

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего  
образования  
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ  
УНИВЕРСИТЕТ»**

**Кафедра  
инфокоммуникаций  
Институт цифрового  
развития**

**ОТЧЁТ  
по лабораторной работе №7  
Дисциплина: «Основы программной инженерии»  
Тема: «Работа со списками в языке Python»**

Выполнил: студент 2  
курса группы Пиж-б-о-  
21-1  
Рязанцев Матвей  
Денисович

Ставрополь 2022

Цель работы: приобретение навыков по работе со списками при написании программ с помощью языка программирования Python версии 3.x.

## Выполнение работы

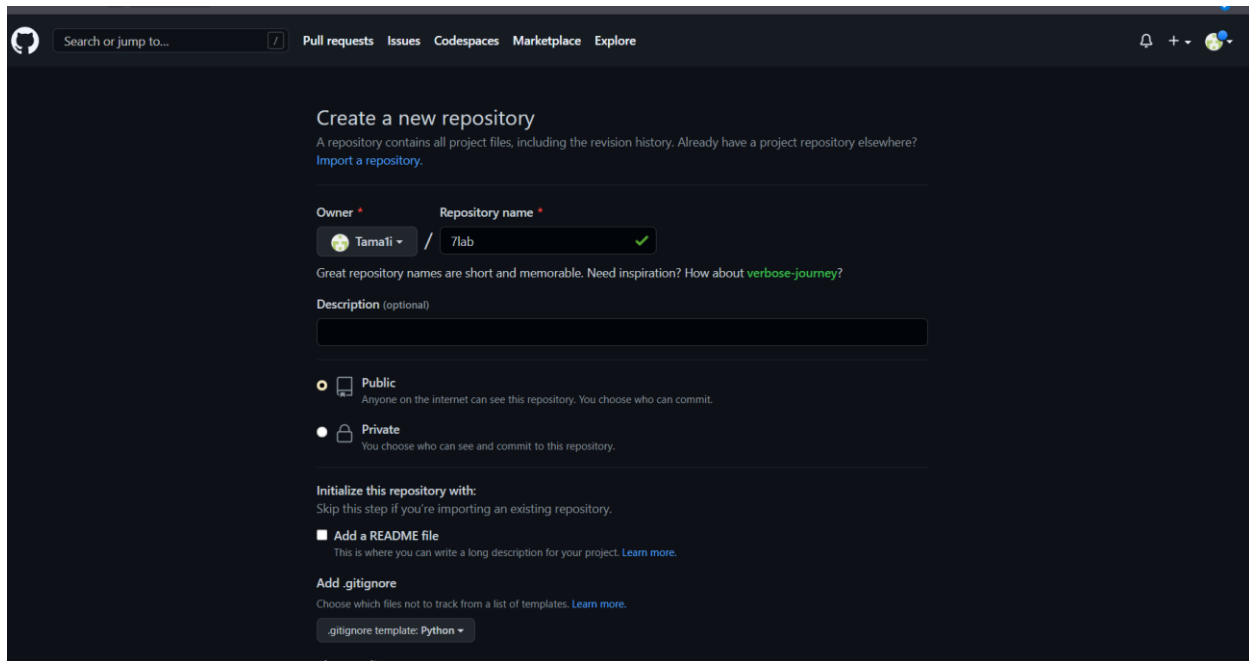


Рисунок 1 -создание репозитория



Рисунок 2 - Организация репозитория по модели ветвления git glow

```
pythonApplication3.py* X
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    # Ввести список одной строкой.
    A = list(map(int, input().split()))
    # Проверить количество элементов списка.
    if len(A) != 10:
        print("Неверный размер списка", file=sys.stderr)
        exit(1)

    # Найти искомую сумму.
    s = sum([a for a in A if abs(a) < 5])
    print(s)
```

Рисунок 3 - Код программы пример 1 с помощью списковых включений следующим образом

```
C:\Users\iyaza\AppData\Local\Programs\Python\Python39\python.exe
1 2 4 5 6 7 8 9 10 3 6
10
Press any key to continue . . .
```

Рисунок 4 – результат работы программы пример 1

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

import sys

if __name__ == '__main__':
    # Ввести список одной строкой.
    A = list(map(int, input().split()))
    # Проверить количество элементов списка.
    if len(A) != 10:
        print("Неверный размер списка", file=sys.stderr)
        exit(1)

