Национальный исследовательский университет информационных технологий, механики и оптики  
Факультет программной инженерии и компьютерной техники

Лабораторная работа № 4  
«Регулярные выражения»

Выполнил: Лысенко Данила Сергеевич  
Группа: P3110  
Вариант: 18

Преподаватель: Балакшин Павел Валерьевич

Санкт-Петербург  
2020

**Задание на 70 баллов**

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. Пример тестов приведён в таблице.

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.

|  |  |
| --- | --- |
| Ввод | Вывод |
| Студент Вася вспомнил, что на своей лекции Балакшин П.В. упоминал про старшекурсников, которые будут ему помогать: Анищенко А.А. и Машина Е.А. | Анищенко  Балакшин  Машина |

**Доп. задание №1 (+12 баллов)**

1) Для выполнения задания скачайте текстовый документ по ссылке, соответствующей вашему варианту.

|  |  |  |
| --- | --- | --- |
| Номер в ИСУ % 30 | Ссылка на текст | Описание |
| 18 | https://drive.google.com/file/d/1lBo2ExJSIWc-goGx4Ton517dlIr2Q8qC/view | Макбет |

2) Реализуйте программный продукт на языке Python, используя регулярные выражения, выделите все предложения из вашего текста, соответствующие критерию по вашему варианту.

3) Придумайте способ, как проверить корректность работы вашего регулярного выражения и проверьте его.

|  |  |
| --- | --- |
| Номер в ИСУ % 4 | Задание |
| 2 | Все предложения, в которых используется две и более запятых |

**Доп. задание №2 (+18 баллов)**

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.

|  |  |
| --- | --- |
| Номер в ИСУ % 6 | Задание |
| 0 | 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:   |  |  | | --- | --- | | Ввод | Вывод | | students.spam@yandex.ru | yandex.ru | | example@example | Fail! | | example@example.com | example.com | |

**Выполнение заданий**

1.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | **import** **re**  **def** **task1**():  **for** i **in** range(**1**, **6**):  input = open('tests\_1/' + str(i) + '.txt', 'r', encoding="utf-8")  nextLine = input.readline()  lines = ''  **while** nextLine:  lines = lines + nextLine  nextLine = input.readline()  result = re.findall(r'[А-Я]\w+(?= +[А-Я]\.[А-Я]\.$| +[А-Я]\.[А-Я]\.\s)', lines)  result.sort()  **print**("**\n**Результаты")  **for** v **in** result:  **print**(v)  task1() |

2.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | **import** **re**  **def** **task2**():  input = open('Macbeth.txt', 'r', encoding="utf-8")  nextLine = input.readline()  lines = ''  **while** nextLine:  lines = lines + nextLine  nextLine = input.readline()  result = re.findall(r'(?:[^!.?,]+,[^!.?,]+)+,[^!.?,]+[!.?]', lines)  **for** v **in** result:  **print**(v)  task2() |

3.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | **import** **re**  **def** **task3**():  **for** i **in** range(**1**, **11**):  input = open('tests\_3/' + str(i) + '.txt', 'r', encoding="utf-8")  nextLine = input.readline()  lines = ''  **while** nextLine:  lines = lines + nextLine  nextLine = input.readline()  result = re.match(r'^[a-zA-Z0-9\_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z-.]+', lines)  **if** result:  result = re.search(r'(?<=@)[a-zA-Z0-9-]+\.[a-zA-Z-.]+', lines)  **print**(result.group(**0**))  **else**:  **print**('incorrect')  task3() |

**Вывод**

В ходе выполнения лабораторной работы разобрался с основными принципами работы регулярных выражений и их реализации на языке программирования Python.