3 Лабораторная работа №3. «Регулярные выражения и языки разметки документов. Python»

Для определения варианта используйте свой табельный номер, которые можно найти в ИСУ. (*Пример номера:* 125598)

Задание 1. (20% из 100)

1) Реализуйте программный продукт на языке Python, используя регулярные выражения по варианту, представленному в таблице.

2) Для своей программы придумайте минимум 5 тестов. Каждый тест является отдельной сущностью, передаваемой регулярному выражению для обработки. Для каждого теста необходимо самостоятельно (без использования регулярных выражений) найти правильный ответ. После чего сравнить ответ, выданный программой, и полученный самостоятельно.

3) Программа должна считать количество смайликов определённого вида (вид смайлика описан в таблице вариантов) в предложенном тексте. Все смайлики имеют такую структуру:

[*глаза*][*нос*][*рот*].
Вариантом является различные наборы глаз, носов и ртов.

Номер в ИСУ % 5	Глаза	Номер в ИСУ % 4	Нос	Номер в ИСУ % 7	Рот
0	:	0	-	0	(
1	;	1	<	1	)
2	X	2	-{	2	O
3	8	3	<{	3	|
4	=			4	\
				5	/
				6	P

*Пример смайлика: 8<{*P

Задание 2. (40% из 100)

1) Реализуйте программный продукт на языке Python, используя регулярные выражения по варианту, представленному в таблице.

2) Для своей программы придумайте минимум 5 тестов.

3) Протестируйте свою программу на этих тестах.

<table>
<tr><th>Номер в ИСУ % 6</th><th>Задание</th></tr>
<tr><td>0</td><td>Написать регулярное выражение, которое проверяет корректность email и в качестве ответа выдаёт почтовый сервер (почтовый сервер – часть email идущая после «@»).
Для простоты будем считать, что почтовый адрес может содержать в себе буквы, цифры, «.» и «_», а почтовый сервер только буквы и «.». При этом почтовый сервер, обязательно должен содержать верхний уровень домена («.ru», «.com», etc.)

Пример:

<table>
<tr><td>Ввод</td><td>Вывод</td></tr>
<tr><td>students.spam@yandex.ru</td><td>yandex.ru</td></tr>
<tr><td>example@example</td><td>Fail!</td></tr>
<tr><td>example@example.com</td><td>example.com</td></tr>
</table>
</td></tr>
<tr><td>1</td><td>Студент Вася очень любит курс «Компьютерная безопасность». Однажды Васе задали домашнее задание зашифровать данные, переданные в сообщение. Недолго думая, Вася решил заменить все целые числа на функцию от этого числа. Функцию он придумал не сложную $3x2+5$,где $x-$исходное число. Помогите Васе с его домашним заданием.
Пример:

<table>
<tr><td>Ввод</td><td>Вывод</td></tr>
<tr><td>20 + 22 = 42</td><td>1205 + 1457 = 5297</td></tr>
</table>
</td></tr>
<tr><td>2</td><td>Вывесили списки стипендиатов текущего семестра, которые представляют из себя список людей ФИО и номер группы этого человека. Вы решили подшутить над некоторыми из своих одногруппников и удалить их из списка.
С помощью регулярного выражения найдите всех студентов своей группы, у которых инициалы начинаются на одну и туже букву и исключите их из списка.
Пример (группа P000):

<table>
<tr><td>Ввод</td><td>Вывод</td></tr>
<tr><td>Петров П.П. P000
Анищенко А.А. P33113</td><td>Анищенко А.А. P33113
Примеров Е.В. P000</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td><table><tr><td>Примеров Е.В. Р000
Иванов И.И. Р000</td><td></td></tr></table></td></tr>
<tr><td>3</td><td>Дан текст. Необходимо найти в нём любой фрагмент, где сначала идёт слово «ВТ», затем не более 4 слов, и после этого идёт слово «ИТМО». Для простоты будем считать словом любую последовательность букв, цифр и знаков «_» (то есть символов \w).
Пример:
<table><tr><td>Ввод</td><td>Вывод</td></tr><tr><td>А ты знал, что ПИиКТ – лучший факультет в ИТМО?</td><td>ПИиКТ лучший факультет в ИТМО</td></tr></table></td></tr>
<tr><td>4</td><td>Дан текст. Требуется найти в тексте все фамилии, отсортировав их по алфавиту.
Фамилией для простоты будем считать слово с заглавной буквой, после которого идут инициалы.

Пример:
<table><tr><td>Ввод</td><td>Вывод</td></tr><tr><td>Студент Вася вспомнил, что на своей лекции Болдырева Е.А. упоминала про старшекурсников, которые будут ей помогать: Анищенко А.А. и Машина Е.А.</td><td>Анищенко
Болдырева
Машина</td></tr></table></td></tr>
<tr><td>5</td><td>Анатолий выложил пост с расписанием доп. занятий по информатике, но везде перепутал время. Поэтому нужно заменить все вхождения времени на строку (TBD).
Время – это строка вида HH:MM:SS или HH:MM, в которой HH – число от 00 до 23, а MM и SS – число от 00 до 59.