    # Найти искомую сумму.
    s = 0
    for item in A:
        if abs(item) < 5:
            s += item
    print(s)
```

Рисунок 5 – код программы пример 1

```
C:\Users\ryaza\AppData\Local\Programs\Pyt
1 2 3 4 5 6 7 8 9 10
10
Press any key to continue . . .
```

Рисунок 6 – результат работы программы

```

import sys

if __name__ == '__main__':
    # Ввести список одной строкой.
    a = list(map(int, input().split()))
    # Если список пуст, завершить программу.
    if not a:
        print("Заданный список пуст", file=sys.stderr)
        exit(1)

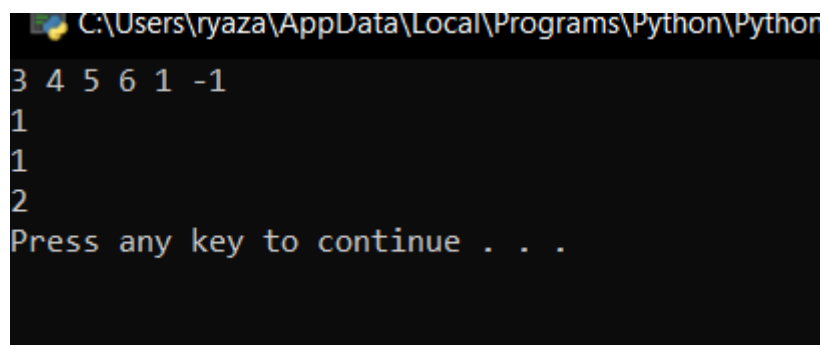
    # Определить индексы минимального и максимального элементов.
    a_min = a_max = a[0]
    i_min = i_max = 0
    for i, item in enumerate(a):
        if item < a_min:
            i_min, a_min = i, item
        if item >= a_max:
            i_max, a_max = i, item

    # Проверить индексы и обменять их местами.
    if i_min > i_max:
        i_min, i_max = i_max, i_min

    # Посчитать количество положительных элементов.
    count = 0
    for item in a[i_min+1:i_max]:
        if item > 0:
            count += 1

```

Рисунок 8 – код программы пример 2



```

C:\Users\ryaza\AppData\Local\Programs\Python\Python
3 4 5 6 1 -1
1
1
2
Press any key to continue . . .

```

Рисунок 9 – результат работы программы

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

import sys

if __name__ == '__main__':
    a = list(map(float, input().split()))
    if len(a) != 10:
        print("error")
        exit(1)
    k = 0
    s = 0
    i = 0

    for i in a:
        if (i % 7 == 0) and (i < 0):
            s += i
            k += 1
    for i in a:
        print(i, " ")
    print(f"sum = {s}    kolvo = {k}")
```

Рисунок 10 – код программы идз 1

```
C:\Users\ryaza\AppData\Local\Programs\Python\
-7 -7 -14 -5 -5 6 5 4 4 4
-7.0
-7.0
-14.0
-5.0
-5.0
6.0
5.0
4.0
4.0
4.0
sum = -28.0    kolvo = 3
Press any key to continue . . .
```

Рисунок 11 – результат работы программы

```

for i in a:
    if abs(i) > k:
        k = abs(i)
        k1 = a.index(i)

    if s == 1:
        s1 += i
    if i > 0:
        s = 1
k = 0

print(f"sum = {s1}    nom = {k1}")
for i in a:
    if ((i >= in1) and (i <= in2)):
        print(i, " ")
    else:
        v[k] = i
        k += 1
for i in range(0,k):
    print(" ", v[i])

```

Рисунок 12 – код программы идз 2

```

C:\Users\ryaza\AppData\Local\Programs\Python\Py
6 6 -7 -8 5 4 -6 4 -1
interval 1 -8
interval 2 -5
sum = -3.0    nom = 3
-7.0
-8.0
-6.0
 6.0
 6.0
 5.0
 4.0
 4.0
-1.0
Press any key to continue . . .

```

Рисунок 13 – результат работы программы

## Контрольные вопросы

1. Что такое строки в языке Python? Строки в Python - упорядоченные последовательности символов, используемые для хранения и представления текстовой информации, поэтому с помощью строк можно работать со всем, что может быть представлено в текстовой форме.

