

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

Кафедра инфокоммуникаций

Отчет по лабораторной работе № 19(2.17)
Рекурсия в языке Python

по дисциплине «Технологии программирования и алгоритмизации»

Выполнил студент группы ИВТ-б-о-
20-1

Злыгостев И.С. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____

(подпись)

Ставрополь 2021

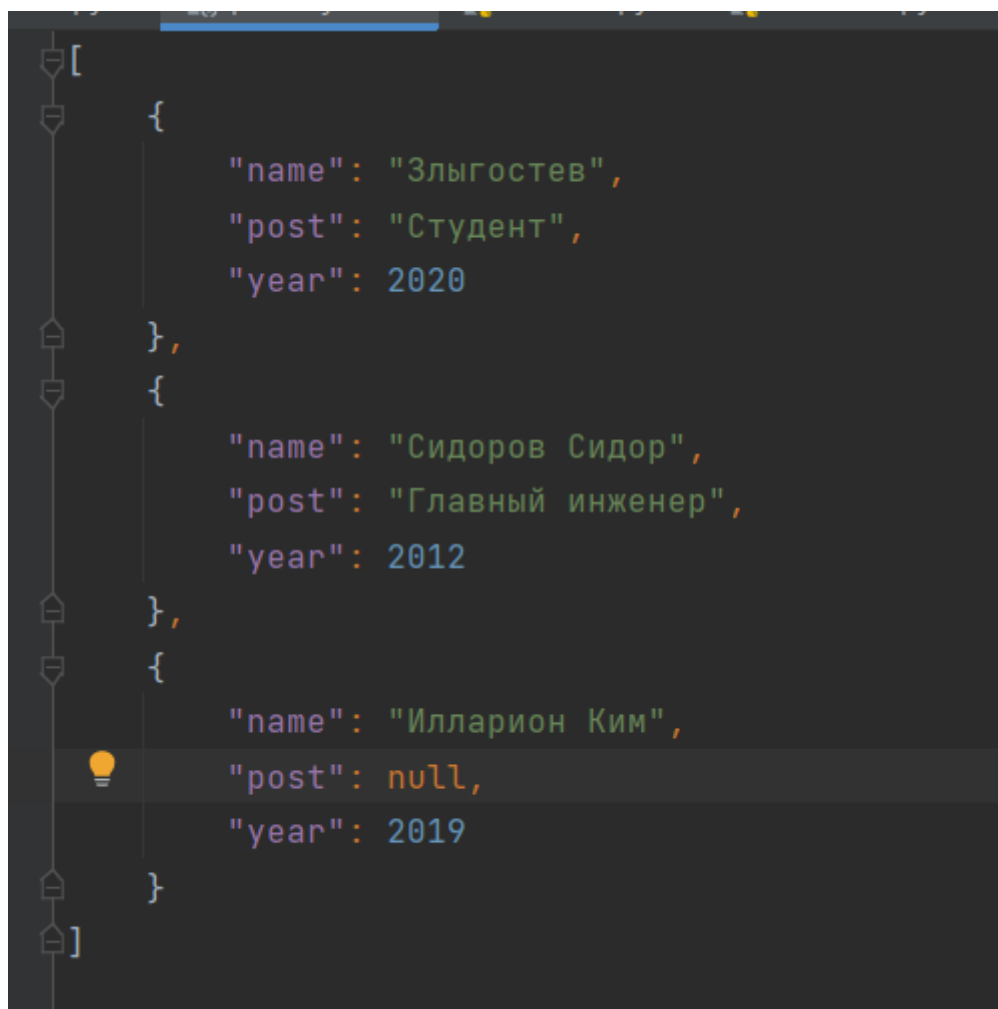
Цель работы: приобретение построения приложений с интерфейсом командной строки с помощью языка программирования Python версии 3.x.

Ход работы:

1. Изучив методические указания, приступил к разбору примера лабораторной работы: разобрал код задания и посмотрел результат его выполнения.

```
(venv) C:\Users\zligo\git\demo-2.17\Zadanie>python primer.py add primer.json -n="Илларион Ким" -y="2019"
```

Рисунок 19.1 – Ввод данных



```
[  
  {  
    "name": "Злыгостев",  
    "post": "Студент",  
    "year": 2020  
  },  
  {  
    "name": "Сидоров Сидор",  
    "post": "Главный инженер",  
    "year": 2012  
  },  
  {  
    "name": "Илларион Ким",  
    "post": null,  
    "year": 2019  
  }  
]
```

Рисунок 19.2 – Результат выполнения

2. Затем приступил к выполнению индивидуальных заданий.

3. По примеру лабораторной работы написал код к первому заданию и проверил его работоспособность.

Рисунок 19.3 – Введённые данные

Рисунок 19.4 – Результат выполнения скрипта

4. Ко второму индивидуальному заданию самостоятельно изучил библиотеку click и приступил к написанию кода.

Рисунок 19.5 – Использование библиотеки

```
(venv) C:\Users\zligo\git\demo-2.17\Zadanie>python zadanie2.py -c add zadanie2.json -s Lissabon -v Starship -n RF-862317
Рейс добавлен
```

Рисунок 19.6 – Введённые данные

```
{
  "stay": "Moscow",
  "number": "RF-86231",
  "value": "Airbus"
},
{
  "stay": "London",
  "number": "RF-862131",
  "value": "Boing"
},
{
  "stay": "Paris",
  "number": "RF-862311",
  "value": "Airbus"
},
{
  "stay": "Lissabon",
  "number": "RF-862317",
  "value": "Starship"
}
```

Рисунок 19.7 – Результат выполнения кода

Контрольные 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: в ходе выполнения лабораторной работы были изучены основные методы по работе с построением приложений с интерфейсом командной строки при помощи языка программирования Python.