

**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 №2.3**

Дисциплина: «Основы кроссплатформенного программирования»

Тема: «Работа со строками в языке Python»

Выполнил: студент 1 курса

группы ИВТ-б-о-21-1

Гошков Виталий Игоревич

Ставрополь 2022

## Выполнение работы:

1. Создал репозиторий в GitHub «rep 2.2» в который добавил .gitignore, который дополнил правила для работы с IDE PyCharm с ЯП Python, выбрал лицензию MIT, клонировал его на лок. сервер и организовал в соответствии с моделью ветвления git-flow.

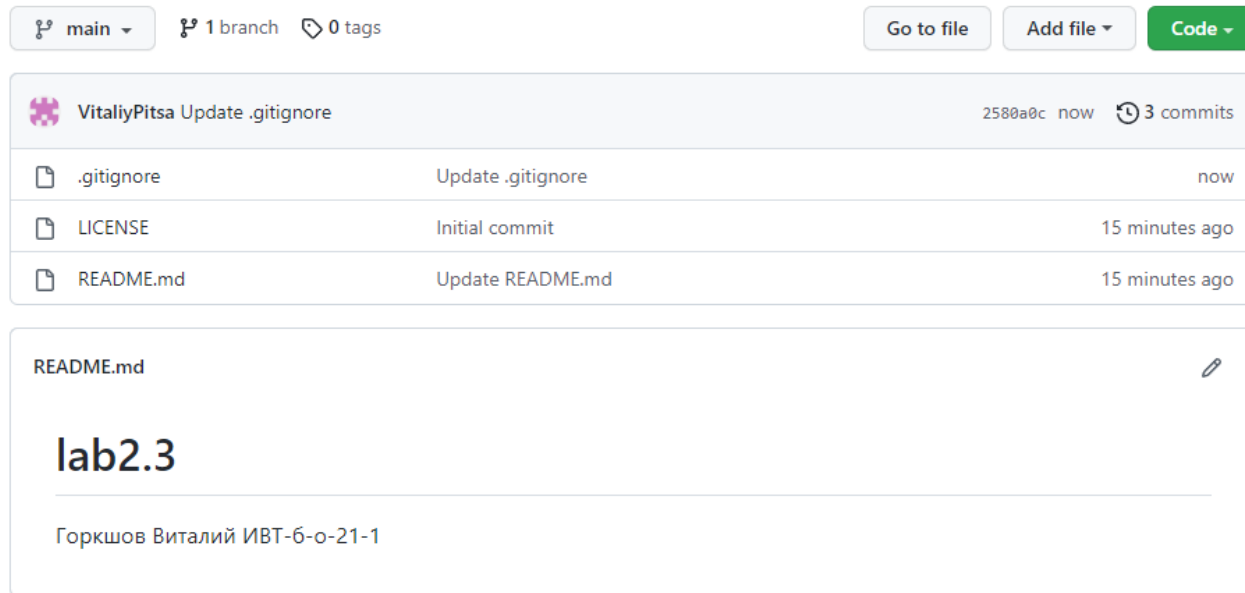


Рисунок 1 Создание репозитория

```
C:\Users\Виталий>cd /d C:\lab2.3

C:\lab2.3>git clone https://github.com/VitaliyPitsa/lab2.3.git
Cloning into 'lab2.3'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 11 (delta 2), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (11/11), 5.09 KiB | 5.09 MiB/s, done.
Resolving deltas: 100% (2/2), done.
```

Рисунок 2 Клонирование репозитория

```

C:\lab2.3\lab2.3>git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/lab2.3/lab2.3/.git/hooks]

