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 ЛАБОРАТОРНОЙ РАБОТЕ № 2.3**

**дисциплины «Программирование на Python»**

**Вариант 15**

|  |
| --- |
| Выполнил:  Кенесбаев Хилол Куат улы  2 курс, группа ИВТ-б-о-22-1,  09.03.01 «Информатика и вычислительная техника», направленность (профиль)  «Программное обеспечение средств вычислительной техники и автоматизированных систем», очная форма обучения  (подпись) |
| Руководитель практики: Воронкин Р.А., доцент кафедры инфокоммуникаций  (подпись) |

Отчет защищен с оценкой Дата защиты

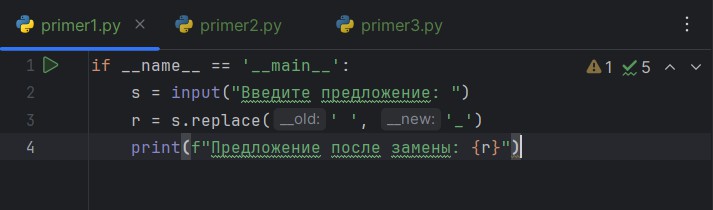
Ставрополь, 2023 г.

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

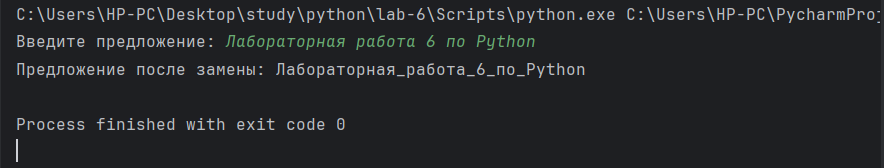
**Цель:** приобретение навыков по работе со строками при написании программ с помощью языка программирования Python версии 3.х.

**Порядок выполнения работы:**

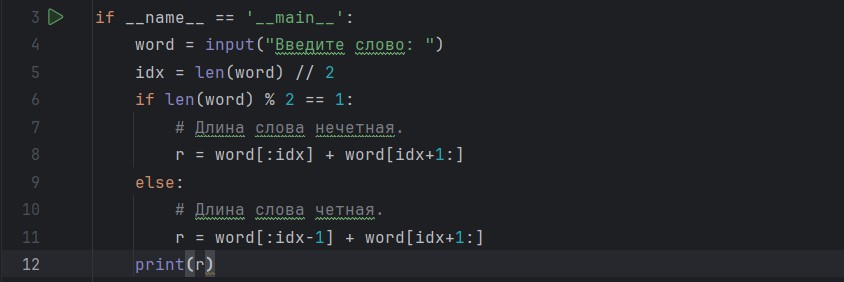
1. **Проработал примеры лабораторной работы:**



