

**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

Богдан А. А. « » _____ 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 репозиторий в GitHub, дополнил правила в .gitignore для работы с IDE PyCharm с ЯП Python, выбрал лицензию MIT, клонировал его на компьютер и организовал в соответствии с моделью ветвления git-flow.

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

```
D:\WAR\gi\Lab-2.17\Primers>python pre1.py display file1.json
```

No	Ф.И.О.	Должность	Год
1	А Иван	директор	2004

Рисунок 2 – Результат работы примера

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

```
D:\WAR\gi\Lab-2.17\tasks>python ind1.py display ind1.json
```

Ф.И.О.	Группа	Оценки
Андрей	A4	3,3,3,3,3
Виктор	A4	4,5,4,5,4
Петр	B4	5,5,5,5,5

Рисунок 3 – Проверка работы программы

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

```
@click.command()
@click.option("-c", "--command")
@click.argument('file_name')
@click.option("-n", "--name")
@click.option("-g", "--group")
@click.option("-pr", "--progress")
def main(command, name, group, progress, file_name):
    students_load = load_students(file_name)
```

Рисунок 5 – Решение задачи с помощью пакета click

```
D:\WAR\gi\Lab-2.17\tasks>python ind2.py display ind1.json
```

Ф.И.О.	Группа	Оценки
Андрей	A4	3,3,3,3,3
Виктор	A4	4,5,4,5,4
Петр	B4	5,5,5,5,5

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 спо-

соб – 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.