

Лабораторная работа 8. Регулярные выражения

<https://tproger.ru/articles/regexp-for-beginners/>
<https://regex101.com/>

Тесты - 1 балл

Чтение из файла - 2 балла

Основная часть - 7 баллов

1 вариант. Убрать повторяющиеся пробелы и знаки табуляции, оставить по одному пробелу между словами и по два между предложениями.

Пример:

Extra spaces => Extra spaces

Sentence. Sentence. => Sentence. Sentence.

<http://callumacrae.github.io/regex-tuesday/challenge12.html>

2 вариант. Выбрать все валидные IPv4 - адреса

Выбрать IPv4 адреса во всех возможных, представлениях: десятичном, шестнадцатеричном и восьмеричном. С точками и без. Подробнее про IP адреса можно узнать [в википедии](#)

Примеры:

99.198.122.146 — Да

0xFF.255.0377.0x12 — Да

256.256.256.256 — Нет

<http://callumacrae.github.io/regex-tuesday/challenge6.html>

3 вариант.

Выбрать все валидные доменные имена для протоколов http и https, с необязательным слешем в конце. Специальные символы не используются.

Примеры:

http://example.com/ — Да

example.com — Нет

<http://callumacrae.github.io/regex-tuesday/challenge7.html>

4 вариант.

Выбрать валидные даты (год с 1900 по 2020) в формате YYYY/MM/DD HH:MM(:SS)
(секунды не обязательно будут). Високосные годы не учитывать, считать, что в любом
месяце 30 дней.

<http://callumacrae.github.io/regex-tuesday/challenge3.html>

5 вариант.

Выбрать валидные даты (год с 1900 по 2020) в формате YYYY/MM/DD
Учитывать високосные годы и 28/30/31 день в месяцах

<http://callumacrae.github.io/regex-tuesday/challenge3.html>

6 вариант

Валидные числа (с точкой/запятой/пробелом для разделения разрядов)
10,000,000.45 или 10 000 000,45 - да
123.456.789. - нет

<http://callumacrae.github.io/regex-tuesday/challenge5.html>

7 вариант

Валидные e-mail адреса.

8 вариант

Валидные белорусские телефонные номера.
коды: 29, 25, 33, 44, 17
+375441234567
80441234567
+375-44-123-45-67
+375(44)123-45-67

9 вариант

Преобразовать текст, обрамленный в звездочки, в курсив. Не трогать текст в
двойных звездочках (жирный).

Примеры:

```
*this is italic*" => <em>this is italic</em>
```

****bold text (not italic)** => **bold text (not italic)****

<http://callumacrae.github.io/regex-tuesday/challenge4.html>

10 вариант

Вывод всех цитат в тексте (в кавычках). Могут располагаться на нескольких строках.

11 вариант

Найти все комментарии в Java-коде (`//`, `/* */`, `/** */`)

12 вариант

Разделить предложение на токены (слова)

Есть несколько правил:

- Несколько слов в кавычках должны попасть в один токен

This "huge test" is pointless => this,huge test,is,pointless

- Слова написанные через дефис так же попадают в один токен.

Слова написанные через несколько дефисов (тире), или имеющие дефис в начале или в конце попадают в отдельные токены.

Suzie Smith-Hopper test--hyphens => Suzie,Smith-Hopper,test,hyphens.

- Сокращения (стяжения) попадают в один токен

I can't do it => I,can't,do,it.

- Вся пунктуация кроме апострофов и дефисов должна быть убрана.

Too long; didn't read => Too,long,didn't,read.

<http://callumacrae.github.io/regex-tuesday/challenge10.html>

13 вариант

Вывести только валидные значения переменных.

Перечень основных правил именования переменных:

- Имена переменных должны начинаться с буквы, знака \$ или знака подчеркивания `_`. Имена не могут начинаться с цифр!
- После первого символа в имени могут идти любые комбинации букв, цифр, знаков подчеркивания и \$.

- Нет ограничения на количество символов, которое может содержать имя переменной.
- Имена переменных в Java регистрозависимые - переменные с именами foo и FOO это две различные переменные.

Вот примеры *допустимых* имен:

```
int a;  
int b1;  
int _c;  
int $d;  
int i_am_damn_long_variable_identifier;
```

А вот примеры *недопустимых*:

```
int :e;  
int /d;  
int f#;  
int .g;  
int 14h;
```