

**Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Институт цифрового развития  
Кафедра инфокоммуникаций**

**ОТЧЕТ  
ПО ПРАКТИЧЕСКОЙ РАБОТЕ №2.2  
дисциплины  
«Основы кроссплатформенного программирования»**

Выполнила студентка группы  
ИВТ-б-о-21-1  
Яковлева Е.А. « » \_\_\_\_\_ 20\_\_ г.  
Подпись студента \_\_\_\_\_  
Работа защищена  
« » \_\_\_\_\_ 20\_\_ г.

Проверил доцент  
Кафедры инфокоммуникаций,  
старший преподаватель  
Воронкин Р.А.

---

(подпись)

Ставрополь, 2022 г.

**Тема:** Работа со строками в языке Python

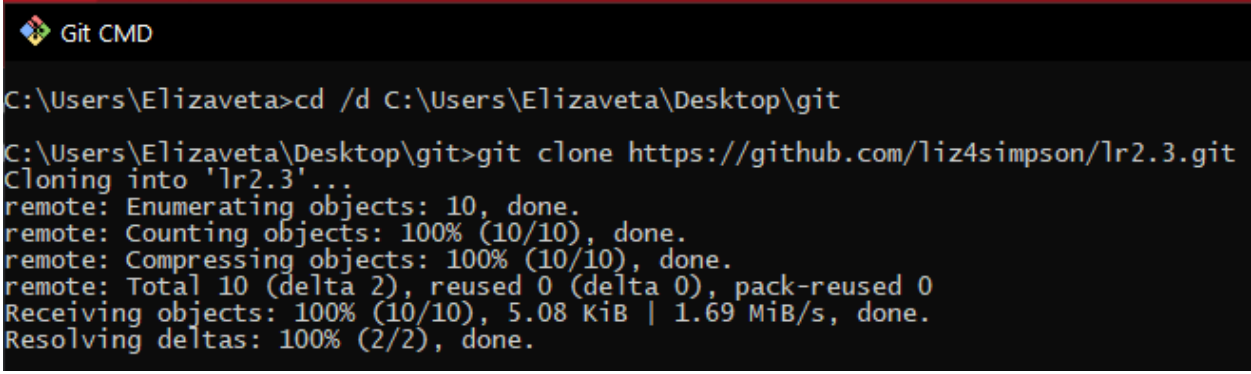
**Цель:** приобретение навыков по работе со строками при написании программ с помощью языка программирования Python версии 3.x.

**Порядок выполнения работы:**

Создала общедоступный репозиторий на GitHub, в котором использованы лицензия MIT и язык программирования Python.

Рисунок 1. Создание репозитория

Выполнила клонирование созданного репозитория.



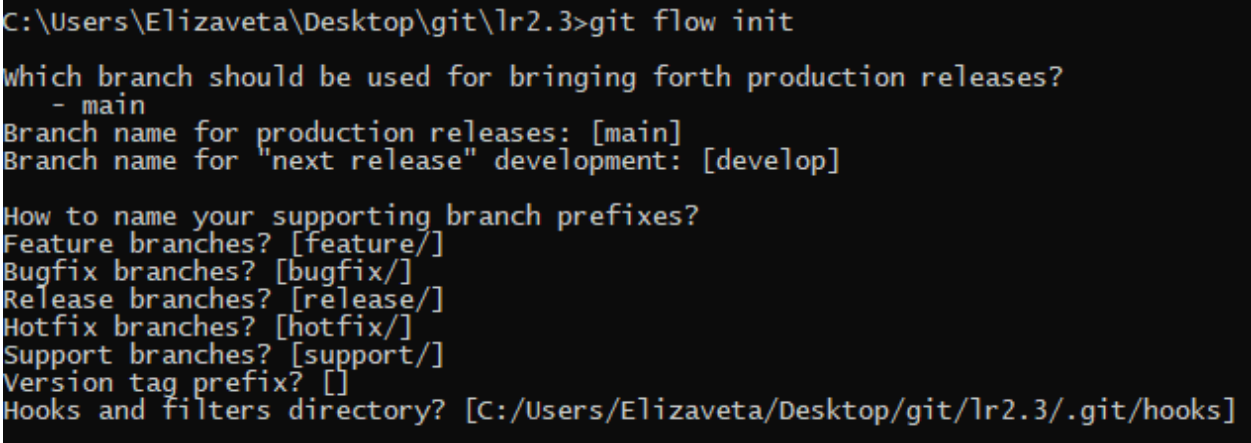
```
Git CMD

C:\Users\Elizaveta>cd /d C:\Users\Elizaveta\Desktop\git

C:\Users\Elizaveta\Desktop\git>git clone https://github.com/liz4simpson/lr2.3.git
Cloning into 'lr2.3'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 10 (delta 2), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (10/10), 5.08 KiB | 1.69 MiB/s, done.
Resolving deltas: 100% (2/2), done.
```

