



МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет Информационных технологий
Кафедра Информатики и информационных технологий

направление подготовки
09.03.02 «Информационные системы и технологии»

ЛАБОРАТОРНАЯ РАБОТА № 7

Дисциплина: Backend-разработка

Тема: Строки

Выполнил(а): студент(ка) группы 221-379

Кодиров Жамшид Мурод угли
(Фамилия И.О.)

Проверил: _____
(Фамилия И.О., степень, звание) (Оценка)

Дата, подпись _____
(Дата) (Подпись)

Замечания: _____

Москва
2025

Задание

Цель работы: Написать код, который демонстрирует работу со строками в Python.

Должны быть реализованы следующие конструкции:

1. Функция, принимающая на вход некий набор параметров (минимум 3 параметра).

Внутри себя эта функция содержит заранее определённую строку, в которую можно подставлять значения переменных.

Функция возвращает строку, в которую вставлены значения, такие что

1.1. Как минимум, одно значение - это просто строка

1.2. Как минимум, одно значение - это результат арифметической операции

1.3. Как минимум, одно значение - это результат вызова другой функции

2. Функция, которая формирует строку, состоящую из повторений комбинации других строк.

Эта функция выводит получившуюся строку, где каждое повторение выводится на отдельной строке.

3. Функция, которая считает количество вхождений подстроки в строку без учёта регистра.

4. Функция, принимающая на вход строку и выводящая подстроку, содержащуюся между двумя индексами.

Индексы ДОЛЖНЫ быть больше нуля и меньше длины строки минус 1.

Тело функции ДОЛЖНО быть написано в одну строку.

5. Функция, принимающая на вход произвольное количество разных строк, где содержатся любые кириллические буквы, а также могут содержаться латинские буквы, но только такие, которые визуально неотличимы от кириллических. Регистр букв произвольный.

Эта функция ищет слова, в которых содержатся латинские буквы.

На выход возвращаются строки, где были обнаружены латинские символы и количество слов, в которых была обнаружена хотя бы одна латинская буква.

6. Функция, определяющая, является ли строка палиндромом (одинаково читается с начала и с конца).

Строка МОЖЕТ содержать как цифры, так и буквы.

7. Функция, принимающая на вход строку, содержащую несколько слов, которые разделены одним или несколькими пробелами.

У входной строки могут быть несколько пробелов в начале и в конце.

Функция убирает лишние пробелы: то есть все пробелы в начале и в конце

строки, а между словами оставляет только один пробел.

Функция возвращает длину строки после удаления лишних пробелов.

8. Функция, принимающая на вход строку, содержащую текст из нескольких предложений.

Функция заменяет символы окончания предложения на символ переноса строки.

Функция возвращает получившуюся строку.

9. Минимум 3 функции, содержащие произвольные алгоритмы работы со строками.

Функции ДОЛЖНЫ решать алгоритмы, отличные от реализованных в п. 1-8

10. Функция, которая последовательно вызывает ВСЕ вышесозданные функции.

Функция ДОЛЖНА завершаться корректно и НЕ ДОЛЖНА иметь необработанных исключений

Дополнительные требования:

Функции, созданные в шагах 1-9 ДОЛЖНЫ быть размещены в одном или нескольких отдельных файлах.

Функция из шага 10 ДОЛЖНА быть размещена в файле main.py.

В файле main.py ДОЛЖНА быть конструкция `if __name__ == "__main__":`, внутри которой ДОЛЖНА вызываться функция из шага 10.

В комментариях к каждой функции ДОЛЖНО быть отмечено, к какому шагу относится эта функция.

КРАЙНЕ ЖЕЛАТЕЛЬНО, чтобы реализуемые функции имитировали какую-то реальную логику, были как-нибудь связаны между собой и содержали как можно меньше искусственных примеров.

ЗАПРЕЩАЕТСЯ полностью копировать примеры из лекции.

РАЗРЕШАЕТСЯ использовать примеры из лекции за основу.

Ход работы

Для выполнения лабораторной работы была разработана программа на языке Python, демонстрирующая работу со строками в контексте барного бизнеса.

Программа разделена на два файла: `string_utils.py`, содержащий функции для обработки строк, и `main.py`, содержащий основную функцию для их вызова.

Все данные вводятся пользователем через терминал с помощью функции `input()`. Ниже приведён код программы и описание её работы.