2. Какие существуют способы задания строковых литералов в языке Python? Строки в апострофах и в кавычках, экранированные последовательности, "сырые" строки, строки в тройных апострофах или кавычках

3. Какие операции и функции существуют для строк? Сложение, дублирование, длина строки, извлечение среза и т. д.

4. Как осуществляется индексирование строк? Доступ к символам в строках основан на операции индексирования – после строки или имени переменной, ссылающейся на строку, в квадратных скобках указываются номера позиций необходимых символов.

5. Как осуществляется работа со срезами для строк? Есть три формы срезов. Самая простая форма среза: взятие одного символа строки, а именно, `S[i]` — это срез, состоящий из одного символа, который имеет номер `i`, при этом считая, что нумерация начинается с числа 0. То есть если `S = 'Hello'`, то `S[0]='H'`, `S[1]='e'`, `S[2]='l'`, `S[3]='l'`, `S[4]='o'`. Если указать отрицательное значение индекса, то номер будет отсчитываться с конца, начиная с номера -1. Срез с двумя параметрами: `S[a:b]` возвращает подстроку из `b-a` символов, начиная с символа с индексом `a`, то есть до символа с индексом `b`, не включая его.

6. Почему строки Python относятся к неизменяемому типу данных? Строки — один из типов данных, которые Python считает неизменяемыми, что означает невозможность их изменять. Python дает возможность изменять (заменять и перезаписывать) строки.

7. Как проверить то, что каждое слово в строке начинается с заглавной



буквы? `string.istitle()`

8. Как проверить строку на вхождение в неё другой строки? `string.find()`

9. Как найти индекс первого вхождения подстроки в строку? `s.partition()`

10. Как подсчитать количество символов в строке? `len(s)`

11. Как подсчитать то, сколько раз определённый символ встречается в строке? `s.count()`

12. Что такое f-строки и как ими пользоваться? Эти строки улучшают читаемость кода, а также работают быстрее чем другие способы форматирования. F-строки задаются с помощью литерала «f» перед кавычками. Пример: `print(f'Меня зовут {name} Мне {age} лет.')`

13. Как найти подстроку в заданной части строки? `s.find(значение, начало, конец)`

14. Как вставить содержимое переменной в строку, воспользовавшись методом `format()`? `print('{}'.format(s))`

15. Как узнать о том, что в строке содержатся только цифры? `s.isdigit()`

16. Как разделить строку по заданному символу? `str.split()`

17. Как проверить строку на то, что она составлена только из строчных букв? `s.isalpha()`

18. Как проверить то, что строка начинается со строчной буквы? `s.istitle()`

19. Можно ли в Python прибавить целое число к строке? Нет

20. Как «перевернуть» строку? `s.reverse()` 21. Как объединить список строк в одну строку, элементы которой разделены дефисами? `str.split('-')`

22. Как привести всю строку к верхнему или нижнему регистру? `s.upper()` `s.lower`

23. Как преобразовать первый символ строки к верхнему регистру? `s.capitalize()`

24. Как проверить строку на то, что она составлена только из прописных букв? `s.isupper()`

25. В какой ситуации вы воспользовались бы методом `splitlines()` ? `s.splitlines()` делит `s` на строки и возвращает их в списке. Любой из следующих

символов или последовательностей символов считается границей строки.

26. Как в заданной строке заменить на что-либо все вхождения некоей подстроки? `s.replace(old, new)`

27. Как проверить то, что строка начинается с заданной последовательности символов, или заканчивается заданной последовательностью символов? `str.startswith()` и `str.endswith()`

28. Как узнать о том, что строка включает в себя только пробелы? `s.isspace()`

29. Что случится, если умножить некую строку на 3? `Asd*3 = AsdAsdAsd`

30. Как привести к верхнему регистру первый символ каждого слова в строке? `s.title()`

31. Как пользоваться методом `partition()`? Метод `partition()` разбивает строку при первом появлении строки аргумента и возвращает кортеж, содержащий часть перед разделителем, строку аргумента и часть после разделителя.

32. В каких ситуациях пользуются методом `rfind()`? `s.rfind()` возвращает индекс последнего вхождения подстроки `<sub>`, который соответствует началу `<sub>`.