

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 РАБОТЕ №2.17
дисциплины «Программирование на языке Python»

Выполнил:

Хачатрян Владимир Владимирович
2 курс, группа ИТС-б-о-22-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети», очная форма обучения

(подпись)

Руководитель практики:

Воронкин Р.А., канд. тех. наук, доцент,
доцент кафедры инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: разработка приложений с интерфейсом командной строки (CLI) в Python3.

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

Выполнение работы

Изучил теоретический материал работы, создал общедоступный репозиторий на GitHub, в котором использована лицензий MIT и язык программирования Python, также добавил файл .gitignore с необходимыми правилами.

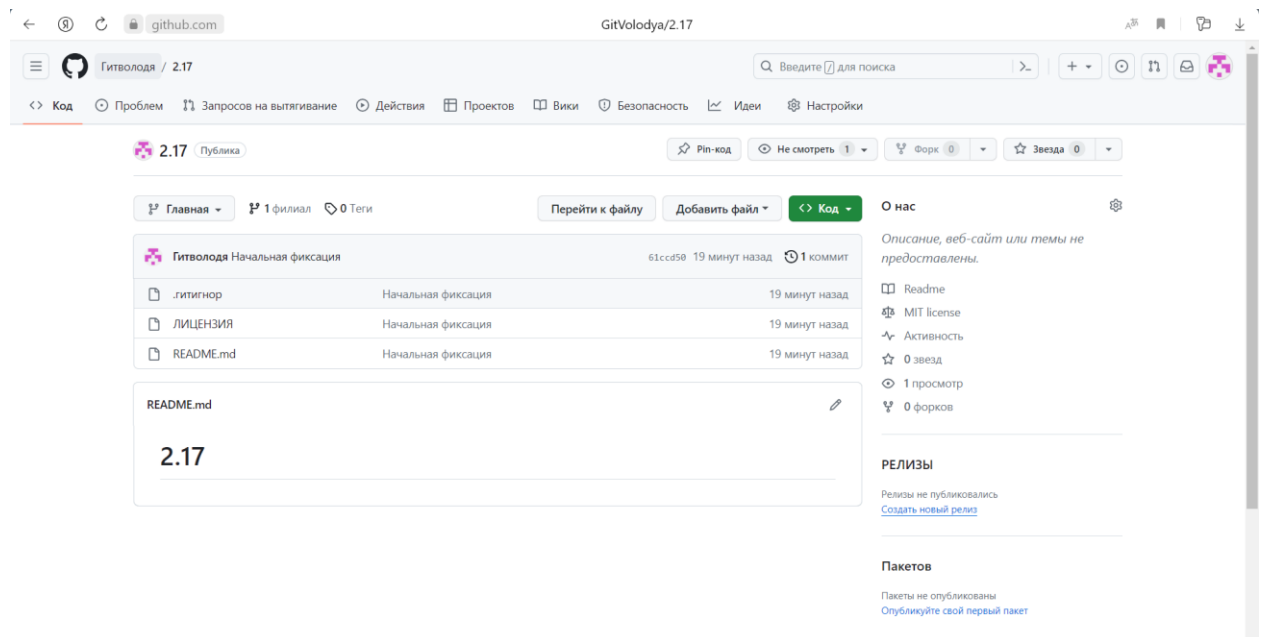


Рисунок 1. Новый репозиторий

Клонировал созданный репозиторий на компьютер.

 Git CMD

```
C:\Users\vovax>git clone https://github.com/GitVolodya/2.17.git
Cloning into '2.17'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.

C:\Users\vovax>cd C:\Users\vovax\2.17

C:\Users\vovax\2.17>
```

Рисунок 2. Клонирование

Создал виртуальное окружение (ВО) Miniconda и активировал его, также установил необходимые пакеты isort, black, flake8.

```
Anaconda Powershell Prompt (miniconda3)

(base) PS C:\Users\vovax> cd ^C
(base) PS C:\Users\vovax> cd C:\Users\vovax\2.17
(base) PS C:\Users\vovax\2.17> conda create -n 2.17 python=3.11
Retrieving notices: ...working... done
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

Рисунок 3. Создание ВО

```
(base) PS C:\Users\vovax\2.17> conda activate 2.17
(2.17) PS C:\Users\vovax\2.17> conda install -c conda-forge black
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

Рисунок 4. Установка пакета black

```
(2.17) PS C:\Users\vovax\2.17> conda install -c conda-forge flake8
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

Рисунок 5. Установка пакета flake8

```
(2.17) PS C:\Users\vovax\2.17> conda install -c conda-forge isort
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

Рисунок 6. Установка пакета isort

Пакет isort является инструментом для автоматической сортировки импортов в Python-кодах. Он используется для удобства чтения и поддержания порядка в коде.