```

Рисунок 3 Организация репозитория в соответствии с моделью ветвления  
git-flow

276 lines (220 sloc) | 5.57 KB

```

1  .idea/
2  # Created by https://www.toptal.com/developers/gitignore/api/python,pycharm
3  # Edit at https://www.toptal.com/developers/gitignore?templates=python,pycharm
4
5  ### PyCharm ###
6  # Covers JetBrains IDEs: IntelliJ, RubyMine, PhpStorm, AppCode, PyCharm, CLion, Android Studio, WebStorm and Rider
7  # Reference: https://intellij-support.jetbrains.com/hc/en-us/articles/206544839
8
9  # User-specific stuff
10 .idea/**/workspace.xml
11 .idea/**/tasks.xml
12 .idea/**/usage.statistics.xml
13 .idea/**/dictionaries
14 .idea/**/shelf
15
16 # AWS User-specific
17 .idea/**/aws.xml
18
19 # Generated files
20 .idea/**/contentModel.xml
21
22 # Sensitive or high-churn files
23 .idea/**/dataSources/
24 .idea/**/dataSources.ids
25 .idea/**/dataSources.local.xml

```

Рисунок 4 Изменение .gitignore

2. Создал проект PyCharm в папке репозитория, проработал примеры ЛР.

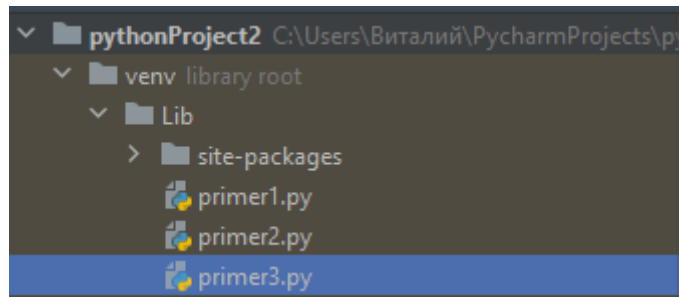


Рисунок 5 Создание проекта в PyCharm

```
Введите предложение: asdf asdf asdfnjk   asd s sasdf
Предложение после замены: asdf_asdf_asdfnjk___asd_s_sasdf
```

Рисунок 6 Рез-т выполнения программы

```
Введите слово: gashkdf
gaskdf
```

Рисунок 7 Рез-т выполнения программы

```
Введите предложение: qwe qweq qqw   er w a asdfda rt
Введите длину: 39
qwe   qweq  qqw   er  w  a  asdfda  rt
```

Рисунок 8 Рез-т выполнения программы

3. (22 вариант). Выполнил 3 индивидуальных задания и задание повышенной сложности.

**Задание 1.** Дано предложение. Составить программу, которая печатает «столбиком» все вхождения в предложение некоторого символа.

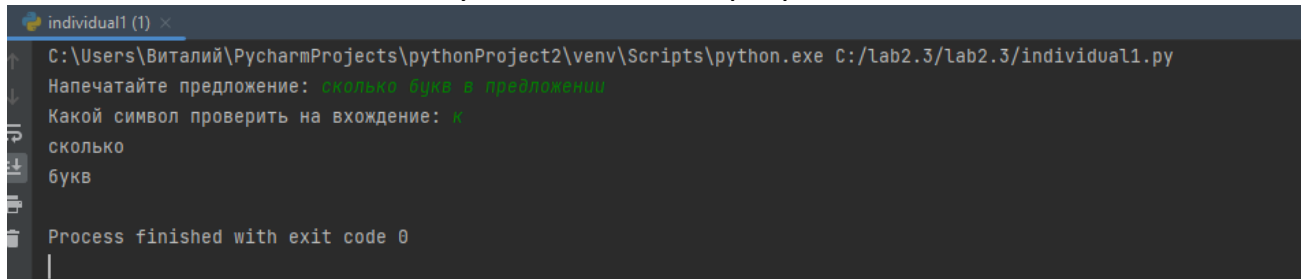
```

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

> if __name__ == '__main__':
    s = list(input('Напечатайте предложение: ').split())
    c = input('Какой символ проверить на вхождение: ')
    for i in range(len(s)):
        if (c in s[i]): print(s[i])

```

Рисунок 9 Листинг программы



```

individual1 (1) x
C:\Users\Виталий\PycharmProjects\pythonProject2\venv\Scripts\python.exe C:/lab2.3/lab2.3/individual1.py
Напечатайте предложение: сколько букв в предложении
Какой символ проверить на вхождение: к
сколько
букв
Process finished with exit code 0

