ДЕКЛАРАТИВНОЕ ПРОГРАММИРОВАНИЕ

Регулярные выражения

Регулярные выражения

.	Любой одиночный символ
[abc]	Любой из указанных символов: a, b, c
[^abc]	Любой символ, кроме указанных: a, b, c
[a-z]	Любая буква в указанном диапазоне
[[:digit:]]	Любая цифра ([0-9])
[[:alpha:]]	Любая буква (латиница)
[[:alnum:]]	Любая буква или цифра
[[:space:]]	Любой пробельный символ
\	Экранирование специальных символов
^	Начало строки
$	Конец строки
|	Или

https://regex101.com/ (PCRE)

Квантификаторы

- * 0 или более повторений
- + 1 или более повторений
- ? 0 или 1 повторение
- {n} Ровно n повторений
- {n,} n или более повторений
- {n,m} От n до m повторений

Контекст важен

`.*` – последовательность любых символов

`[.*]` – один символ `.` или `*`

`^no` – строка, начинающаяся с `no`

`[^no]` – символ, который не является ни `n` ни `o`

`A-Z` – строка из трех символов `A-Z`

`[A-Z]` – один из символов `A, B, C, ..., X, Y,` или `Z`

Еще примеры

- ^[a-zA-Z_][a-zA-Z_0-9]*$ -- имена переменных в Си
- (2[0-3]|1[0-9]|0[0-9]):[0-5][0-9] – время
- <[hH][1-6]> -- HTML-заголовки <h1> <H1> <h2> …

Упражнения

Напишите регулярные выражения для нахождения:

1. за символом **a** следует один или более символов **b**
2. дата в формате yyyy-mm-dd, где y,m,d – любые цифры
3. слово (состоит только из букв латинского алфавита) длины не менее 3 символов и начинающееся на гласную букву
4. слово с заглавной буквы
5. слово, в котором есть перенос на другую строку

Библиотеки Haskell

Установка

- regex-posix
  - cabal update
  - cabal install regex-posix
- regex-pcre

  см. инструкции в classroom

Использование

```
import Text.Regex.Posix
```

=~

```
ghci> import Text.Regex.Posix
ghci> :t (=~)
(=~)
  :: (RegexMaker Regex CompOption ExecOption source,
      RegexContext Regex source1 target) =>
     source1 -> source -> target
```

((=~) :: text -> regex -> result)

Наличие вхождения

```
ghci> "abc" =~ "[a-c]*" :: Bool
True
ghci> "abcd" =~ "[a-c]*" :: Bool
True
ghci> "defg" =~ "[a-c]*" :: Bool
True
ghci> "defg" =~ "^[a-c]*$" :: Bool
False
ghci> "abcd" =~ "^[a-c]*$" :: Bool
False
ghci> "abc" =~ "^[a-c]*$" :: Bool
True
```

Количество вхождений

```
ghci> "honorificabilitudinitatibus" =~ "a" :: Int
2
ghci> "honorificabilitudinitatibus" =~ "[aeiouy]" :: Int
13
ghci> "honorificabilitudinitatibus" =~ "[^aeiouy]" :: Int
14
ghci> "honorificabilitudinitatibus" =~ "[a-z]" :: Int
27
```

Упражнение

Почему?

```
ghci> "very long long long text" =~ ".*" :: Int
2
```

Первое вхождение

```
ghci> "A Skateboard is reduced 5% in price " =~ "[0-9]%" :: String
"5%"
ghci> "A Skateboard is reduced 25% in price" =~ "[0-9]%" :: String
"5%"
ghci> "A Skateboard is reduced 25% in price" =~ "[0-9]{1,2}%" :: String
"25%"
```

Интервал (начало, длина)

```
ghci> "contrariwise" =~ "wise" :: (Int, Int)
(8,4)
ghci> "contrariwise" =~ "WISE" :: (Int, Int)
(-1,0)
```

(префикс, вхождение, постфикс)

```
ghci> "it's my own invention" =~ "own" :: (String, String, String)
("it's my ","own"," invention")
ghci> "it's my own invention" =~ "OWN" :: (String, String, String)
("it's my own invention","","")
```

Все вхождения

```
> getAllTextMatches$"A Skateboard is reduced 25% in price" =~"[a-zA-Z]+"::[String]
["A","Skateboard","is","reduced","in","price"]
```

Упражнение

Что не так?

```
import Text.Regex.Posix

isCorrectHaskellFile :: String -> Bool
isCorrectHaskellFile filename = filename  =~ ".hs"
```

Упражнение

Напишите функцию, которая проверяет, является ли строка приветствием (Начинается со слова Hi в любом регистре)

- Hi Alex how are you doing
- hI dave how are you doing
- Hi, Alex!
- Good by Alex
- hidden agenda

Упражнения

- Проверить, является ли строка двоичным числом
  - 1011110 - да
  - 101018 - нет
  - hello – нет
- Проверить, является ли строка номером телефона в формате +7-xxx-xxx-xxxx (где x – любая цифра)
- Проверить, что строка начинается и заканчивается одним и тем же словом. Извлечь это слово.
  - **she** said no one is as lovely as **she** – да
  - ow, I don't know – нет

<Упражнение>

Верните уникальные имена всех html-тегов в **файле**

Текст:

<**p**>This <**br**> is a paragraph <**br**> with <**br**> line breaks</p>

<**a** href="https://www.wikipedia.org/">A link to Wikipedia!

<**p**>Oh well, <**span** lang="fr">c'estla vie, as they say in France.</p>

Теги:

p, br, a, span

</Упражнение>