Рисунок 2. Клонирование репозитория

Организовала свой репозиторий в соответствии с моделью ветвления git-flow.



```
C:\Users\Elizaveta\Desktop\git\lr2.3>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:/Users/Elizaveta/Desktop/git/lr2.3/.git/hooks]
```

Рисунок 3. Организация репозитория согласно модели ветвления git-flow

2. Создала проект PyCharm в папке репозитория, проработала примеры из лабораторной работы.

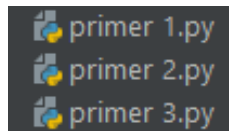


Рисунок 4. Проработанные примеры

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

Рисунок 5. Результат выполнения программы

```
Введите слово: gashkdf
gaskdf
```

Рисунок 6. Результат выполнения программы

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

Рисунок 7. Результат выполнения программы

3. Выполнила 3 индивидуальных задания и задание повышенной сложности. (в25)

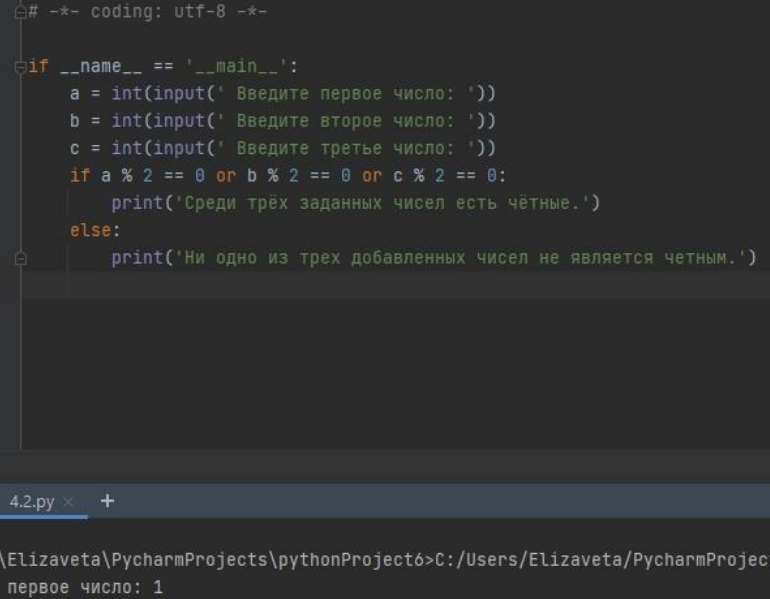
A screenshot of a code editor window titled 'лаба2.3'. It shows a Python file 'zadanie1.py' with the following code:

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == '__main__':
5      p = input("Введите предложение: ")
6      sl_2 = input("Введите букосочетание из двух букв: ")
7      sl_n = input("Введите букосочетание: ")
8
9      print('Количество букосочетания ро в предложении = ', p.count('po'))
10     print('Количество букосочетания ', sl_2, ' в предложении = ', p.count(sl_2))
11     print('Количество букосочетания ', sl_n, ' в предложении = ', p.count(sl_n))
12
```

The terminal at the bottom shows the execution results:

```
y
Введите предложение: шла саша по шоссе
Введите букосочетание из двух букв: шо
Введите букосочетание: са шл
Количество букосочетания ро в предложении = 0
Количество букосочетания шо в предложении = 1
Количество букосочетания са шл в предложении = 0
C:\Users\Elizaveta\Desktop\лаба2.3>
```

Рисунок 8. Индивидуальное задание №1



The screenshot displays the PyCharm IDE interface. The top pane shows a Python script named `4.2.py` with the following code:

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == '__main__':
5      a = int(input(' Введите первое число: '))
6      b = int(input(' Введите второе число: '))
7      c = int(input(' Введите третье число: '))
8      if a % 2 == 0 or b % 2 == 0 or c % 2 == 0:
9          print('Среди трёх заданных чисел есть чётные.')
10     else:
11         print('Ни одно из трех добавленных чисел не является четным.')
12
```

The bottom pane shows the terminal output for the script:

```
Terminal: 4.2.py x +
C:\Users\Elizaveta\PycharmProjects\pythonProject6>C:/Users/Elizaveta/PycharmProjects/pythonProject6\pythonProject6\4.2.py
Введите первое число: 1
Введите второе число: 3
Введите третье число: 2
Среди трёх заданных чисел есть чётные.
```

