

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

Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №6
дисциплины
«Основы кроссплатформенного программирования»

Выполнил:
Пугачев Кирилл Дмитриевич
2 курс, группа ИТС-б-о-23-1,
11.03.02«Инфокоммуникационные
технологии и системы связи», очная
форма обучения

(подпись)

Проверил:
Доцент департамента цифровых,
робототехнических систем и
электроники Воронкин Р.А.

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 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:

1. Изучил теоретический материал.
2. Приступил к выполнению заданий.
3. Создание репозитория.

Ссылка на **GitHub** <https://github.com/chillkirill/LABA-6.git>

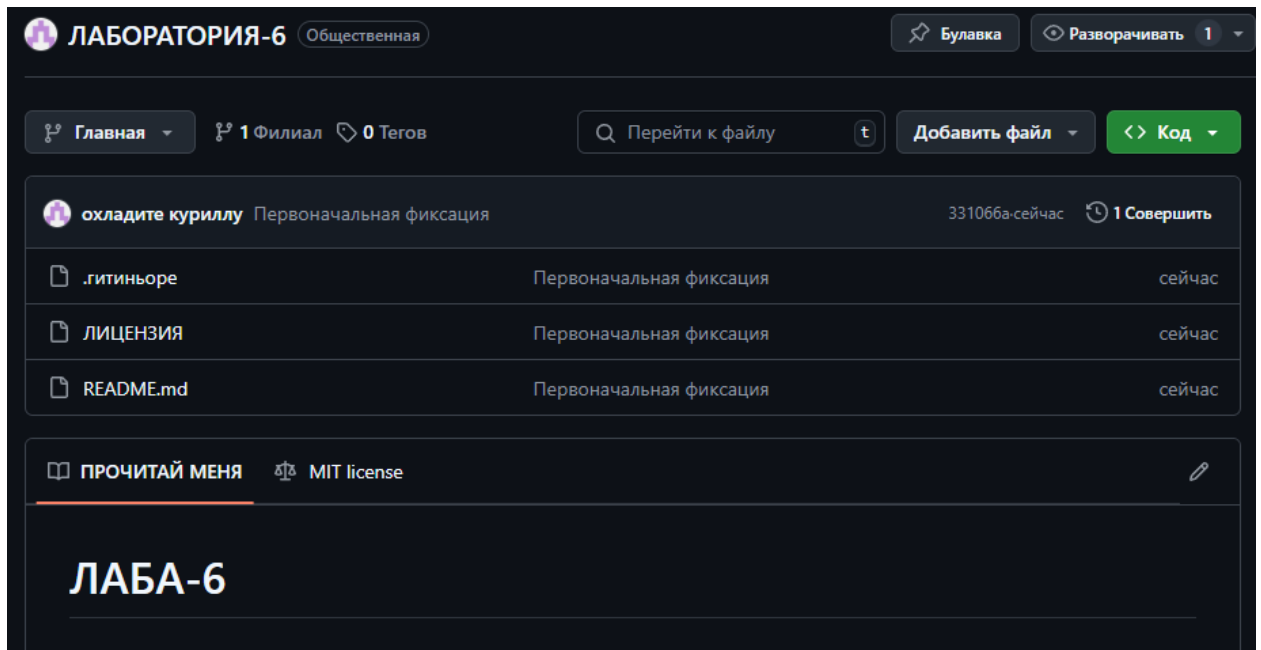


Рисунок 1. Репозиторий

4. Клонирование репозитория.

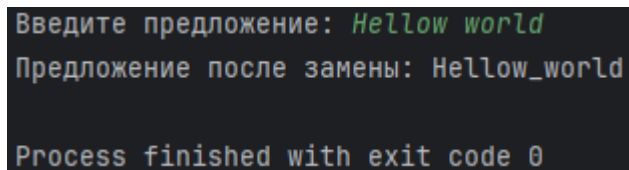
```
C:\Users\user>git clone https://github.com/chillkirill/LABA-6.git
Cloning into 'LABA-6'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (5/5), done.
```

Рисунок 2. Клонирование

5. Приступил к выполнению примеров.

