

Регулярни изрази

магия в действие

В миналия епизод видяхте...

Тревъплащението

```
string.replaceAll("java", "жаба")
```


Грозните

. | [] {} () ^ \$? * + \

АНГЛИЙСКИ

- gr[ae]y

- autori[sz]ation

Предефинирани класове

$\backslash w+ \backslash s \backslash w+ \backslash s \backslash w+ \backslash s \backslash w+ \backslash s \backslash w+ \backslash s \backslash w+$

$\backslash w+ \backslash d \backslash d. \backslash d \backslash d$

Повторения

- colour?r
- пада+m
- (\w+\s){6}

Групиране

- (? :зелена|златна|премазана) жаба
- (\w+) \1

Жабарски истории

```
p = Pattern.compile("\\w+");  
m = p.matcher(string);  
while (m.find()) {  
    System.out.println(m.group());  
}
```


КОТВИ

\wedge - начало на ред

$\$$ - край на ред

$\backslash b$ - начало или край на дума

$\backslash B$ - не е граница на дума

$\backslash A$ - начало на низ

$\backslash G$ - край на предишно търсене

$\backslash Z$ - край на низ, преди последния нов ред

$\backslash z$ - край на низ


```

public static void statistics(String text) {
    Map<String, String> patterns = new HashMap<String, String>();
    Map<String, Integer> counters = new HashMap<String, Integer>();
    patterns.put("punctuation marks", "[.,?!]");
    patterns.put("letters", "\\w");
    patterns.put("words", "\\w+");
    patterns.put("sentences",
        "[A-Z0-9].*?[.!?](?=\\s+|\\s*\\z)"); // [.!?]
    patterns.put("paragraphs",
        "[^\\r\\n\\s](?:\\s*\\r?\\n){2,}|\\s*\\z");// \\n{2,} +1

    for (String statistic : patterns.keySet()) {
        Pattern pattern = Pattern.compile(patterns.get(statistic));
        Matcher matcher = pattern.matcher(text);
        int counter = 0;
        while (matcher.find()) {
            counter++;
        }
        counters.put(statistic, counter);
    }

    for (String statistic : counters.keySet()) {
        System.out.println(statistic + ": " +
            counters.get(statistic));
    }
}

```


Огледай се и пресечи look-around

- (?=X) - позитивно гледане напред
- (?!X) - негативно гледане напред
- (?<=X) - позитивно гледане назад
- (?<!X) - негативно гледане назад

Модификатори

- Pattern.CANON_EQ -
- Pattern.CASE_INSENSITIVE (?i)
- Pattern.COMMENTS (?x)
- Pattern.MULTILINE (?m)
- Pattern.DOTALL (?s)
- Pattern.LITERAL \Q - \E
- Pattern.UNICODE_CASE (?u)
- Pattern.UNIX_LINES (?d)

Алчность

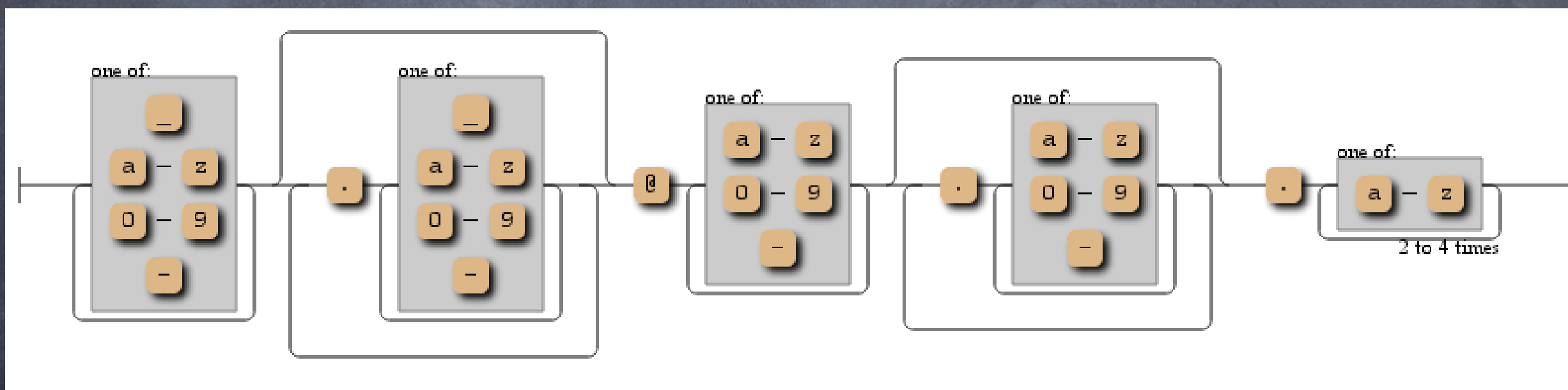
- алчны по природе
- скромны по принуде (?)
- властны (+)

Валидация на email

```
^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*(\.[a-z]{2,4})$
```