Рисунок 9. Индивидуальное задание №2

```
1  #!/usr/bin/env python3
2  #- coding: utf-8 -*-
3
4
5  f = ''
6  if __name__ == '__main__':
7      s = "Python"
8      f += s[-1] + s[-2] + s[-3] + s[-4] + s[-5] + s[-6]
9      print(f)
10
```

zadanie2 x

"C:\Program Files\Python37\python.exe" C:/Users/Elizaveta/Desktop/na6a2.3  
nohtyP

Process finished with exit code 0

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 if __name__ == '__main__':
4
5     # Условие задачи: Дано слово из 12 букв. Переставить его буквы сле.
6     # вторая, одиннадцатая, ..., пятая, восьмая, шестая, седьмая.
7
8
9     # присваиваем переменной "a" слово из 12 букв
10    a = "анестезиолог"
11    # Вывод перестановка и вывод букв
12    print(a[0]+a[11]+a[1]+a[10]+a[4:7:1]+a[4]+a[7]+a[5]+a[6])
13
14
15 if __name__ == '__main__'
```

Terminal: uslojnennoe.py × uslojnennoe.py × zadanie1.py × zadanie3.py ×

Microsoft Windows [Version 10.0.19044.1889]  
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\Elizaveta\Desktop\лаба2.3>C:/Users/Elizaveta/Desktop/лаба2.3/zadanie3.p  
y  
агнотезтиез

Рисунок 11. Индивидуальное задание №3

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 if __name__ == '__main__':
6     s = input('Введите предложение:\n')
7     s1 = s.split()
8     i = 0
9     s1.sort(key=len, reverse=True)
10    s = ''
11    for i in range(0, len(s1)):
12        s += s1[i] + ' '
13    print(s)
14
15 if __name__ == '__main__'
```

Terminal: uslojnennoe.py × zadanie1.py × zadanie3.py × uslojnennoe.py ×

(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\Elizaveta\Desktop\лаба2.3>C:/Users/Elizaveta/Desktop/лаба2.3/uslojnenno  
e.py  
Введите предложение:  
шла саша по шоссе и сосала сушку  
сосала шоссе сушку саша шла по и

Рисунок 13. Индивидуальное задание повышенной сложности

4. Сделала коммит, выполнила слияние с веткой main, и запустила изменения в удаленный репозиторий.

```
C:\Users\Elizaveta\Desktop\git\lr2.2>git add .  
  
C:\Users\Elizaveta\Desktop\git\lr2.3>git commit -m "added project"  
[develop c62f936] added project  
7 files changed, 122 insertions(+)
```

Рисунок 15. Фиксация и коммит файлов

```
C:\Users\Elizaveta\Desktop\git\lr2.3>git checkout main  
Switched to branch 'main'  
Your branch is up to date with 'origin/main'.  
  
C:\Users\Elizaveta\Desktop\git\lr2.3>git merge develop  
Updating 325587e..650795b  
Fast-forward  
uslojnennoe.py | 13 +++++  
.../zadanie1.py | 11 +++++  
.../zadanie2.py | 9 +++++  
.../zadanie3.py | 12 +++++  
.../primer1.py | 7 +++  
.../primer2.py | 13 +++++  
.../primer3.py | 57 +++++  
7 files changed, 122 insertions(+)
```

Рисунок 16. Слияние ветки develop с main

```
C:\Users\Elizaveta\Desktop\git\lr2.3>git push  
Enumerating objects: 12, done.  
Counting objects: 100% (12/12), done.  
Delta compression using up to 4 threads  
Compressing objects: 100% (11/11), done.  
Writing objects: 100% (11/11), 2.66 KiB | 453.00 KiB/s, done.  
Total 11 (delta 1), reused 0 (delta 0), pack-reused 0  
remote: Resolving deltas: 100% (1/1), done.  
To https://github.com/liz4simpson/lr2.3.git  
325587e..650795b main -> main
```

Рисунок 17. Отправка изменений на удаленный репозиторий

The screenshot shows a GitHub repository page for 'liz4simpson add project'. The repository has 4 commits. The file list includes 'Задания', 'Примеры', '.gitignore', 'LICENSE', 'README.md', and 'uslojnennoe.py'. The 'uslojnennoe.py' file has a link to 'add project'. The README.md file is visible at the bottom with the text 'Лабораторная работа №2.3 Яковлева Елизавета ИВТ-6-о-21-1'.

Рисунок 18. Изменения на удаленном репозитории

**Вывод:** В ходе лабораторной работы были приобретены навыки по работе со строками при написании программ с помощью языка программирования Python версии 3.x.

### **Ответы на 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`-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>`