

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Отчет по лабораторной работе №2.17

**Разработка приложений с интерфейсом командной строки (CLI) в
Python3.**

по дисциплине «Технологии программирования и алгоритмизация»

Выполнил студент группы ИВТ-б-о-20-1

Лысенко И.А. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверила Воронкин Р.А. _____

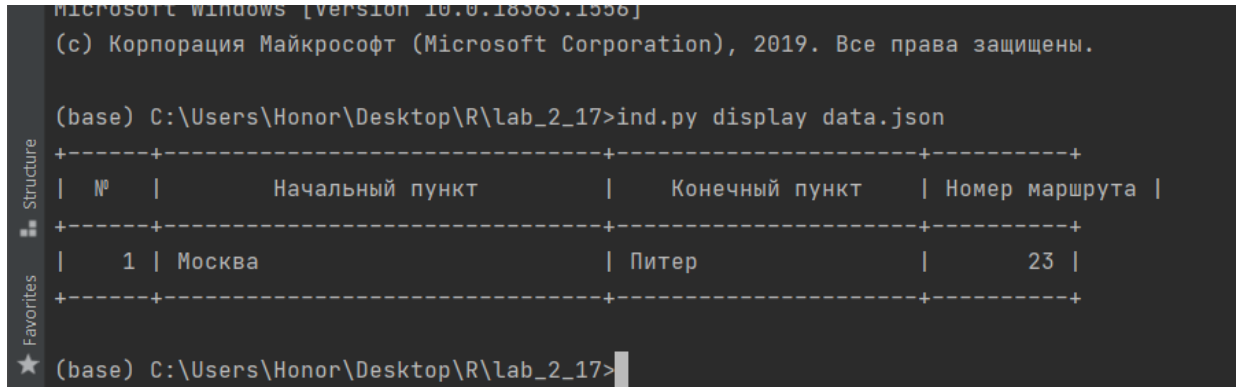
(подпись)

Ставрополь 2021

Цель работы: приобретение построения приложений с интерфейсом командной строки с помощью языка программирования Python версии 3.x.

Ход работы:

1. Проработал и разобрал примеры.
2. Реализовал для своего варианта лабораторной работы 2.16 интерфейс командной строки (CLI).



```
Microsoft Windows [Version 10.0.18363.1556]
(c) Корпорация Майкрософт (Microsoft Corporation), 2019. Все права защищены.

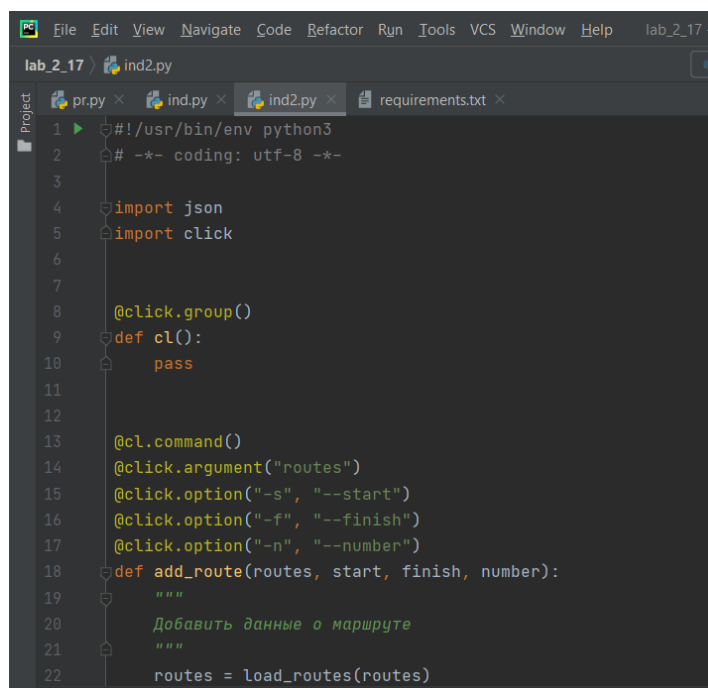
(base) C:\Users\Honor\Desktop\R\lab_2_17>ind.py display data.json

+-----+-----+-----+-----+
| № | Начальный пункт | Конечный пункт | Номер маршрута |
+-----+-----+-----+-----+
| 1 | Москва | Питер | 23 |
+-----+-----+-----+-----+

(base) C:\Users\Honor\Desktop\R\lab_2_17>
```

