

**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 №6**  
Дисциплина: «Основы программной инженерии»  
Тема: «Условные операторы и циклы в языке Python»

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

Ставрополь 2022

Цель работы: приобретение навыков программирования разветвляющихся алгоритмов и алгоритмов циклической структуры. Освоить операторы языка Python версии 3.x if , while , for , break и continue , позволяющих реализовывать разветвляющиеся алгоритмы и алгоритмы циклической структуры.

## Выполнение работы

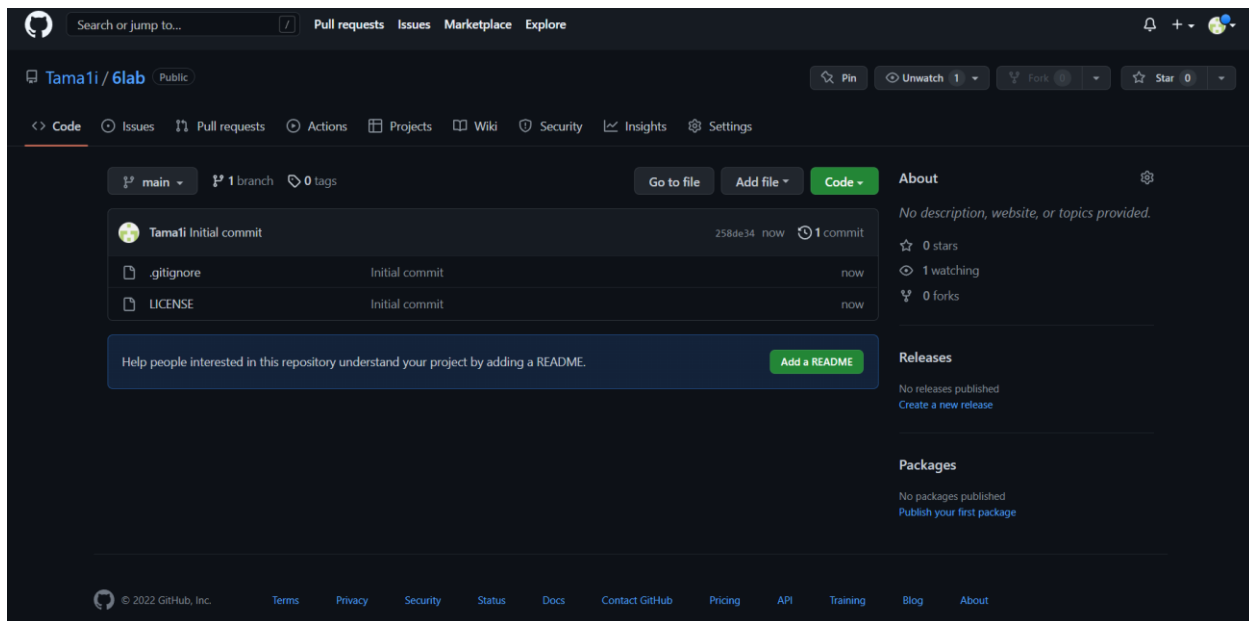


Рисунок 1 -создание репозитория

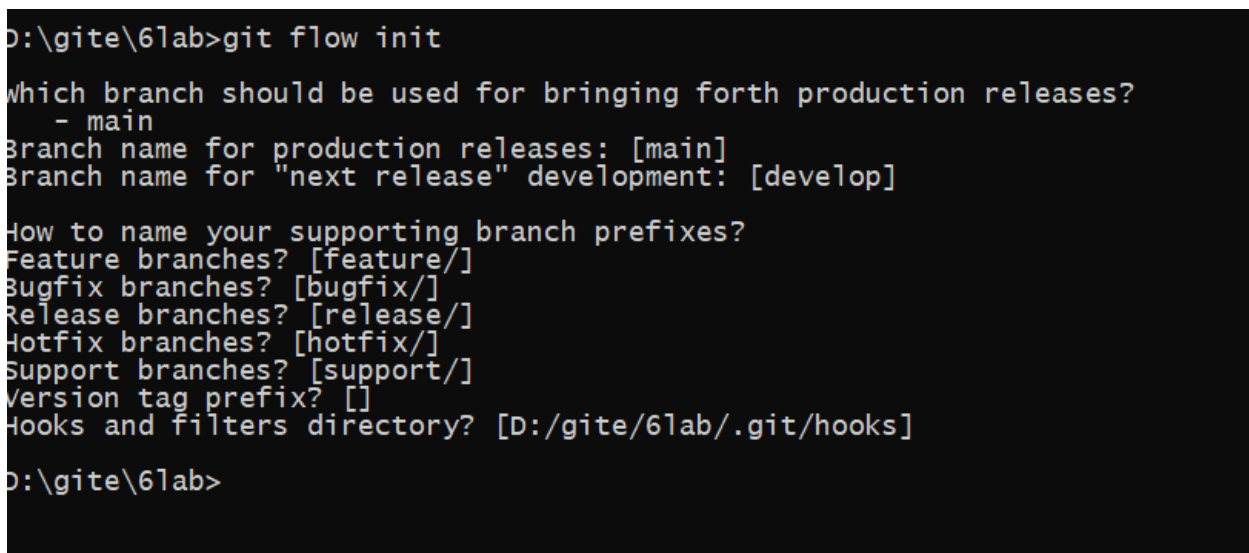


Рисунок 2 - Организация репозитория по модели ветвления git glow

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

if __name__ == '__main__':
    s = input("Введите предложение: ")
    r = s.replace(' ', '_')
    print("Предложение после замены: ", r)
```

Рисунок 3 – пример 1