```

Рисунок 10 Выполнение программы

**Задание 2.** Дано предложение. Определить порядковые номера первой пары одинаковых соседних символов. Если таких символов нет, то должно быть напечатано соответствующее сообщение.

```

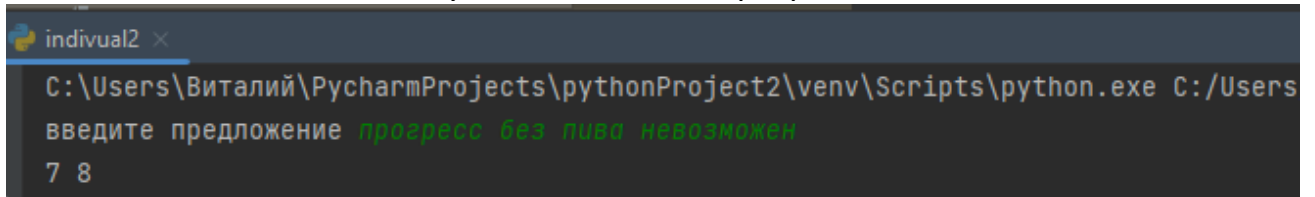
> #!/usr/bin/env python3
# -*- coding: utf-8 -*-

> if __name__ == '__main__':
    line = input("введите предложение ")

    for ind in range(len(line) - 1):
        if line[ind] == line[ind + 1]:
            print(ind + 1, ind + 2)
            break
    else:
        print('Таких нет!!!')

```

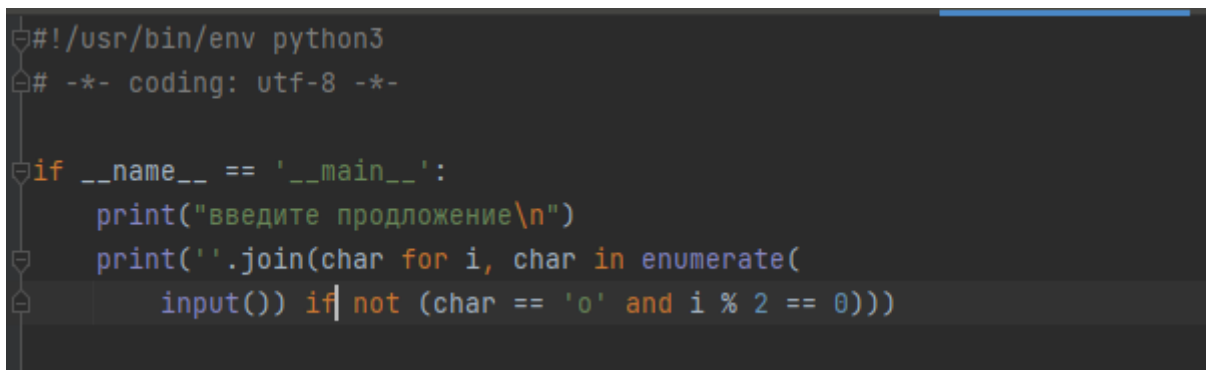
Рисунок 11 Листинг программы



```
individual2 x
C:\Users\Виталий\PycharmProjects\pythonProject2\venv\Scripts\python.exe C:/Users
введите предложение прогресс без пива невозможен
7 8
```