Рисунок 2.1 – Результат работы

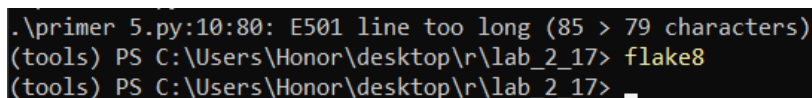
3. Изучил библиотеку click.



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import json
5 import click
6
7
8 @click.group()
9 def cl():
10     pass
11
12
13 @cl.command()
14 @click.argument("routes")
15 @click.option("-s", "--start")
16 @click.option("-f", "--finish")
17 @click.option("-n", "--number")
18 def add_route(routes, start, finish, number):
19     """
20     Добавить данные о маршруте
21     """
22     routes = load_routes(routes)
```

Рисунок 3.1 – Проверка flake8

4. Проверил программы через flake8.



```
.\primer 5.py:10:80: E501 line too long (85 > 79 characters)
(tools) PS C:\Users\Honor\Desktop\R\lab_2_17> flake8
(tools) PS C:\Users\Honor\Desktop\R\lab_2_17>
```

Рисунок 4.1 – Проверка flake8

Ответы на вопросы.

1. В чем отличие терминала и консоли?

Терминал – устройство или ПО, выступающее посредником между человеком и вычислительной системой.

Консоль – исторически реализация терминала с клавиатурой и текстовым дисплеем

2. Что такое консольное приложение?

Консольное приложение — вид ПО, разработанный с расчётом на работу внутри оболочки командной строки, т.е. опирающийся на текстовый ввод-вывод.

3. Какие существуют средства языка программирования Python для построения приложений командной строки?

- Argparse;
- Docopt;
- Click;
- Fire;
- Cement;
- Cleo

4. Какие особенности построение CLI с использованием модуля sys?

Это базовый модуль, который с самого начала поставлялся с Python. Он использует подход, очень похожий на библиотеку C, с использованием `argc` и `argv` для доступа к аргументам. Модуль `sys` реализует аргументы командной строки в простой структуре списка с именем `sys.argv`.

Каждый элемент списка представляет собой единственный аргумент. Первый элемент в списке `sys.argv [0]` – это имя скрипта Python. Остальные элементы списка, от `sys.argv [1]` до `sys.argv [n]`, являются аргументами командной строки с 2 по `n`. В качестве разделителя между аргументами используется пробел. Значения аргументов, содержащие пробел, должны быть заключены в кавычки, чтобы их правильно проанализировал `sys`.

5. Какие особенности построение CLI с использованием модуля getopt?

Модуль getopt в Python идет немного дальше и расширяет разделение входной строки проверкой параметров. Основанный на функции C getopt, он позволяет использовать как короткие, так и длинные варианты, включая присвоение значений. На практике для правильной обработки входных данных требуется модуль sys. Для этого необходимо заранее загрузить как модуль sys, так и модуль getopt. Затем из списка входных параметров мы удаляем первый элемент списка и сохраняем оставшийся список аргументов командной строки в переменной.

6. Какие особенности построение CLI с использованием модуля argparse?

- 1) Для начала работы с argparse необходимо задать парсер.
- 2) Далее, парсеру стоит указать, какие объекты Вы от него ждете.
- 3) Если действие для данного аргумента не задано, то по умолчанию он будет сохраняться в namespace, причем мы также можем указать тип этого аргумента. Если имя возвращаемого аргумента задано, его значение будет сохранено в соответствующем атрибуте namespace.

Вывод. Были приобретены навыки построения приложений с интерфейсом командной строки с помощью языка программирования Python версии 3.x.