Файл `string_utils.py`

```
# Функция 1: Формирование строки с подстановкой значений
def create_bar_order():
    name = input("Введите имя клиента: ")
    drink_count = int(input("Введите количество напитков: "))
    base_string = "Клиент " + name + " заказал " + str(drink_count * 2) + " напитков со скидкой " + str(get_discount()) + "%"
    return base_string

# Вспомогательная функция для функции 1
def get_discount():
    return 15

# Функция 2: Повторение комбинации строк
def repeat_bar_menu():
    menu_part1 = input("Введите первый пункт меню (например, 'Пиво - 300p'): ")
    menu_part2 = input("Введите второй пункт меню (например, 'Виски - 500p'): ")
    full_menu = menu_part1 + " " + menu_part2
    for i in range(3):
        print(full_menu)

# Функция 3: Подсчет вхождений подстроки без учета регистра
def count_cocktail():
    string = input("Введите текст с напитками: ")
    substring = input("Введите название напитка для поиска: ")
    string_lower = string.lower()
    substring_lower = substring.lower()
    count = string_lower.count(substring_lower)
    return count
```

```
# Функция 4: Извлечение подстроки между индексами
def get_bar_name_part():
    bar_name = input("Введите название бара: ")
    start = int(input("Введите начальный индекс (больше 0): "))
    end = int(input("Введите конечный индекс (меньше длины строки минус 1): "))
    if start > 0 and end < len(bar_name) - 1:
        return bar_name[start:end]
    else:
        return "Ошибка: индексы вне допустимого диапазона"

# Функция 5: Поиск латинских букв в строках
def find_latin_in_orders():
    orders = []
    count = int(input("Сколько заказов вы хотите ввести? "))
    for i in range(count):
        order = input(f"Введите заказ {i+1}: ")
        orders.append(order)
    latin_count = 0
    result_orders = []
    latin_chars = "АВСЕНКМОРТХ" # Латинские буквы,
    похожие на кириллические
    for order in orders:
        words = order.split()
        has_latin = False
        for word in words:
            for char in word:
                if char.upper() in latin_chars:
                    has_latin = True
                    break
            if has_latin:
                break
        if has_latin:
            latin_count = latin_count + 1
            result_orders.append(order)
    return result_orders, latin_count
```

```

# Функция 6: Проверка на палиндром
def find_latin_in_orders():
    orders = []
    count = int(input("Сколько заказов вы хотите
ввести? "))
    for i in range(count):
        order = input(f"Введите заказ {i+1}: ")
        orders.append(order)
    latin_count = 0
    result_orders = []
    latin_chars = "АВСЕНКМОРТХ" # Латинские буквы,
похожие на кириллические
    for order in orders:
        words = order.split()
        has_latin = False
        for word in words:
            for char in word:
                if char.upper() in latin_chars:
                    has_latin = True
                    break
            if has_latin:
                break
        if has_latin:
            latin_count = latin_count + 1
            result_orders.append(order)
    return result_orders, latin_count

# Функция 6: Проверка на палиндром
def is_palindrome():
    text = input("Введите код или название для проверки
на палиндром: ")
    text = text.lower()
    return text == text[::-1]

# Функция 7: Удаление лишних пробелов
def clean_order_spaces():
    order = input("Введите заказ с пробелами: ")
    words = order.split()
    clean_text = " ".join(words)
    return len(clean_text)

```

```

# Функция 8: Замена окончаний предложений на перенос
строки
def split_bar_rules():
    rules = input("Введите правила бара (с точками,
воскл. или вопр. знаками): ")
    result = rules.replace(".", "\n").replace("!",
"\n").replace("?", "\n")
    return result

# Функция 9.1: Подсчет гласных в строке
def count_vowels():
    bar_name = input("Введите название бара для
подсчета гласных: ")
    vowels = "аеёиоуыэюяаеiou"
    count = 0
    for char in bar_name.lower():
        if char in vowels:
            count = count + 1
    return count

# Функция 9.2: Переворот слов в строке
def reverse_words():
    order = input("Введите заказ для переворота слов:
")
    words = order.split()
    reversed_words = words[::-1]
    return " ".join(reversed_words)

# Функция 9.3: Замена пробелов на дефисы
def dash_bar_menu():
    menu = input("Введите пункты меню через пробел: ")
    return menu.replace(" ", "-")

```

Файл main.py

```

from string_utils import *

# Функция 10: Вызов всех функций
def run_all_bar_functions():
    # Вызов функции 1
    order = create_bar_order()

```

```
print("Заказ:", order)

# Вызов функции 2
print("\nМеню бара:")
repeat_bar_menu()

# Вызов функции 3
mojito_count = count_cocktail()
print("\nКоличество упоминаний напитка:",
mojito_count)

# Вызов функции 4
name_part = get_bar_name_part()
print("Часть названия:", name_part)

# Вызов функции 5
latin_orders, latin_word_count =
find_latin_in_orders()
print("\nЗаказы с латинскими буквами:",
latin_orders)
print("Количество заказов с латинскими буквами:",
latin_word_count)

# Вызов функции 6
if is_palindrome():
    print("Введённый текст - палиндром!")
else:
    print("Введённый текст - не палиндром")

# Вызов функции 7
clean_length = clean_order_spaces()
print("Длина заказа без лишних пробелов:",
clean_length)

# Вызов функции 8
split_rules = split_bar_rules()
print("\nПравила бара:")
print(split_rules)

# Вызов функции 9.1
```



```

    vowel_count = count_vowels()
    print("Количество гласных в названии:",
vowel_count)

    # Вызов функции 9.2
    reversed_order = reverse_words()
    print("Перевернутый заказ:", reversed_order)

    # Вызов функции 9.3
    dashed_menu = dash_bar_menu()
    print("Меню с дефисами:", dashed_menu)

if __name__ == "__main__":
    run_all_bar_functions()