- не валидирайте адреси с регулярен израз

Визуално представяне



Валидация на email

```
(?:[a-z0-9!#$%&'*/+=?^_`{|}~-]+(?:\[a-z0-9!  
#$%&'*/+=?^_`{|}~-]+)*|"(?:  
[\x01-\x08\x0b\x0c\x0e-\x1f\x21\x23-\x5b\  
x5d-\x7f]|\\\  
[\x01-\x09\x0b\x0c\x0e-\x7f])*")@(?::(?:[a-  
z0-9](?:[a-z0-9-]*[a-z0-9])?\.|)+[a-z0-9](?:[a-  
z0-9-]*[a-z0-9])?|\[(?:(?:(25[0-5]|2[0-4][0-9]|  
[01]?[0-9][0-9]?)\.)?){3}(?:25[0-5]|2[0-4][0-9]|  
[01]?[0-9][0-9]?)|[a-z0-9-]*[a-z0-9]:(?:  
[\x01-\x08\x0b\x0c\x0e-\x1f\x21-\x5a\x53-  
\x7f]|\\\[ \x01-\x09\x0b\x0c\x0e-\x7f]))+)
```


Unicode

- `\p{InCyrillic}` - символ на кирилица
- `\p{L}` - `\p{Letter}` - буква
- `\p{Lu}` - главна буква
- `\p{Sc}` - валута
- `\p{N}` - цифра
- `\P{InCyrillic}` - всеки символ различен от кирилица

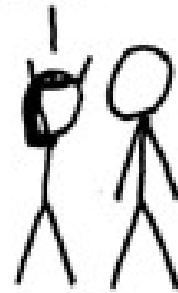
Някой каза ли: Още примери?

Some people, when confronted with a problem, think "I know, I'll use regular expressions." Now they have two problems.

Jamie Zawinski

WHENEVER I LEARN A NEW SKILL I CONCOCT ELABORATE FANTASY SCENARIOS WHERE IT LETS ME SAVE THE DAY.

OH NO! THE KILLER MUST HAVE FOLLOWED HER ON VACATION!

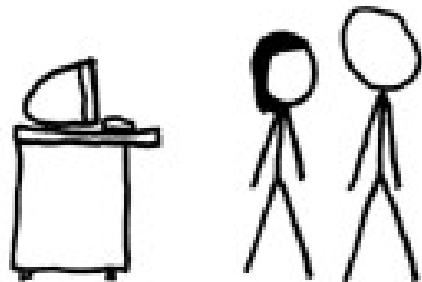


BUT TO FIND THEM WE'D HAVE TO SEARCH THROUGH 200 MB OF EMAILS LOOKING FOR SOMETHING FORMATTED LIKE AN ADDRESS!

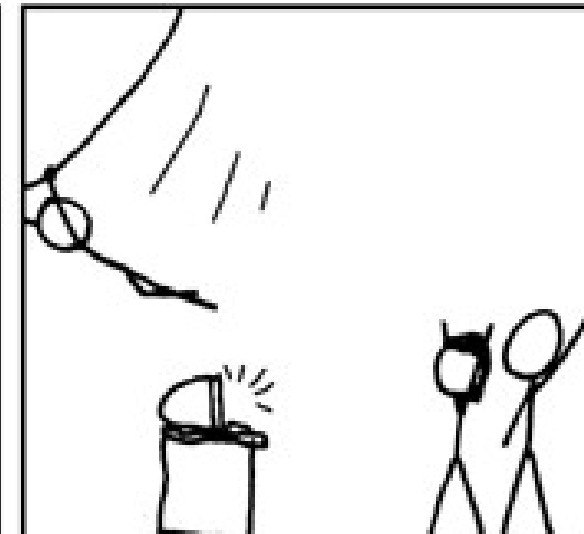
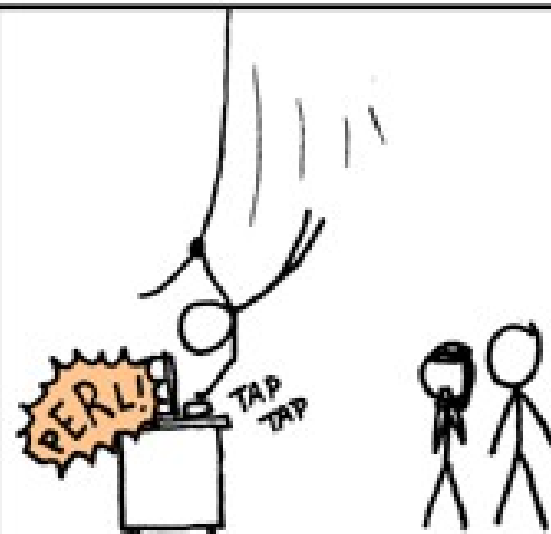
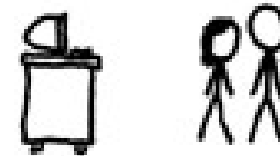


IT'S HOPELESS!

EVERYBODY STAND BACK.



I KNOW REGULAR EXPRESSIONS.



Упражнение 1

- Напишете програма, която извлича от текстов файл всички валидни телефони на български оператори.
- валидни формати:
 - 088 88888888
 - 0888 888 888
 - +359888 88-88-88
 - 00359 88 888 88 88
 - и вариации на горните с различните представки

Упражнение 2

- Напишете програма, която прочита текст от входен файл. Обръща реда на всички параграфи. Обръща реда на всички изречения в съответните параграфи. Прави всяка първа буква от изречение малка и всяка последна главна. И записва резултата в друг файл.