

## Лабораторная работа №6

По дисциплине: Основы программной инженерии

Грובהва Софья

ПИЖ-6-о-20-1

Программа task1.py, код и результат работы

Задача: Дано предложение. Напечатать все его буквы «и».

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':
    lst = []
    sentnc = input('Введите предложение: ')
    for chr in sentnc:
        if chr.lower() == 'и':
            print(chr)
```

Рисунок 6.1 – Код программы task1.py

```
D:\Пользователь\Desktop\ОПИ\lab6\venv\Scripts\python.exe D:/Пользователь/Desktop/ОПИ/lab6/task/task1.py
Введите предложение: Я вас люблю, Любовь моя, быть может, я даже мой устал не совсем, но пусть она вас больше не тревожит, я не хочу печалить вас ничем.
и
и
и
и
и
Process finished with exit code 0
```

Рисунок 6.2 – Результат работы программы task1.py

Программа task2.py, код и результат работы

Задача: Дано предложение. Определите количество букв н, предшествующих первой запятой предложения. Рассмотреть два случая:

- известно, что запятые в предложении есть;
- запятых в предложении может не быть.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':
    sentnc = input('Введите предложение: ')
    flag = False
    num = 0
    for chr in sentnc:
        if chr.lower() == 'н':
            num += 1
        elif chr == ',':
            flag = True
            if flag:
                break
    print(num)
```

Рисунок 6.3 – Код программы task2.py

```
D:\Пользователь\Desktop\ОПИ\lab6\venv\Scripts\python.exe D:/Пользователь/Desktop/ОПИ/lab6/task/task2.py
Введите предложение: Самое главное в жизни, чтобы семья была счастлива и здорова.
2
Process finished with exit code 0
```

Рисунок 6.4 – Результат работы программы task2.py

Программа task3.py, код и результат работы

Задача: Дано предложение. Удалить из него все повторяющиеся буквы, оставив их первые вхождения, т.е. в слове должны остаться только различные буквы.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':
    word = input('Введите слово: ')
    lst = []
    for ind, chr in enumerate(word):
        flag = True
        for elem in lst:
            if elem == chr.lower():
                flag = False
                break
        if flag or ind == 0:
            lst.append(chr)

    print(''.join(lst))
```

Рисунок 6.5 – Код программы task3.py

```
D:\Пользователь\Desktop\ОПИ\lab6\ven
Введите слово: Много букв
Мног букв

Process finished with exit code 0
```

Рисунок 6.6 – Результат работы программы task3.py

### Вопросы для защиты работы

1. Строки в Python - упорядоченные последовательности символов, используемые для хранения и представления текстовой информации.
2. Строки в апострофах и кавычках, экранированные последовательности – служебные символы, “Сырые” строки, строки в тройных апострофах или кавычках.
3. Сложение, умножение. Строковых функций много, вот некоторые из них:  
len() – длина строки  
str() – изменяет тип объекта на string
4. <название строковой переменной> [число от 0 до длины строки - 1]
5. Если s это строка, выражение формы s[m:n] возвращает часть s , начинающуюся с позиции m , и до позиции n , но не включая позицию:
6. Так их легче представить в памяти.
7. s.isitle()
8. if s1 in s2
9. s.find(s2)
10. len(s)
11. s.count(<char>)
12. Они позволяют проще формировать строки. Пример: s = f"Ваш id = {id}"
13. s.find(<sub>[, <start>[, <end>]])
14. "Ваш id = {}".format(id)
15. s.isdigit()
16. 'foo.bar.baz.qux'.rsplit(sep='.')
17. s.islower()
18. s[0].islower()
19. Нет, можно только преобразовать число в строку и уже его прибавить.
20. s = s[::-1]
21. '-'.join[<iterable>]
22. Верхний – s.upper(), нижний – s.lower()
23. s[0].upper() s[len(s)-1].upper()
24. s.isupper()
25. В случае, если надо сохранить символы конца строки.
26. s.replace("что менять", "на что менять")
27. string.endswith(<suffix>[, <start>[, <end>]]) – заканчивается, string.startswith(<suffix>[, <start>[, <end>]]) – начинается.
28. s.isspace()
29. Будет получена копия строки, состоящая из 3 исходных.
30. s.title()
31. string.partition(<sep>) делит строку на основе разделителя. s.partition(<sep>) отделяет от s подстроку длиной от начала до первого вхождения <sep> . Возвращаемое значение представляет собой кортеж из трех частей:  
Часть s до <sep>  
Разделитель <sep>  
Часть s после <sep>
32. Индекс последнего вхождения подстроки в строку.