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

Рисунок 3. Код к 1 примеру



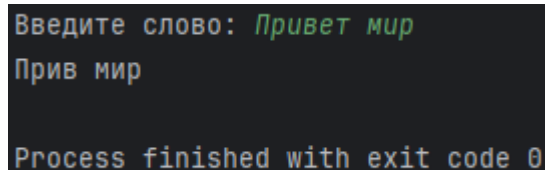
Введите предложение: *Hello world*
Предложение после замены: Hello_world

Process finished with exit code 0

Рисунок 4. Пример 1.

```
if __name__ == '__main__':  
    word = input("Введите слово: ")  
    idx = len(word) // 2  
    if len(word) % 2 == 1:  
        # Длина слова нечетная.  
        r = word[:idx] + word[idx+1:]  
    else:  
        # Длина слова четная.  
        r = word[:idx-1] + word[idx+1:]  
    print(r)
```

Рисунок 5. Код к 2 примеру



Введите слово: *Привет мир*
Прив мир

Process finished with exit code 0

Рисунок 6. Пример 2.

```

#!/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)
        exit(1)

    # Количество пробелов для добавления.
    delta = n - len(s)

    for word in words:
        delta -= len(word)

    # Количество пробелов на каждое слово.
    w, r = divmod(delta, len(words) - 1)

    # Сформировать список для хранения слов и пробелов.
    lst = []

    # Пронумеровать все слова в списке и перебрать их.
    for i, word in enumerate(words):
        lst.append(word)

        # Если слово не является последним, добавить пробелы.
        if i < len(words) - 1:
            # Определить количество пробелов.
            width = w

            if r > 0:
                width += 1

```

Рисунок 7. Код к 3 примеру

```

Введите предложение: Hello world
Введите длину: 13
Helloworld

Process finished with exit code 0

```

Рисунок 8. Пример 3.

6. Зафиксировал изменения в репозитории.

```
C:\Users\user\LABA-6>git add .
C:\Users\user\LABA-6>git commit -m "Добавление примеров"
[main f9e4ae9] Добавление примеров
3 files changed, 66 insertions(+)
create mode 100644 primer1.py.txt
create mode 100644 primer2.py.txt
create mode 100644 primer3.py.txt
C:\Users\user\LABA-6>git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.50 KiB | 1.50 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/chillkirill/LABA-6.git
331066a..f9e4ae9 main -> main
```

Рисунок 9. Изменения

7. Привел скриншоты результатов каждой из программы.

8. Приступил к выполнению индивидуального задания.

```
def calculate_percentage_of_a(sentence):
    total_letters = sum(1 for char in sentence if char.isalpha())
    count_a = sentence.lower().count('a') # Количество букв "a"

    return (count_a / total_letters) * 100 if total_letters > 0 else 0

sentence = input("Введите предложение: ")

result = calculate_percentage_of_a(sentence)
print(f"Доля букв 'a' в предложении: {result:.2f}%")
```

Рисунок 10. Код к 1 заданию

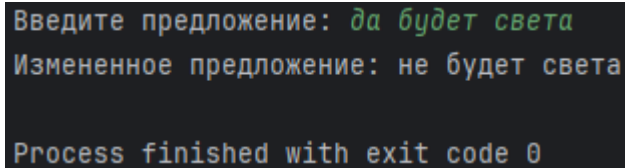
```
Введите предложение: Антон Андреевич
Доля букв 'a' в предложении: 14.29%

Process finished with exit code 0
```

Рисунок 11. Задание 1

```
sentence = input("Введите предложение: ")  
modified_sentence = sentence.replace('да', 'не').replace('Да', 'Не')  
print("Измененное предложение:", modified_sentence)
```

Рисунок 12. Код к 2 заданию

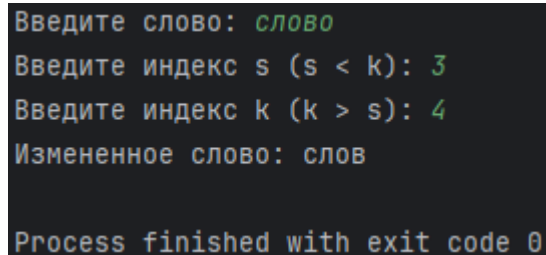


```
Введите предложение: да будет света  
Измененное предложение: не будет света  
Process finished with exit code 0
```

Рисунок 13. Задание 2

```
word = input("Введите слово: ")  
  
s = int(input("Введите индекс s (s < k): "))  
k = int(input("Введите индекс k (k > s): "))  
  
modified_word = word[:s] + word[s+1:k] + word[s] + word[k+1:]  
  
print("Измененное слово:", modified_word)
```

Рисунок 14. Код к 3 заданию



```
Введите слово: слово  
Введите индекс s (s < k): 3  
Введите индекс k (k > s): 4  
Измененное слово: слов  
Process finished with exit code 0
```