Пример:
<table><tr><td>Ввод</td><td>Вывод</td></tr><tr><td>Уважаемые студенты! В эту субботу в 15:00 планируется доп. занятие на 2 часа. То есть в 17:00:01 оно уже точно кончится.</td><td>Уважаемые студенты! В эту субботу в (TBD) планируется доп. занятие на 2 часа. То есть в (TBD) оно уже точно кончится.</td></tr></table></td></tr>
</table>

Задание 3. (40% из 100)

1. Определить номер варианта как остаток деления номера в ИСУ на 36. В случае, если в данный день недели нет занятий, то увеличить номер варианта на восемь.
2. Изучить форму Бэкуса-Наура.
3. Изучить особенности протоколов и форматов обмена информацией между системами: JSON, YAML, XML.
4. Понять устройство страницы с расписанием для своей группы: https://itmo.ru/ru/schedule/0/P3110/schedule.htm
5. Исходя из структуры расписания конкретного дня, сформировать файл с расписанием в формате, указанном в задании в качестве исходного.
6. **Обязательное задание**: написать программу на языке Python 3.x, которая бы осуществляла парсинг и конвертацию исходного файла в новый.
7. Нельзя использовать готовые библиотеки, в том числе регулярные выражения в Python и библиотеки для загрузки XML-файлов.

8. **Дополнительное задание №1** (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).

a) Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов.

b) Переписать исходный код, применив найденные библиотеки. Регулярные выражения также нельзя использовать.

c) Сравнить полученные результаты и объяснить их сходство/различие.

9. **Дополнительное задание №2** (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).

a) Переписать исходный код, добавив в него использование регулярных выражений.

b) Сравнить полученные результаты и объяснить их сходство/различие.

Варианты для задания 3:

№ варианта	Исходный формат	Результирующий формат	День недели
0	YAML	XML	Понедельник
1	JSON	XML	Понедельник
2	XML	JSON	Понедельник
3	JSON	YAML	Понедельник
4	YAML	JSON	Понедельник
5	XML	YAML	Понедельник
6	YAML	XML	Вторник
7	JSON	XML	Вторник
8	XML	JSON	Вторник
9	JSON	YAML	Вторник
10	YAML	JSON	Вторник
11	XML	YAML	Вторник
12	YAML	XML	Среда
13	JSON	XML	Среда
14	XML	JSON	Среда
15	JSON	YAML	Среда
16	YAML	JSON	Среда
17	XML	YAML	Среда
18	YAML	XML	Четверг
19	JSON	XML	Четверг
20	XML	JSON	Четверг
21	JSON	YAML	Четверг
22	YAML	JSON	Четверг
23	XML	YAML	Четверг
24	YAML	XML	Пятница
25	JSON	XML	Пятница
26	XML	JSON	Пятница
27	JSON	YAML	Пятница
28	YAML	JSON	Пятница
29	XML	YAML	Пятница
30	YAML	XML	Суббота
31	JSON	XML	Суббота
32	XML	JSON	Суббота
33	JSON	YAML	Суббота
34	YAML	JSON	Суббота
35	XML	YAML	Суббота

Требования и состав отчёта

1. Отчёт должен быть выполнен на листе размером А4 с использованием Microsoft Word, Libre Office и т.п.
2. Отчёт должен начинаться с титульного листа с названием вуза и факультета, номером и названием лабораторной работы, вариантом, ФИО студента, № группы, ФИО преподавателя, городом и годом.
3. Отчет должен содержать автособираемое оглавление (обязательные разделы – Задание, Основные этапы выполнения, Вывод, Список использованных источников).
4. Отчет должен содержать изображения, оформленные и подписанные в соответствии с ГОСТ 7.32-2017 «Отчет о научно-исследовательской работе. Структура и правила оформления» (минимум одно изображение), и список литературы со ссылками на источники (минимум два источника).
5. Станицы отчёта должны быть пронумерованы, при этом нумерация на титульном листе не должна ставиться.
6. В отчёте нужно кратко представить описание решаемой задачи, полный листинг программ .py, содержание файла в исходном и результирующем форматах.
7. Отчёт предоставить в электронном виде. По просьбе преподавателя нужно быть готовыми скомпилировать и запустить свою программу.

Подготовка к защите

1. Изучить и закрепить необходимый материал из следующего пособия:

Лямин А.В., Череповская Е.Н. Объектно-ориентированное программирование. Компьютерный практикум. – СПб: Университет ИТМО, 2017. – 143 с. – Режим доступа: https://books.ifmo.ru/file/pdf/2256.pdf.

2. Прочитать и повторить информацию из статьи в Википедии: https://ru.wikipedia.org/wiki/Форма_Бэкуса_—_Наура.
3. Прочитать и повторить информацию из статьи «Пишем изящный парсер на Питоне»: https://habr.com/ru/post/309242/.
4. Уметь объяснить каждую строку программы, представленной в отчёте.
5. При защите отчёта надо уметь отвечать на вопросы по работе программы, вопросы по материалам лекций №3 и №4 и следующие вопросы:

1) В чём разница между Markup и Markdown?

2) В чём заключается особенность PROTOBUF по сравнению с другими форматами?

3) Чем формат CSV отличается от формата TSV?

4) Чем обусловлено постоянное появление новых форматов представления данных?

5) Каким образом в формате XML представляются символы ‘>’ и ‘<’?

6) Что такое сериализация данных?

7) Каким образом в YAML обозначаются комментарии?

8) Пояснить, как в языке разметки Markdown создать заголовки разных уровней, оформить код, вывести полужирный, курсивный и зачеркнутый текст?

9) Какие форматы обмена данных используются в современных популярных мессенджерах (Viber, WhatsApp, Telegram и т.д.)?

10) Как расшифровывается аббревиатура SVG?

11) Привести пример использования в языке HTML тега, который создаёт гиперссылку на url.

12) Какие две структуры может представлять собой в закодированном виде JSON-текст?