

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙ-
СКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

Отчет по лабораторной работе №2.17

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

Выполнил студент группы

ИВТ-б-о-21-1

Харченко Б.Р. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил доцент

Кафедры инфокоммуникаций, старший
преподаватель

Воронкин Р.А.

(подпись)

Ставрополь 2022

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

Ход работы:

1. Создал проект Pycharm в папке репозитория, проработал примеры ЛР.

```
(env) D:\Git\Python Labs\Lab_12\Python_Lab_12>python primers\main.py display workers.json
```

No	Ф.И.О.	Должность	Год
1	Харченко Б.Р.	Старший слесарь	2014
2	Горшков В.И.	Слесарь	2017

Рисунок 2 – Результат работы примера

Индивидуальное задание. Для своего варианта лабораторной работы 2.16 необходимо дополнительно реализовать интерфейс командной строки (CLI).

```
(env) D:\Git\Python Labs\Lab_12\Python_Lab_12>python individual\individual1.py display students.json
```

No	Ф.И.О.	Группа	Успеваемость
1	Харченко Б.Р.	2	5 5 5 5 5
2	Криворот В.Г.	2	2 2 3 4 5

```
(env) D:\Git\Python Labs\Lab_12\Python_Lab_12>
```

Рисунок 3 – Проверка работы программы

```
(env) D:\Git\Python Labs\Lab_12\Python_Lab_12>python individual/individual1.py select students.json
```

No	Ф.И.О.	Группа	Успеваемость
1	Криворот В.Г.	2	2 2 3 4 5

Рисунок 4 – Проверка работы программы

Задание повышенной сложности. Для своего варианта лабораторной работы 2.16 необходимо реализовать интерфейс командной строки с использованием пакета click.

```
@click.group()
def cli():
    pass

@cli.command()
@click.argument('filename')
@click.option("-n", "--name")
@click.option("-g", "--group")
@click.option("-p", "--progress")
def add(filename, name, group, progress):
```

Рисунок 5 – Решение задачи с помощью пакета click

No	Ф.И.О.	Группа	Успеваемость
1	dsa	2	5 4 3 2 5
2	Bogdan	1	5 5 5 5 5

```
(env) D:\Git\Python Labs\Lab_12\Python_Lab_12>
```

Рисунок 6 – Проверка работы программы

Вывод: в результате выполнения лабораторной работы были получены практические навыки и теоретические сведения для построения приложений 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.