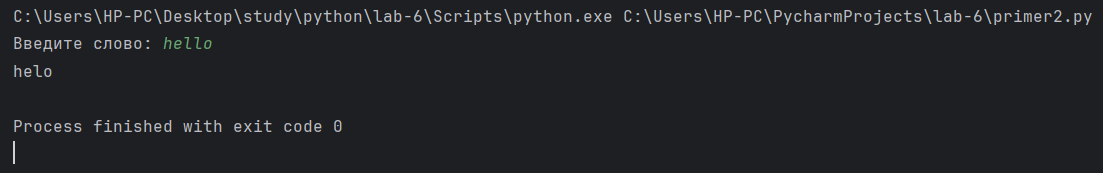
*Рисунок 1 primer1.py*



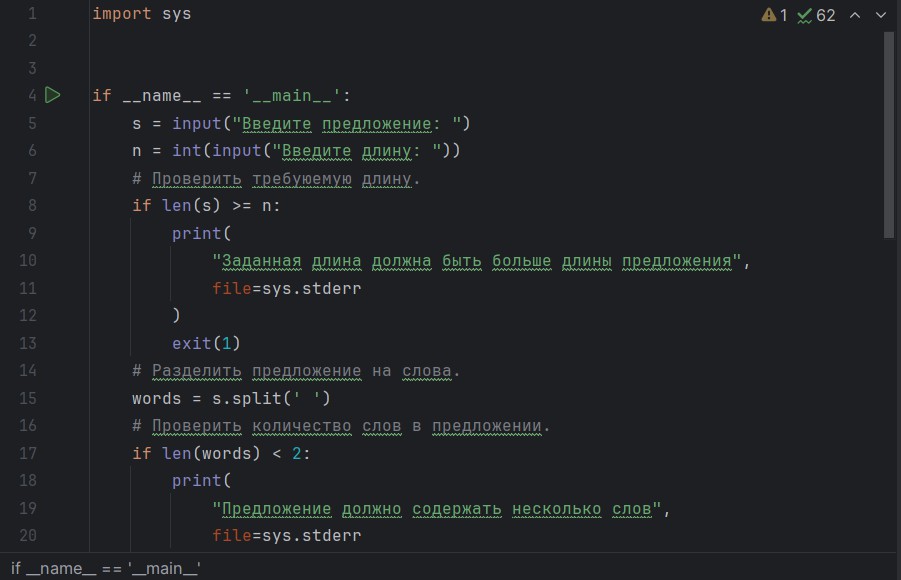
*Рисунок 2 Запуск программы primer1.py*



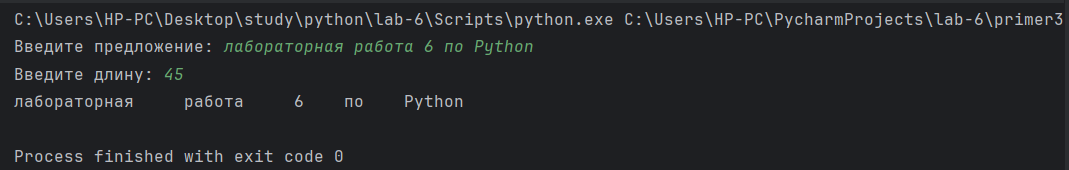
*Рисунок 3 primer2.py*



*Рисунок 4 Запуск программы primer2.py*



*Рисунок 5 primer3.py*



*Рисунок 6 Запуск программы primer3.py*

1. **Выполнил первое индивидуальное задание:** Дано предложение. Определить число букв о в нем. **Код программы:**

def count\_letter\_o(sentence):

count = 0

for letter in sentence:

if letter.lower() == 'о': # Проверяем каждую букву, игнорируя регистр

count += 1

return count

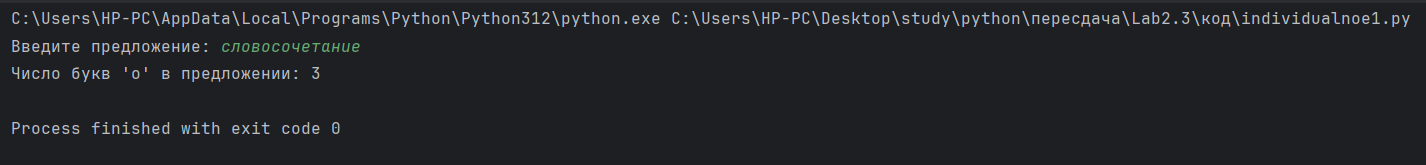
# Ввод предложения от пользователя

sentence = input("Введите предложение: ")

# Вычисление числа букв 'о' в предложении

letter\_count = count\_letter\_o(sentence)

print(f"Число букв 'о' в предложении: {letter\_count}")



*Рисунок 7 Результат работы программы individualnoe1.py*

1. **Выполнил второе индивидуальное задание:**

Дана предложение. Все его символы, стоящие на четных местах, заменить буквой ы.

