

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

Отчет по лабораторной работе № 2.17

Разработка приложений с интерфейсом командной строки (CLI) в Python3

Выполнил студент группы ИВТ-б-о-20-1

Ищенко М.А.

Работа защищена « » _____ 20__ г.

Проверил(а) _____

Ставрополь 2021

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

Создан общедоступный репозиторий на GitHub. Дополнен файл .gitignore необходимыми правилами для работы с IDE PyCharm.

Проработан пример из лабораторной работы, рис. 1

```
Terminal: Local x +
(lab2.17) C:\Users\maks\doc\lab2.17>python primer.py add data.json --name="Нектаров Павел" --post="Пчеловод" --year=2020

(lab2.17) C:\Users\maks\doc\lab2.17>python primer.py add data.json --name="Двойной Вениамин" --post="Медовар" --year=2018

(lab2.17) C:\Users\maks\doc\lab2.17>python primer.py display data.json
+-----+-----+-----+-----+
| № |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+
|  1 | Нектаров Павел          |      Пчеловод      |      2020     |
+-----+-----+-----+-----+
|  2 | Двойной Вениамин        |      Медовар        |      2018     |
+-----+-----+-----+-----+

(lab2.17) C:\Users\maks\doc\lab2.17>python primer.py select data.json --period=1
+-----+-----+-----+-----+
| № |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+
|  1 | Нектаров Павел          |      Пчеловод      |      2020     |
+-----+-----+-----+-----+
```

Рисунок 1 – Пример

Выполнено индивидуальное задание варианта 6, рис. 2

```
Terminal: Local x +
(lab2.17) C:\Users\maks\doc\lab2.17>python ind1.py add data.json --destination="Зальцбург" --number=34 --time="14:00"

(lab2.17) C:\Users\maks\doc\lab2.17>python ind1.py add data.json --destination="Колыма" --number=87 --time="12:00"

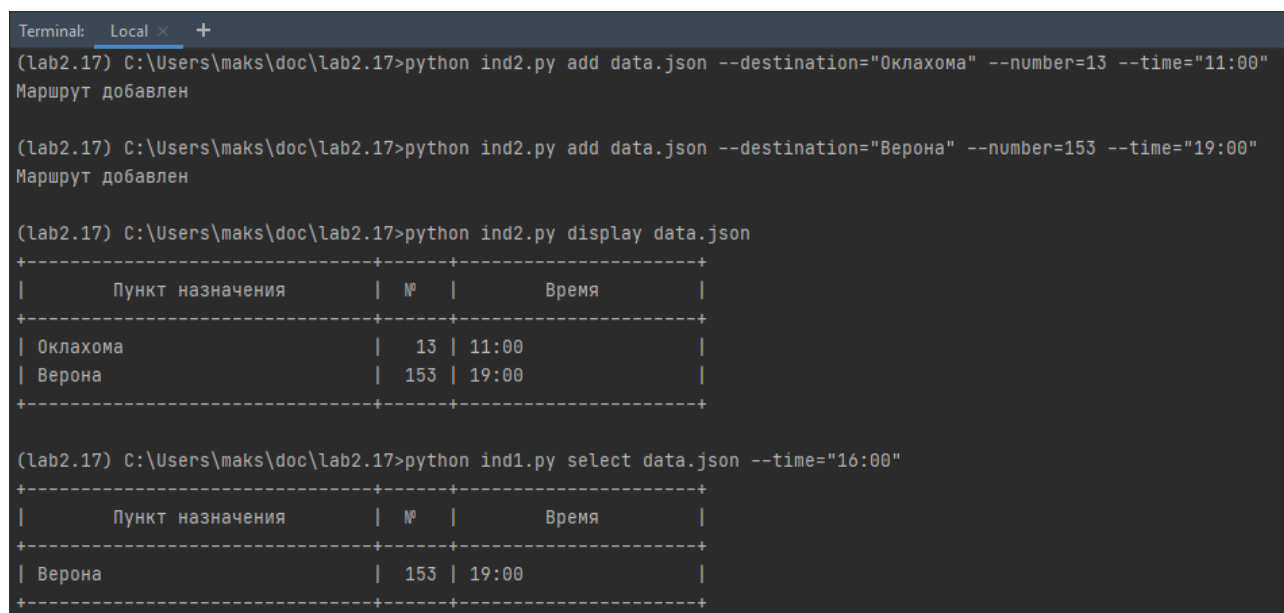
(lab2.17) C:\Users\maks\doc\lab2.17>python ind1.py display data.json
+-----+-----+-----+-----+
|      Пункт назначения      | № |      Время      |
+-----+-----+-----+-----+
| Зальцбург                  | 34 | 14:00           |
| Колыма                     | 87 | 12:00           |
+-----+-----+-----+-----+

(lab2.17) C:\Users\maks\doc\lab2.17>python ind1.py select data.json --time="13:00"
+-----+-----+-----+-----+
|      Пункт назначения      | № |      Время      |
+-----+-----+-----+-----+
| Зальцбург                  | 34 | 14:00           |
+-----+-----+-----+-----+

(lab2.17) C:\Users\maks\doc\lab2.17>
```

