|  |  |
| --- | --- |
| **К Г Э У** | МИНИСТЕРСТВО ВЫСШЕГО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»**  (ФГБОУ ВО «КГЭУ») |

**Кафедра «Информатики и информационных управляющих систем»**

ОТЧЕТ

Дисциплина «Алгоритмизация и программирование»

Выполнил: Соловьёв Л.А.  
Группа: ПИ-1-22  
Преподаватель: Шорина Т.В.

Казань, 2024

***Постановка задачи***

1. Изучение работы функций библиотеки re

Re – это расширение языка Python, позволяющее работать с регулярными выражениями. Её чаще всего используют для обработки текста.

**Основные команды модуля re**

|  |  |
| --- | --- |
| re.search(pattern, string) | Находит в строке string первую подстроку, подходящую под шаблон pattern |
| re.fullmatch(pattern, string) | Проверяет, подходит ли строка string под шаблон pattern |
| re.split(pattern, string, maxsplit=0) | Аналог метода split(), но разделение происходит по подстрокам, подходящим под шаблон pattern |
| re.findall(pattern, string) | Находит в строке string все непересекающиеся шаблоны pattern |
| re**.**finditer(pattern, string) | Итератор всех непересекающихся шаблонов pattern в строке string(выдаёт match-объект) |
| re.sub(pattern, repl, string, count=0) | Заменяет в строке string все непересекающиеся шаблоны pattern на repl |
| re.compile(pattern, flags=0) | Компилирует шаблон регулярного выражения в объект регулярного выражения, к которому применимы функции match(), search() и т.д. |
| re.sub(pattern, repl, string, count=0, flags=0) | Возвращает строку, полученную заменой крайних левых неперекрывающихся вхождений pattern в string с заменой repl. Если шаблон не найден, string возвращается без изменений |
| re.subn(pattern, repl, string, count=0, flags=0) | Аналогичен sub(), но возвращает кортеж (new\_string, number\_of\_subs\_made) |

**Примеры обработки данных с использованием модуля**

**Задача 1.** Написать программу, которая будет искать все вхождения email адресов в строке.

Алгоритм выполнения:

Для начала, импортируем библиотеку re с помощью команды import re. Затем создаем строку text, содержащую несколько email адресов. После этого, используя функцию re.findall() находим все email адреса в тексте. Затем, выводим массив со всеми email адресами с помощью функции print.

Ожидаемый итог:

После выполнения кода ожидается, что на выходе мы получим массив, содержащий email адреса, содержащиеся в строке.

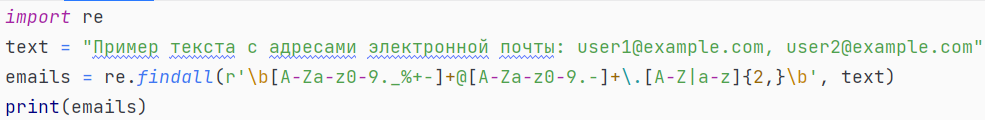


Рисунок 1.1 – Код к задаче №1



Рисунок 1.2 – Вывод к задаче №1

**Задача 2.** Написать программу, которая будет искать все вхождения хэштегов в строке.

Алгоритм выполнения:

Для начала, импортируем библиотеку re с помощью команды import re. Затем создаем строку text, содержащую несколько хэштэгов. После этого, используя функцию re.findall() находим все хэштеги в тексте. Затем, выводим массив со всеми хэштэгами с помощью функции print.

Ожидаемый итог:

После выполнения кода ожидается, что на выходе мы получим массив, содержащий хэштеги, содержащиеся в строке.



Рисунок 2.1 – Код к задаче №2



Рисунок 2.2 – Вывод к задаче №2

**Задача 3.** Написать программу, которая будет проверять валидность пароля.

Алгоритм выполнения:

Для начала, импортируем библиотеку re с помощью команды import re. Затем создаем функцию check\_password(password). После этого создаем переменную, которая содержит пароль и проверяем этот пароль, используя созданную функцию. Затем, выводим результат работы функции с помощью функции print.

Ожидаемый итог:

После выполнения кода ожидается, что на выходе мы логическое значение, означающее корректность созданного пароля.



Рисунок 3.1 – Код к задаче №3



Рисунок 3.2 – Вывод к задаче №3

**Задача 4.** Написать программу, которая будет форматировать даты в тексте.

Алгоритм выполнения:

Для начала, импортируем библиотеку re с помощью команды import re. Затем создаем функцию change\_date\_format(text). После этого создаем переменную, которая содержит текст и форматируем этот текст, используя созданную функцию. Затем, выводим результат работы функции с помощью функции print.

Ожидаемый итог:

После выполнения кода ожидается, что на выходе все даты будут подведены под другой формат.

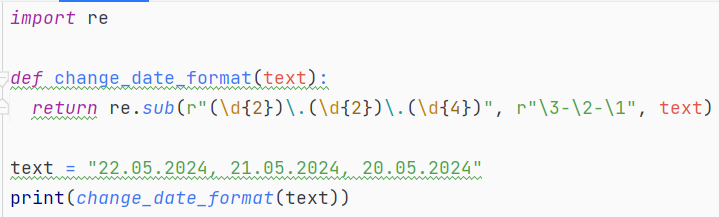


Рисунок 4.1 – Код к задаче №4



Рисунок 4.2 – Вывод к задаче №4

**Задача 5.** Написать программу, которая будет проверять корректность введенного номера телефона.

Алгоритм выполнения:

Для начала, импортируем библиотеку re с помощью команды import re. Затем создаем функцию check\_phone\_number(phone\_number). После этого создаем переменную, которая содержит телефон и проверяем этот телефон, используя созданную функцию. Затем, выводим результат работы функции с помощью функции print.

Ожидаемый итог:

После выполнения кода ожидается, что на выходе все даты будут подведены под другой формат.



Рисунок 5.1 – Код к задаче №5



Рисунок 5.2 – Вывод к задаче №5

**Вывод**

Библиотека re является незаменимым инструментом для обработки текстовых данных в Python. Она позволяет эффективно находить, извлекать и модифицировать текст, используя мощные возможности регулярных выражений. Изучение re даёт разработчикам Python широкие возможности для работы с текстами и создания гибких и эффективных решений.  
Все изученные материалы были успешно применены в практических задачах, что позволило более глубоко понять возможности и преимущества использования библиотеки re в разработке на языке программирования Python.

**Список использованных источников**

1. <https://docs.python.org/3/library/re.html>