**Код программы:**

def replace\_even\_chars(sentence):

modified\_sentence = ""

for i in range(len(sentence)):

if i % 2 == 1: # Проверяем индекс символа (четный индекс)

modified\_sentence += 'ы'

else:

modified\_sentence += sentence[i]

return modified\_sentence

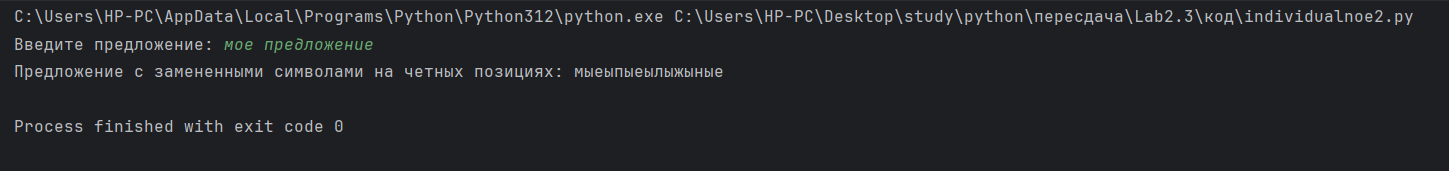
# Ввод предложения от пользователя

sentence = input("Введите предложение: ")

# Замена символов на четных позициях на 'ы'

result\_sentence = replace\_even\_chars(sentence)

print(f"Предложение с замененными символами на четных позициях: {result\_sentence}")



*Рисунок 8 Результат работы программы individualnoe2.py*

1. **Выполнил третье индивидуальное задание:**

Дано ошибочно написанное слово иинформация. Путем перемещения его букв получить слово информация.

**Код программы:**

def fix\_word(word):

# Переставляем буквы в слове для исправления ошибки

# Меняем местами вторую и третью букву в слове (индексы 1 и 2)

corrected\_word = list(word)

corrected\_word[1], corrected\_word[2] = corrected\_word[2], corrected\_word[1]

corrected\_word = ''.join(corrected\_word)

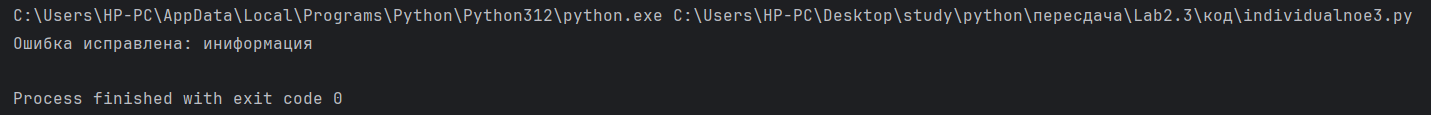
return corrected\_word

# Заданное ошибочное слово

wrong\_word = "иинформация"

# Исправленное слово

corrected\_word = fix\_word(wrong\_word)

print(f"Ошибка исправлена: {corrected\_word}")  
  


*Рисунок 9 Результат работы программы individualnoe3.py*

**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.

* 1. Какие существуют способы задания строковых литералов в языкеPython?

Строки в апострофах и в кавычках, экранированные последовательности

- служебные символы, «сырые» строки - подавляют экранирование, строки втройных апострофах или кавычках.

* 1. Какие операции и функции существуют для строк?

Оператор сложения (+), умножения (\*), принадлежности подстроки (in).

Функции:

chr() - преобразует целое число в символ; ord() - преобразует символ вцелое число; len() - возвращает длину строки;

str() - изменяет тип объекта на string.

* 1. Как осуществляется индексирование строк?

В Python строки являются упорядоченными последовательностями символьных данных и могут быть проиндексированы. Доступ к отдельным символам в строке можно получить, указав имя строки, за которым следует число в квадратных скобках [].

Индексация строк начинается с нуля: у первого символа индекс 0, следующего 1 и так далее. Индекс последнего символа в python  ‘‘длина строки минус один’’.

* 1. Как осуществляется работа со срезами для строк?

Python также допускает возможность извлечения подстроки из строки, известную как «string slice». Если s это строка, выражение формы s[m:n] возвращает часть s , начинающуюся с позиции m , и до позиции n , но не включая позицию.

Существует еще один вариант синтаксиса среза, о котором стоит упомянуть. Добавление дополнительного «:» и третьего индекса означает шаг, который указывает, сколько символов следует пропустить после извлечения каждого символа в срезе.

* 1. Почему строки Python относятся к неизменяемому типу данных?

Строки  один из типов данных, которые Python считает неизменяемыми, что означает невозможность их изменять. На самом деле нет особой необходимости изменять строки. Обычно можно легко

сгенерировать копию исходной строки с необходимыми изменениями.

