

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

Кафедра инфокоммуникаций

Отчет по лабораторной работе №12
Разработка приложений с интерфейсом командной
строки (CLI) в Python3
По дисциплине «Технологии программирования и алгоритмизация»

Выполнил студент группы ИВТ-б-о-20-1

Галяс Д. И. « » _____ 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:

Ссылка на репозиторий: <https://github.com/DIMITRY-GALYAS1/Laba-2.17.git>

1. Создал новый репозиторий на github, после клонировал его и создал в папке репозитория новый проект PyCharm.
2. Выполнил пример.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import argparse
import json
import os.path
from datetime import date

def add_worker(staff, name, post, year):
    """
    Добавить данные о работнике.
    """
    staff.append(
        {
            "name": name,
            "post": post,
            "year": year
        }
    )

    return staff

def display_workers(staff):
    """
    Отобразить список работников.
    """
    # Проверить, что список работников не пуст.
    if staff:
        # Заголовки таблицы
```

Рисунок 1. Код примера

```
PS C:\Users\lizeq\PycharmProjects\pythonProject12> python primer.py add data.json --name="Сидоров Сидор" --post="Главный инженер" --year=2012
PS C:\Users\lizeq\PycharmProjects\pythonProject12> python primer.py display data.json
+-----+-----+-----+-----+
| No |      Ф.И.О.      |      Должность      |      Год      |
+-----+-----+-----+-----+
|  1 | Сидоров Сидор    |      Главный инженер      |      2012      |
+-----+-----+-----+-----+
PS C:\Users\lizeq\PycharmProjects\pythonProject12> python primer.py select data.json --period=12
Список работников пуст.
PS C:\Users\lizeq\PycharmProjects\pythonProject12> python primer.py select data.json --period=1
+-----+-----+-----+-----+
| No |      Ф.И.О.      |      Должность      |      Год      |
+-----+-----+-----+-----+
|  1 | Сидоров Сидор    |      Главный инженер      |      2012      |
+-----+-----+-----+-----+
```

Рисунок 2. Выполнение примера

3. Выполнил индивидуальное задание.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import argparse
import json
import os.path
import sys
import jsonschema

def add_student(students, name, group, progress):
    """
    Запросить данные о студенте.
    """
    students.append(
        {
            'name': name,
            'group': group,
            'progress': progress,
        }
    )
    return students

def display_students(students):
    """
    Отобразить список студентов.
    """
    # Проверить, что список студентов не пуст.
    if students:
        # Заголовок таблицы
```

Рисунок 3. Код индивидуального задания

```
PS C:\Users\lizeq\PycharmProjects\pythonProject12> python individual_1.py add data.json --name="Сидоров Сидор" --group="4" --progress=23445
Данные сохранены
PS C:\Users\lizeq\PycharmProjects\pythonProject12> python individual_1.py add data.json --name="Иванов Иван" --group="2" --progress=34555
Данные сохранены
PS C:\Users\lizeq\PycharmProjects\pythonProject12> python individual_1.py display data.json
Нет ошибок валидации
+-----+-----+-----+-----+
| No |      Ф.И.О.      |      Группа      |      Успеваемость      |
+-----+-----+-----+-----+
| 1 | Сидоров Сидор    | 4                | 23445                  |
| 2 | Иванов Иван      | 2                | 34555                  |
+-----+-----+-----+-----+
PS C:\Users\lizeq\PycharmProjects\pythonProject12> python individual_1.py select data.json
Нет ошибок валидации
+-----+-----+-----+-----+
| No |      Ф.И.О.      |      Группа      |      Успеваемость      |
+-----+-----+-----+-----+
| 1 | Сидоров Сидор    | 4                | 23445                  |
+-----+-----+-----+-----+
PS C:\Users\lizeq\PycharmProjects\pythonProject12>
```

Рисунок 4. Работа кода

4. Выполнил задание повышенной сложности.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import json
import 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):
    """
    Добавление нового студента
    """
    students = open_file(filename)
    students.append(
        {
            'name': name,
            'group': group,
            'progress': progress
        }
    )
    with open(filename, "w", encoding="utf-8") as out:
```

Рисунок 5. Код задания

```
PS C:\Users\lizeq\PycharmProjects\pythonProject12> python individual_2.py add data.json -n Nikolay -g 2 -p 24444
PS C:\Users\lizeq\PycharmProjects\pythonProject12> python individual_2.py display data.json
+-----+-----+-----+-----+
| № |           Ф.И.О.           | Группа | Успеваемость |
+-----+-----+-----+-----+
| 1 | Nikolay | 2 | 24444 |
+-----+-----+-----+-----+
PS C:\Users\lizeq\PycharmProjects\pythonProject12> python individual_2.py select data.json
+-----+-----+-----+-----+
| № |           Ф.И.О.           | Группа | Успеваемость |
+-----+-----+-----+-----+
| 1 | Nikolay | 2 | 24444 |
+-----+-----+-----+-----+
PS C:\Users\lizeq\PycharmProjects\pythonProject12> 
```

Рисунок 6. Работа кода

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`). Вторым способом – это модуль `getopt`, который обрабатывает как короткие, так и длинные параметры, включая оценку значений параметров.

4. Какие особенности построения 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 построения 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.

Вывод: в ходе выполнения лабораторной работы приобрел навыки построения приложений с интерфейсом командной строки с помощью языка программирования Python версии 3.x.