Рисунок 15. Задание 3

9. Зафиксировал изменения в репозитории.

```
C:\Users\user\LABA-6>git add .  
C:\Users\user\LABA-6>git commit -m "Добавление индивидуальных заданий"  
[main 0489021] Добавление индивидуальных заданий  
3 files changed, 23 insertions(+)  
create mode 100644 individual1.py.txt  
create mode 100644 individual2.py.txt  
create mode 100644 individual3.py.txt  
C:\Users\user\LABA-6>git push  
Enumerating objects: 6, done.  
Counting objects: 100% (6/6), done.  
Delta compression using up to 8 threads  
Compressing objects: 100% (5/5), done.  
Writing objects: 100% (5/5), 979 bytes | 979.00 KiB/s, done.  
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)  
remote: Resolving deltas: 100% (1/1), completed with 1 local object.  
To https://github.com/chillkirill/LABA-6.git  
f9e4ae9..0489021 main -> main
```

Рисунок 16. Изменения

Ответы на контрольные вопросы:

1. Что такое строки в языке Python?

Строки в Python — строки в языке Python это последовательности символов, используемые для хранения текстовой информации. Они могут содержать буквы, цифры, пробелы и специальные символы.

2. Какие существуют способы задания строковых литералов в языке Python?

Строки можно задавать с помощью одинарных, двойных, тройных одинарных и тройных двойных кавычек.

3. Какие операции и функции существуют для строк?

Операции включают конкатенацию, повторение, а также функции, такие как `upper`, `lower`, `replace`, `split`.

4. Как осуществляется индексирование строк?

Индексирование строк осуществляется с помощью квадратных скобок. Индексы начинаются с 0 для первого символа, -1 для последнего и тд.

5. Как осуществляется работа со срезами для строк?

Срезы позволяют извлекать подстроки. Синтаксис: `s` - start - начальный индекс, а `end` - конечный индекс.

6. Почему строки Python относятся к неизменяемому типу данных?

Строки являются неизменяемыми, потому что после создания их содержимое нельзя изменить. Любые операции, которые изменяют строку, создают новую строку.

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

`istitle`, `s.istitle`.

8. Как проверить строку на вхождение в неё другой строки?

Оператор `in`: `substring in s`.

9. Как найти индекс первого вхождения подстроки в строку?

Через `find`: `s.find`.

10. Как подсчитать количество символов в строке?

Через функцию len: len(s).

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

Через метод count.

12. Что такое f-строки и как ими пользоваться?

Форматированные строки позволяют вставлять выражения в строку.

13. Как найти подстроку в заданной части строки?

Через метод find(start, end)

14. Как вставить содержимое переменной в строку, воспользовавшись методом format?

Через format: "Пример, {}".format(name).

15. Как узнать о том, что в строке содержатся только цифры?

Через isdigit

16. Как разделить строку по заданному символу?

Через метод split(,).

17. Как проверить строку на то, что она составлена только из строчных букв?

Через метод islower: s.islower.

18. Как проверить то, что строка начинается со строчной буквы?

Через islower для первого символа.

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

Нет. Строки и числа нельзя складывать напрямую.

20. Как «перевернуть» строку?

Через срез

21. Как объединить список строк в одну строку, элементы которой разделены дефисами?

через join

22. Как привести всю строку к верхнему или нижнему регистру?

через методы upper и lower

23. Как преобразовать первый и последний символы строки к верхнему регистру?

Через upper

24. Как проверить строку на то, что она составлена только из прописных букв?

Через метод isupper

25. В какой ситуации вы воспользовались бы методом splitlines()?

Метод splitlines() полезен для разделения строки на строки по символам новой строки.

26. Как в заданной строке заменить на что-либо все вхождения некоей подстроки?

Через метод replace.

27. Как проверить то, что строка начинается с заданной последовательности символов, или заканчивается заданной последовательностью символов?

Через методы startswith() и endswith()

28. Как узнать о том, что строка включает в себя только пробелы?

Используйте метод isspace: s.isspace.

29. Что случится, если умножить некую строку на 3?

Строка будет повторена трижды.

30. Как привести к верхнему регистру первый символ каждого слова в строке?

Через метод title: s.title.

31. Как пользоваться методом partition()?

Метод partition() разделяет строку на три части: до, разделитель и после.

32. В каких ситуациях пользуются методом rfind()?

Метод rfind() используется для поиска последнего вхождения подстроки в строке

Вывод: в ходе работы исследовал базовые возможности системы контроля версий Git для работы с локальными репозиториями.