```

Описание работы программы

1. Программа запускается из файла main.py через конструкцию if __name__ == "__main__".
2. Функция run_all_bar_functions() последовательно вызывает все функции из string_utils.py.
3. Каждая функция запрашивает данные через input():
 - Функция 1: Имя клиента и количество напитков для формирования заказа.
 - Функция 2: Два пункта меню для вывода трёх раз.
 - Функция 3: Текст и подстроку для подсчёта вхождений.
 - Функция 4: Название бара и индексы для извлечения подстроки.
 - Функция 5: Количество заказов и сами заказы для поиска латинских букв.
 - Функция 6: Текст для проверки на палиндром.
 - Функция 7: Заказ с пробелами для подсчёта длины без них.
 - Функция 8: Правила бара для разделения на строки.
 - Функция 9.1: Название бара для подсчёта гласных.
 - Функция 9.2: Заказ для переворота слов.
 - Функция 9.3: Меню для замены пробелов на дефисы.
4. Результаты выводятся в терминал после каждого ввода.

Пример выполнения

При запуске программы в терминале вводятся данные, например:

- "Введите имя клиента: Иван"
- "Введите количество напитков: 3"
- Вывод: "Заказ: Клиент Иван заказал 6 напитков со скидкой 15%"

- "Введите первый пункт меню: Пиво - 300р"
- "Введите второй пункт меню: Виски - 500р"
- Вывод:

```

Меню бара :

Пиво - 300р Виски - 500р

Пиво - 300р Виски - 500р

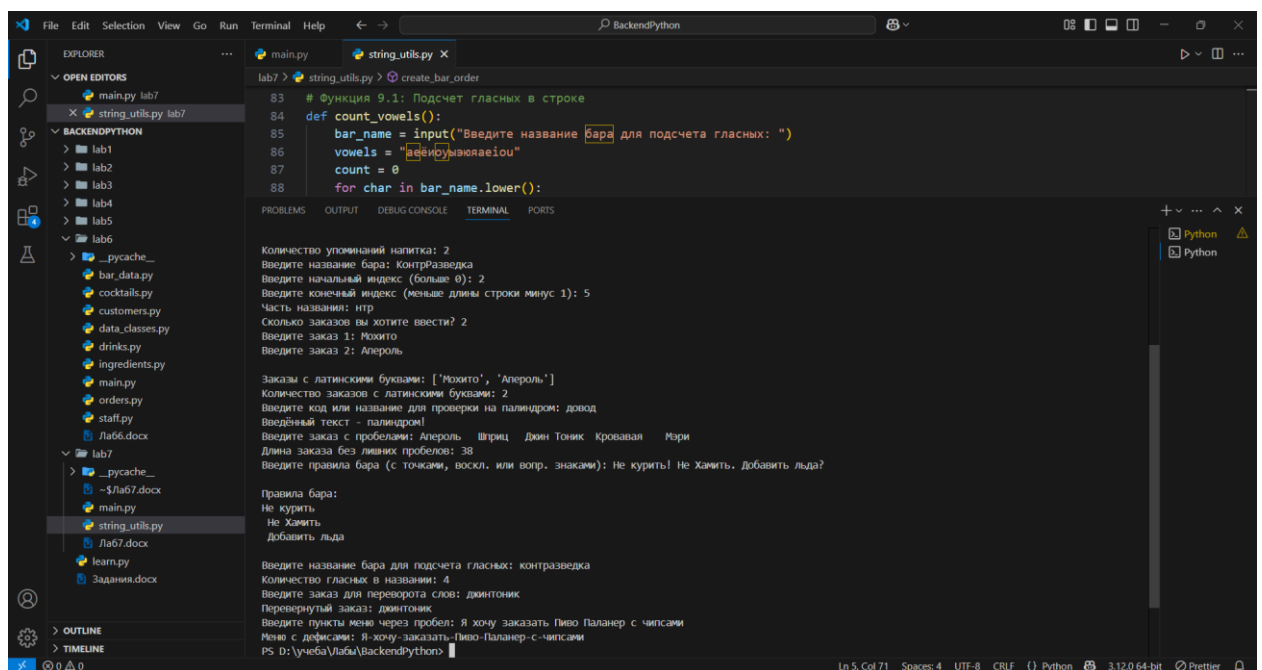
Пиво - 300р Виски - 500р

```

И так далее для всех функций.

Вывод

В ходе лабораторной работы была создана программа, демонстрирующая базовые операции со строками в Python. Реализованы функции для формирования строк, повторения текста, подсчёта подстрок, извлечения подстрок, поиска латинских символов, проверки палиндромов, удаления пробелов, разделения текста и дополнительных алгоритмов (подсчёт гласных, переворот слов, замена пробелов). Все функции успешно работают с данными, введёнными через терминал, и корректно завершаются без необработанных исключений при правильном вводе. Программа соответствует требованиям задания и может быть использована для изучения работы со строками в Python.



Ссылка на Github: <https://github.com/QodirovJM/BackendPython>