

*Государственное образовательное учреждение высшего
профессионального образования*

**«Московский государственный технический
университет имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)**

РУБЕЖНЫЙ КОНТРОЛЬ №2

ПО КУРСУ «АНАЛИЗ АЛГОРИТМОВ»

Конечные автоматы и регулярные выражения

Выполнил: Мирзоян С.А., гр. ИУ7-55Б

Преподаватели: Волкова Л.Л., Строганов Ю.В.

2020 г.

Оглавление

Введение	2
1 Аналитический раздел	3
1.1 Регулярные выражения	3
1.2 Вывод	3
2 Технологический раздел	4
2.1 Требования к программному обеспечению	4
2.2 Средства реализации	4
2.3 Листинг кода	4
2.4 Вывод	6
Заключение	7
Литература	8

Введение

Задача лабораторной работы состоит в том, что нужно при регулярного выражения написать программу вывода текста внутри тегов `/` и `<str>/</str>` находящихся внутри тега `<p>`

1. Аналитический раздел

1.1 Регулярные выражения

Регулярные выражения — язык поиска подстроки или подстрок в тексте. Для поиска используется паттерн (шаблон, маска), состоящий из символов и метасимволов (символы, которые обозначают не сами себя, а набор символов).

Это довольно мощный инструмент, который может пригодиться во многих случаях — поиск, проверка на корректность строки и т.д. Спектр его возможностей трудно уместить в одну статью[?].

1.2 Вывод

В данном разделе были описаны конечные автоматы и регулярные выражения.

2. Технологический раздел

В данном разделе будут предъявлены требования к программному обеспечению, средства реализации и листинги кода.

2.1 Требования к программному обеспечению

Программное обеспечение должно реализовывать поиск подстроки в строке с помощью конечного автомата и регулярного выражения. Причем, для конечного автомата не должно использоваться никаких библиотек, а реализация регулярного выражения может быть осуществлена с использованием библиотек.

2.2 Средства реализации

Для выполнения поставленной задачи был использован язык программирования Python. Среда для разработки IDLE. Для измерения времени была взята функция `time.time()` из библиотеки `time`.

Данный язык обусловлен тем, что функции необходимые для реализации регулярного выражения находятся в встроенной библиотеке `re`.

2.3 Листинг кода

В данном подразделе представлены листинги кода реализации конечного автомата и регулярного выражения, которые были представлены в аналитическом разделе.

Листинг 2.1: Реализация регулярного выражения

```
1 import re
2
3 code_to_search = '<p> Bold Stuff <> texttext gogogo </> <b> (assf)lololo </b> </p> <k> <> texttext gogogo </> xoxoxxoxoxo</k> '
4 result = ''
5
6 pattern = re.compile(r'<p>([^\"]*)</p>')
7 matches = pattern.findall(code_to_search)
8 print(':', matches[0])
9 try:
10
11     pattern = re.compile(r'<>([^\"]*)</>')
12     gstr = pattern.findall(matches[0])
13     print(':', gstr[0])
14 except:
15     print(' - ')
16 try:
```

```
17     pattern = re.compile(r'<b>([^\"]*)</b>')
18     gb = pattern.findall(matches[0])
19     print('B', gb[0])
20 except:
21     print(' - ')
```

2.4 Вывод

В данном разделе были предъявлены требования к программному обеспечению, средства реализации и листинги кода.

Заключение

Были изучены основные принципы работы регулярных выражений для поиска подстроки в тексте, а также разработана программа, решающая задачу поиска в тексте html-кода текста внутри определенных тегов

Литература

Регулярные выражения [Электронный ресурс]. - Режим доступа: <https://habr.com/ru/company/badoo/blog/343310/>, свободный. (Дата обращения: 5.2.2020 г.)