

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

**Кафедра инфокоммуникаций**

**Лабораторная работа 2.17**

Разработка приложений с интерфейсом командной строки (CLI) в Python3

Выполнил студент группы ИВТ-б-о-20-1

Симанский М.Ю « » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверил Воронкин Р.А. \_\_\_\_\_

(подпись)

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

Запустим программу через консоль с нужными параметрами (рис. 1)

```
(base) C:\Users\rac-e\YandexDisk\Лабы 3 семестр\Технологии программирования\Лаб 19(2.17)>python individual1.py add data.json -n = "Magnit" -p = "Chocolate " -pr = 32
usage: shops add [-h] -n NAME [-p PRODUCT] -pr PRICE filename
shops add: error: argument -pr/--price: invalid int value: '='

(base) C:\Users\rac-e\YandexDisk\Лабы 3 семестр\Технологии программирования\Лаб 19(2.17)>python individual1.py add data.json -n = "Magnit" -p = "Chocolate " -pr 32
usage: shops [-h] [--version] {add,display,select} ...
shops: error: unrecognized arguments: Magnit Chocolate

(base) C:\Users\rac-e\YandexDisk\Лабы 3 семестр\Технологии программирования\Лаб 19(2.17)>python individual1.py add data.json -n "Magnit" -p "Chocolate " -pr 32
```

Рисунок 1 – Команда выполнена

В результате выполнения был создан json файл (рис. 2)

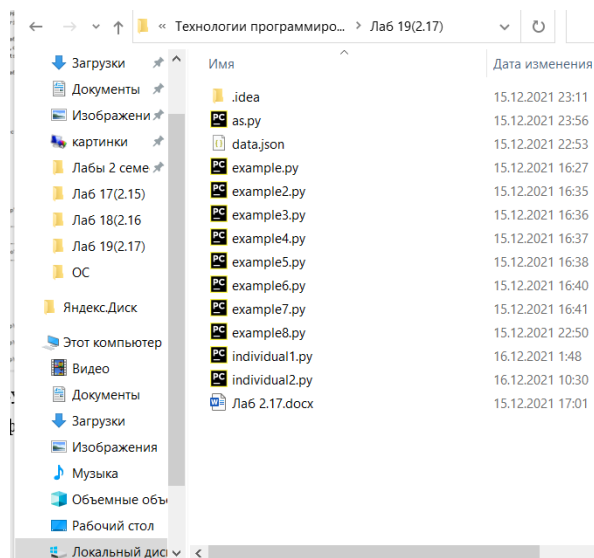


Рисунок 2 – Файл

Вызовем в консоли команду сохранения (рис. 3).

```

5 (base) C:\Users\rac-e\YandexDisk\Лабы 3 семестр\Технологии программирования\Лаб 19(2.17)>python individual1.py add data.json -n "Magnit" -p "Chocolate" -pr
Favorites Данные сохранены
Данные сохранены

```

Рисунок 3 – Сообщение об успехе операции

Проверим данные (рис. 4)

```

(base) C:\Users\rac-e\YandexDisk\Лабы 3 семестр\Технологии программирования\Лаб 19(2.17)>python individual1.py display data.json
Валидация прошла успешно

```

No	Название.	Товар	Цена
1	Magnit	Chocolate	32

Рисунок 4 – Данные из файла

**Вывод:** в результате выполнения работы были приобретены навыки по работе с интерфейсом командной строки с помощью языка программирования Python версии 3.x.

## Ответы на 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`). Второй способ – это модуль `getopt`, который обрабатывает как короткие, так и длинные параметры, включая оценку значений параметров.

## 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`

## 5. Какие особенности построение 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 построение 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.

- Для начала рассмотрим, что интересного предлагает `argparse` :
- анализ аргументов `sys.argv` ;
- конвертирование строковых аргументов в объекты Вашей программы и работа с ними;
- форматирование и вывод информативных подсказок.