```
C:\Users\iyaza\AppData\Local\Programs\Python\Python38\python.exe
Введите предложение: ar ra
Предложение после замены: ar_ra
Press any key to continue . . .
```

Рисунок 4 - результат работы программы

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

if __name__ == '__main__':
    word = input("Введите слово: ")

    idx = len(word) // 2
    if len(word) % 2 == 1:
        r = word[:idx] + word[idx+1:]
    else:
        r = word[:idx-1] + word[idx+1:]

    print(r)
```

Рисунок 5 – код программы пример 2

```
C:\Users\ryaza\AppData\Local\Programs\Python
Введите слово: gsjk
gk
Press any key to continue . . .
```

Рисунок 6 – результат работы программы

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

import sys

if __name__ == '__main__':
    s = input("Введите предложение: ")
    n = int(input("Введите длину: "))

    # Проверить требуемую длину.
    if len(s) >= n:
        print(
            "Заданная длина должна быть больше длины предложения",
            file=sys.stderr
        )
        exit(1)

    # Разделить предложение на слова.
    words = s.split(' ')
    # Проверить количество слов в предложении.
    if len(words) < 2:
        print(
            "Предложение должно содержать несколько слов",
            file=sys.stderr
        )
```

Рисунок 7 – код программы пример 3

```
C:\Users\ryaza\AppData\Local\Programs\Python\Python
Введите предложение: hel wor
Введите длину: 10
hel wor
Press any key to continue . . .
```

Рисунок 8 – результат работы программы

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

import sys
import re

if __name__ == '__main__':
    s = list(input("Напечатайте предложение: ").split())
    c = input("Первый символ: ")
    b = input("Второй символ: ")
    for i in range(len(s)):
        if(c in s[i]):
            print(s[i])
        elif(b in s[i]):
            print(s[i])
```

Рисунок 9 – код программы индивидуальное задание 1

```
C:\Users\ryaza\AppData\Local\Programs\Python\
Напечатайте предложение: help pls
Первый символ: h
Второй символ: s
help
pls
Press any key to continue . . .
```

Рисунок 10 – результат работы программы

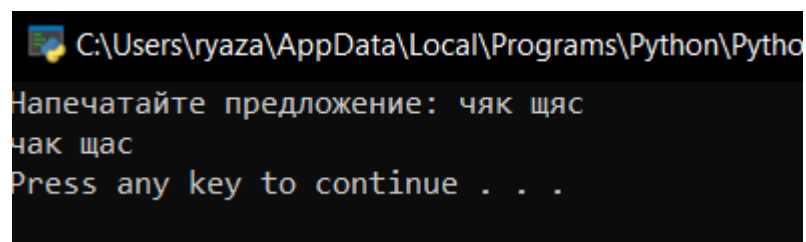
```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys
import re

if __name__ == '__main__':
    s = input("Напечатайте предложение: ")

    b = s.replace("чя", "ча")
    a = b.replace("щя", "ща")
    print(a)
```

Рисунок 11 – код программы идз 2



```
C:\Users\ryaza\AppData\Local\Programs\Python\Python...
Напечатайте предложение: чяк щяс
чак щас
Press any key to continue . . .
```

Рисунок 12 – результат работы программы

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

import sys
import re

if __name__ == '__main__':
    s = input("Напечатайте предложение: ")
    a = input("введите букву ")
    i = 0

    while s[i] != ".":
        if s[i] == "и":
            c = i
            i += 1
        o = s[:c] + a + s[c:]
        print(o)
```

Рисунок 13 – код программы идз 3

```
C:\Users\ryaza\AppData\Local\Programs\Python\Python310\python.exe
Напечатайте предложение: жили ежи а. ежилине жили
введите букву о
жили ежои а. ежилине жили
Press any key to continue . . .
```

Рисунок 14 – результат работы программы

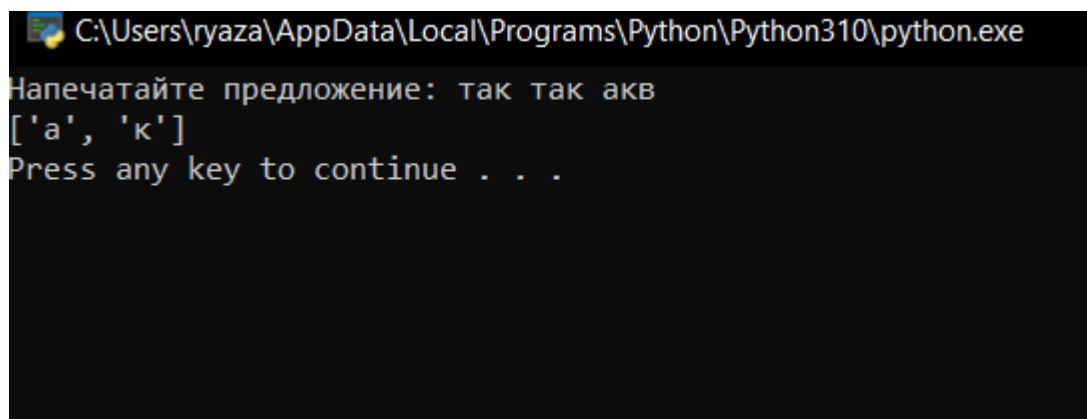


```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys
import re

if __name__ == '__main__':
    s1 = input("Напечатайте предложение: ")
    s = list(s1.split(" "))
    k = []
    for i in s1:
        if ((i in s[0]) and (i in s[1]) and (i in s[2])):
            if i not in k:
                k.append(i)
    print(k)
```

Рисунок 15 – код программы повышенной сложности



```
C:\Users\ryaza\AppData\Local\Programs\Python\Python310\python.exe
Напечатайте предложение: так так акв
['a', 'к']
Press any key to continue . . .
```

Рисунок 16 – 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>`.