler.server.BuildMan
2018-07-08 17:53:54,703 [17380775]    WARN - j.compiler.server.BuildMan
2018-07-08 17:53:54,703 [17380775]    WARN - j.compiler.server.BuildMan
2018-07-08 17:53:54,703 [17380775]    INFO - j.compiler.server.BuildMan
2018-07-08 18:50:39,041 [20785113]    INFO - ij.compiler.impl.CompileDr
2018-07-08 18:50:39,043 [20785115]    INFO - j.compiler.server.BuildMan
2018-07-08 18:50:40,154 [20786226]    INFO - lij.compiler.impl.Compiler
2018-07-08 18:50:40,181 [20786253]    INFO - s.CompilerReferenceService
2018-07-08 18:50:40,647 [20786719]    INFO - j.compiler.server.BuildMan
2018-07-08 18:50:40,693 [20786765]    INFO - j.compiler.server.BuildMan
2018-07-08 18:50:40,693 [20786765]    INFO - j.compiler.server.BuildMan
```

```
public void findFile() {
    File dir = new File(".");
    FileFilter fileFilter = new RegexFileFilter
        ("^.*[mM]y-file(-\\d+)?\\.java$");
    File[] files = dir.listFiles(fileFilter);
    for (int i = 0; i < files.length; i++) {
        System.out.println(files[i]);
    }
}
```



Find in Path ☑ Match case ☐ Words ☑ Regex ? ☐ File mask: *.mxml

([rR]ender) 26 matches in 3 files

In Project Module [✓ Directory] Scope C:\dev\regex-tester\src\com\vgrazi\regextester\component

```
import com.vgrazi.regextester.action.Renderer;                                                                                                      PatternPane.java 4
* This is the JTextPane that will render the pattern. When user positions the cursor just after a parenthesis, highlights that paren and the paired oper PatternPane.java 21
renderMatchingGroupsInCharacterPane();                                                                                                              PatternPane.java 40
renderMatchingGroupsInCharacterPane();                                                                                                              PatternPane.java 46
Renderer.resetColor(getStyledDocument());                                                                                                           PatternPane.java 51
renderMatchingGroupsInCharacterPane();                                                                                                              PatternPane.java 57
* Renders the matching groups, if any, in the character pane with additional highlighting                                                          PatternPane.java 63
```

src/com/vgrazi/regextester/component/**PatternPane.java**

```
/**
 * This is the JTextPane that will render the pattern. When use
 */
public class PatternPane extends JTextPane {
    /**
```

Type	Example		RegEx
Phone:	212-345-6789 or (212)-345-6789	✓	\(?\d{3}\)? [-\s]?\d{3}-?\d{4}
Date:	2019-07-2 or 2019/7/2 But not 2019/7-2	✓ ✗	\d{4}([-/])?\d{1,2}\1\d{1,2}
Zip code:	11223 or 11223-4567	✓	\d{5}(-\d{4})?
Non-numeric	Victor_Grazi	✓	[\w&&[^\d]]+
Alphabetic	AaBbCc	✓	[A-Za-z]+

Summary

File searching

Validation

Text searching

Text extraction

Expression reformulating

Text cleansing