Рисунок 2 – Индивидуальное задание

Выполнено индивидуальное задание варианта 6 с использованием модуля click, рис. 3



```
Terminal: Local x +
(lab2.17) C:\Users\maks\doc\lab2.17>python ind2.py add data.json --destination="Оклахома" --number=13 --time="11:00"
Маршрут добавлен

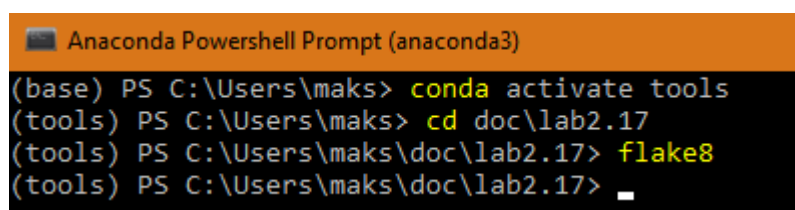
(lab2.17) C:\Users\maks\doc\lab2.17>python ind2.py add data.json --destination="Верона" --number=153 --time="19:00"
Маршрут добавлен

(lab2.17) C:\Users\maks\doc\lab2.17>python ind2.py display data.json
+-----+-----+-----+
| Пункт назначения | № | Время |
+-----+-----+-----+
| Оклахома        | 13 | 11:00 |
| Верона          | 153 | 19:00 |
+-----+-----+-----+

(lab2.17) C:\Users\maks\doc\lab2.17>python ind1.py select data.json --time="16:00"
+-----+-----+-----+
| Пункт назначения | № | Время |
+-----+-----+-----+
| Верона          | 153 | 19:00 |
+-----+-----+-----+
```

Рисунок 3 – Индивидуальное задание с модулем click

Программы проверены на flake8, рис. 4



```
Anaconda PowerShell Prompt (anaconda3)
(base) PS C:\Users\maks> conda activate tools
(tools) PS C:\Users\maks> cd doc\lab2.17
(tools) PS C:\Users\maks\doc\lab2.17> flake8
(tools) PS C:\Users\maks\doc\lab2.17> _
```

Рисунок 4 – Проверка заданий

Контрольные вопросы:

1. В чем отличие терминала и консоли?

Терминал (от лат. terminus — граница) — устройство или ПО, выступающее посредником между человеком и вычислительной системой. Обычно данный термин используется, когда точка доступа к системе вынесена в отдельное физическое устройство и предоставляет свой пользовательский интерфейс на основе внутреннего интерфейса (например, сетевых протоколов). Консоль console — исторически реализация терминала с клавиатурой и текстовым дисплеем. В настоящее время это слово часто используется как

синоним сеанса работы или окна оболочки командной строки. В том же смысле иногда применяется и слово “терминал”.

2. Что такое консольное приложение?

Консольное приложение `console application` — 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?

Python 3 поддерживает несколько различных способов обработки аргументов командной строки.

Встроенный способ – использовать модуль `sys`. С точки зрения имен и использования, он имеет прямое отношение к библиотеке C (`libc`). Вторым способом – это модуль `getopt`, который обрабатывает как короткие, так и длинные параметры, включая оценку значений параметров.

4. Какие особенности построения 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`

5. Какие особенности построения CLI с использованием модуля `getopt`?

Как вы могли заметить ранее, модуль `sys` разбивает строку командной строки только на отдельные фасы. 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.

6. Какие особенности построения CLI с использованием модуля `argparse`

Начиная с версий Python 2.7 и Python 3.2, в набор стандартных библиотек была включена библиотека `argparse` для обработки аргументов (параметров, ключей) командной строки.

`argparse` предлагает:

- анализ аргументов `sys.argv`;
- конвертирование строковых аргументов в объекты вашей программы и работа с ними;
- форматирование и вывод информативных подсказок.

Вывод: в ходе занятия были приобретены навыки по построению приложений с интерфейсом командной строки с помощью языка программирования Python версии 3.x