Рисунок 12 Выполнение программы

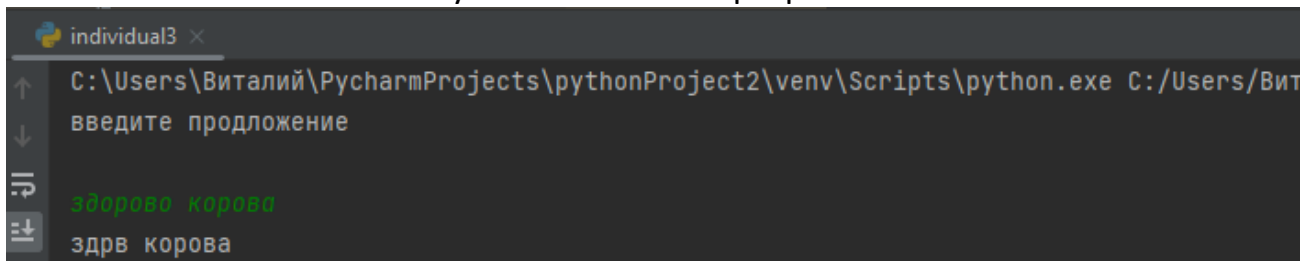
**Задание 3.** Дано предложение. Удалить из него все буквы о, стоящие на нечетных местах.



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

if __name__ == '__main__':
    print("введите предложение\n")
    print(''.join(char for i, char in enumerate(
        input()) if not (char == 'o' and i % 2 == 0)))
```

Рисунок 13 Листинг программы



```
individual3 x
C:\Users\Виталий\PycharmProjects\pythonProject2\venv\Scripts\python.exe C:/Users/Вит
введите предложение
здорово корова
здрв корова
```

Рисунок 14 Выполнение программы

**Усложненное задание.** Даны два слова. Напечатать только те буквы слов, которые встречаются в обоих словах лишь один раз. Например, если заданные слова процессор и информация, то ответом должно быть: п е ф м а я.

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

if __name__ == '__main__':
    s1 = input("введите 1 слово ")
    s2 = input("введите 2 слово ")
    s = s1 + s2
    ls = []
    for c in s:
        if s.count(c) == 1:
            ls.append(c)
    print(ls)
```

Рисунок 16 Листинг программы

```
C:\Users\Виталий\PycharmProjects\pythonProject2\venv\Scripts\python.exe C:/Users/Виталий/PycharmProjects/pythonProject2/venv/Lib/uslojn8.py
введите 1 слово процессор
введите 2 слово информация
['п', 'е', 'н', 'ф', 'м', 'а', 'я']
Process finished with exit code 0
cd
```

Рисунок 17 Выполнение программы

4. Сделал коммит, выполнил слияние с веткой main, и запустил изменения в уд. репозиторий.

```

C:\lab2.3\lab2.3> git checkout main
Already on 'main'
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

C:\lab2.3\lab2.3>git merge develop
Already up to date.

C:\lab2.3\lab2.3>git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 2.45 KiB | 1.22 MiB/s, done.
Total 9 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/VitaliyPitsa/lab2.3.git
   2580a0c..936f6c6  main -> main

```

Рисунок 18 Пуш коммитов

main

1 branch

0 tags

Go to file

Add file

Code

VitaliyPitsa formatted according to pep8

936f6c6 1 minute ago 4 commits

.gitignore	Update .gitignore	22 hours ago
LICENSE	Initial commit	22 hours ago
README.md	Update README.md	22 hours ago
individual1.py	formatted according to pep8	1 minute ago
individual2.py	formatted according to pep8	1 minute ago
individual3.py	formatted according to pep8	1 minute ago
pimer1.py	formatted according to pep8	1 minute ago
primer2.py	formatted according to pep8	1 minute ago
primer3.py	formatted according to pep8	1 minute ago
uslojn8.py	formatted according to pep8	1 minute ago

Рисунок 19 Изменения на уд. сервере



## **Контр. вопросы и ответы на них:**

### **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  $S = \text{'Hello'}$ , то  $S[0] == \text{'H'}$ ,  $S[1] == \text{'e'}$ ,  $S[2] == \text{'l'}$ ,  $S[3] == \text{'l'}$ ,  $S[4] == \text{'o'}$ .

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  $b$ -а 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(<sep>)
```

## **10. Как подсчитать количество символов в строке?**

```
len(s)
```

## **11. Как подсчитать то, сколько раз определённый символ встречается в строке?**

```
s.count(<sub>)
```

## **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(<sub>)` возвращает индекс последнего вхождения подстроки `<sub>` в `s`, который соответствует началу `<sub>`.