

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №3**  
**дисциплины «анализ данных»**

Выполнил:  
Середа Кирилл Витальевич  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника», очная  
форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Роман Александрович

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2024 г.

# Тема: Разработка приложений с интерфейсом командной строки (CLI) в Python3

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

## Ход выполнения заданий

### 1) Проработал пример из лабораторной работы

```
(DA_3) MelancholySeal@Kiras-MacBook-Air prog % python primer.py --help
usage: workers [-h] [--version] {add,display,select} ...

positional arguments:
  {add,display,select}
    add                Add a new worker
    display            Display all workers
    select             Select the workers

optional arguments:
  -h, --help            show this help message and exit
  --version             show program's version number and exit
(DA_3) MelancholySeal@Kiras-MacBook-Air prog % python primer.py add --name="Петров Анатолий" --post="Кассир" --year=2020 --filename="data.json"
(DA_3) MelancholySeal@Kiras-MacBook-Air prog % python primer.py display
usage: workers display [-h] -f FILENAME
workers display: error: the following arguments are required: -f/--filename
(DA_3) MelancholySeal@Kiras-MacBook-Air prog % python primer.py display -f'data.json'
+-----+-----+-----+-----+
| № |      Ф.И.О.      | Должность | Год |
+-----+-----+-----+-----+
| 1 | Петров Анатолий  | Кассир    | 2020 |
+-----+-----+-----+-----+
```

Рисунок 1 – Использование функционала программы

### 2) Сделал индивидуальное задание, переработав функционал программы из предыдущей лабораторной работы

```
(DA_3) MelancholySeal@Kiras-MacBook-Air prog % python ind.py --help
usage: students [-h] [--version] {add,display} ...

positional arguments:
  {add,display}
    add        Add a new student
    display    Display all students

optional arguments:
  -h, --help            show this help message and exit
  --version             show program's version number and exit
(DA_3) MelancholySeal@Kiras-MacBook-Air prog % python ind.py add --name="Петров Петр" --group="2" --grades=4.5 --filename="data2.json"
Traceback (most recent call last):
  File "ind.py", line 85, in <module>
    main()
  File "ind.py", line 76, in main
    students = load_from_json(args.filename)
  File "ind.py", line 48, in load_from_json
    with open(filepath, 'r', encoding='utf-8') as file:
FileNotFoundError: [Errno 2] No such file or directory: 'data2.json'
(DA_3) MelancholySeal@Kiras-MacBook-Air prog % python ind.py add --name="Петров Петр" --group="2" --grades=4.5 --filename="data.json"
(DA_3) MelancholySeal@Kiras-MacBook-Air prog % python ind.py display -f'data.json'
+-----+-----+-----+-----+
|      Ф.И.О.      | Номер группы | Успеваемость |
+-----+-----+-----+-----+
| Петров Петр      | 2            | 4.5          |
+-----+-----+-----+-----+
```

Рисунок 2 – Использование программы ind.py

3) Выполнил задание повышенной сложности, выполнив индивидуальное задание с помощью пакета click

```
(DA_3) MelancholySeal@Kiras-MacBook-Air prog % python ind_click.py --help
Usage: ind_click.py [OPTIONS] COMMAND [ARGS]...

Options:
  --help  Show this message and exit.

Commands:
  add      Add a new student.
  display  Display all students.
(DA_3) MelancholySeal@Kiras-MacBook-Air prog % python ind_click.py add -f students.json -n "Иванов Иван" -g 101 -r "5 4 3"
(DA_3) MelancholySeal@Kiras-MacBook-Air prog % pythhon ind_click.py display -f students.json
zsh: command not found: pythhon
(DA_3) MelancholySeal@Kiras-MacBook-Air prog % python ind_click.py display -f students.json
+-----+-----+-----+
| Ф.И.О. | Номер группы | Успеваемость |
+-----+-----+-----+
+-----+-----+-----+
(DA_3) MelancholySeal@Kiras-MacBook-Air prog % python ind_click.py add -f students.json -n "Иванов Иван" -g 101 -r "5"
(DA_3) MelancholySeal@Kiras-MacBook-Air prog % python ind_click.py display -f students.json
+-----+-----+-----+
| Ф.И.О. | Номер группы | Успеваемость |
+-----+-----+-----+
| Иванов Иван | 101 | 5.0 |
+-----+-----+-----+
★ (DA_3) MelancholySeal@Kiras-MacBook-Air prog %
```

Рисунок 3 – Использование программы ind\_click.py

### Ответы на вопросы

#### 1. Отличие между терминалом и консолью:

- **Терминал** - это программа, обеспечивающая пользовательский интерфейс для доступа к командной строке операционной системы. Он может быть графическим или текстовым, и включает в себя такие функции, как ввод команд, вывод результатов и т.д.

- **Консоль** - это окно или область, в которой отображается командная строка операционной системы. Терминал использует консоль для взаимодействия с пользователем. Терминал и консоль часто используются взаимозаменяемо, хотя они имеют различные значения.

#### 2. Консольное приложение:

- Консольное приложение - это программа, предназначенная для выполнения в текстовом интерфейсе, обычно через командную строку (терминал). Консольные приложения обычно не имеют графического интерфейса пользователя и выполняются в текстовом режиме.

#### 3. Средства Python для построения приложений командной строки:

- **argparse**: Этот модуль предоставляет простой способ создания интерфейса командной строки. Он позволяет определять аргументы и параметры командной строки, автоматически генерирует справку и поддерживает различные типы аргументов.

- **getopt**: Этот модуль обеспечивает парсинг опций командной строки в стиле Unix. Он более низкоуровневый, чем **argparse**, и требует больше усилий для использования, но предоставляет более гибкий контроль над парсингом аргументов.

- **sys.argv**: Это стандартный список аргументов командной строки в Python. Он предоставляет доступ к аргументам командной строки, переданным скрипту при его запуске.

#### 4. Особенности построения CLI с использованием модуля **sys**:

- **sys.argv** содержит список аргументов командной строки, переданных скрипту при его запуске.

- Парсинг аргументов должен быть реализован вручную с использованием методов строк и списков.

#### 5. Особенности построения CLI с использованием модуля **getopt**:

- **getopt** предоставляет функции для парсинга аргументов командной строки в стиле Unix.

- Он поддерживает короткие и длинные опции с корректным разбором аргументов.

- Требуется больше усилий для использования по сравнению с **argparse**.

#### 6. Особенности построения CLI с использованием модуля **argparse**:

- **argparse** предоставляет более высокоуровневый и удобный интерфейс для определения аргументов командной строки и создания справки.

- Он автоматически генерирует справку на основе определенных аргументов.
- Поддерживает различные типы аргументов и проверку их значений.
- Обеспечивает легкую интеграцию с кодом программы и ясный синтаксис для определения аргументов

Вывод: в ходе лабораторной работы были приобретены знания о построение приложения с интерфейсом командной строки с помощью языка программирования Python версии 3.x.