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

string.istitle() определяет, начинаются ли слова строки с заглавной буквы.

* 1. Как проверить строку на вхождение в неё другой строки? s.find(<sub>) возвращает первый индекс в s который соответствует

началу строки <sub> , сели же в s нет <sub>, то функция выдаст -1

* 1. Как найти индекс первого вхождения подстроки в строку? s.find(<sub>) возвращает первый индекс в s который соответствует началу строки <sub> , сели же в s нет <sub>, то функция выдаст -1
  2. Как подсчитать количество символов в строке?len(s) возвращает количество символов в строке s.
  3. Как подсчитать то, сколько раз определённый символ встречаетсяв строке?

s.count(<sub>) возвращает количество точных вхождений подстроки

<sub> в s.

* 1. Что такое f-строки и как ими пользоваться?

В Python версии 3.6 был представлен новый способ форматирования строк. Эта функция официально названа литералом отформатированной строки, но обычно упоминается как f- строки (f- string).

Возможности форматирования строк огромны и не будут подробно описана здесь.

Одной простой особенностью f-строк, которые вы можете начать

использовать сразу, является интерполяция переменной. Вы можете указать имя переменной непосредственно в f-строковом литерале (f'string'), и python заменит имя соответствующим значением.

Пример: print(f ”Произведение {n} на {m} равно {prod}”), где m, n, prodэто переменные.

* 1. Как найти подстроку в заданной части строки?s.find(подстрока, начало, конец).
  2. Как вставить содержимое переменной в строку, воспользовавшисьметодом format()?

print(‘{}’.format(s)).

* 1. Как узнать о том, что в строке содержатся только цифры? s.isdigit() возвращает True когда строка s не пустая и все ее символы

являются цифрами, а False если нет.

* 1. Как разделить строку по заданному символу?str.split(‘заданный символ’).
  2. Как проверить строку на то, что она составлена только из строчных букв?

s.islower() возвращает True , если строка s не пустая, и все содержащиеся в ней буквенные символы строчные, а False если нет. Не алфавитные символыигнорируются.

* 1. Как проверить то, что строка начинается со строчной буквы? S[0].islower() выдаст True если строка начинается со строчной буквы и

False если нет.

* 1. Можно ли в Python прибавить целое число к строке?Нет.
  2. Как «перевернуть» строку?s[::-1].
  3. Как объединить список строк в одну строку, элементы которойразделены дефисами?

str.join(‘-‘, s), где s – это список строк.

* 1. Как привести всю строку к верхнему или нижнему регистру?s.upper(), s.lower().
  2. Как преобразовать первый и последний символы строки кверхнему регистру?

string[0].upper() + string[1:-1] + string[-1].upper()

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

s.isupper().

* 1. В какой ситуации вы воспользовались бы методом splitlines()? splitlines() делит s на строки и возвращает их в списки. Любой из

следующих символов или последовательностей символов считается

границейстроки: \n, \r, \r\n, \v или же \x0b, \f или же \x0c, \x1c, \x1d, \x1e,

\x85, \u2028,

\u2029.

* 1. Как в заданной строке заменить на что-либо все вхождения некоейподстроки?

s.replace(old, new).

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

str.startswich() и str.endswich().

* 1. Как узнать о том, что строка включает в себя только пробелы?s.isspace().
  2. Что случится, если умножить некую строку на 3?Она напечатается 3 раза.
  3. Как привести к верхнему регистру первый символ каждого словав строке?

s.title().

* 1. Как пользоваться методом partition()?

Разбивает строку при первом появлении строки аргумента и возвращает кортеж, содержащий часть перед разделителем, строку аргумента и часть после разделителя.

* 1. В каких ситуациях пользуются методом rfind()?

rfind() и find() оба используются для поиска вхождения подстроки в строку, но есть различие в том, что rfind() ищет справа налево (с конца строки), в то время как find() ищет слева направо (с начала строки). То

есть rfind() находит последнее вхождение, а find() первое вхождение подстроки в строку.

**Вывод:** в результате выполнения работы были приобретены навыки по работе со строками при написании программ с помощью языка программирования Python версии 3.x.