Пакет black представляет инструмент автоматического форматирования кода для языка Python. Он помогает обеспечить единообразие стиля кодирования в проекте и улучшает читаемость кода.

Пакет flake8 отвечает за статический анализ и проверку Python-кода. Он проводит проверку на соответствие стилю кодирования PEP 8, а также наличие потенциальных ошибок и проблемных паттернов в коде.

Пример №1

Для примера 1 лабораторной работы 2.16 разработайте интерфейс командной строки.

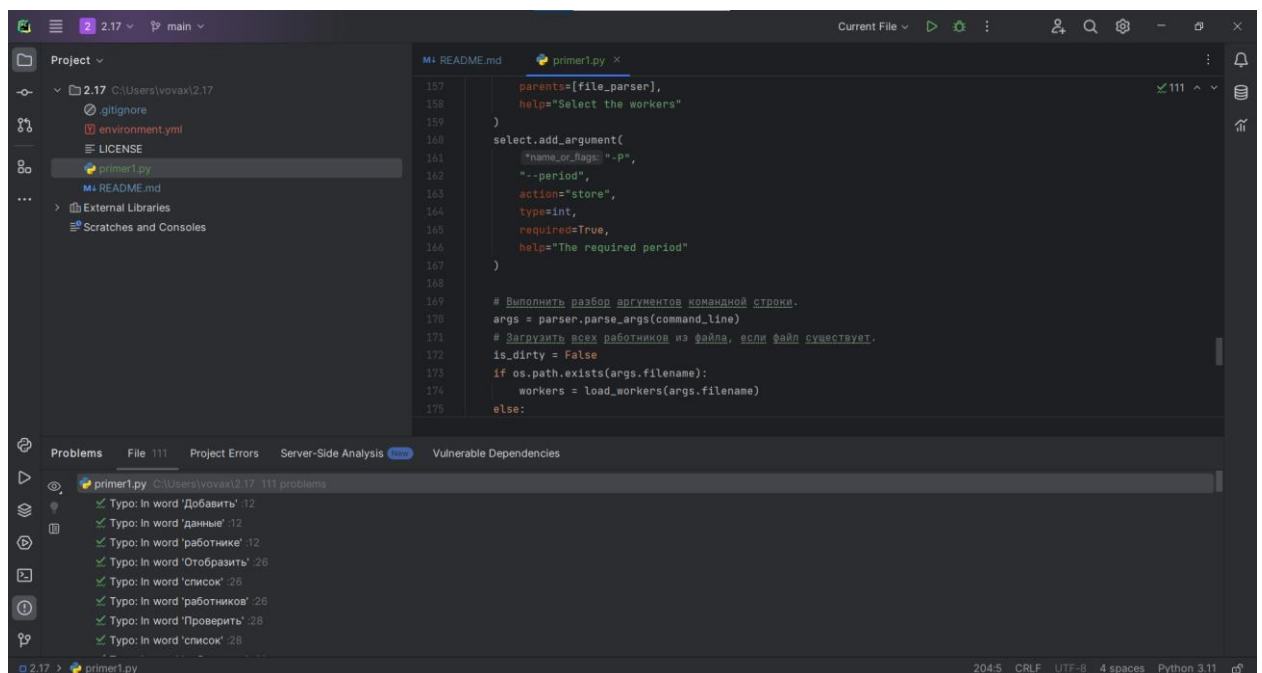


Рисунок 7. Пример 1

Индивидуальное задание

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

Условие задания: Использовать словарь, содержащий следующие ключи: фамилия и инициалы; номер группы; успеваемость (список из пяти элементов). Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть упорядочены по алфавиту; вывод на дисплей фамилий и номеров групп для всех студентов, имеющих хотя бы одну оценку 2; если таких студентов нет, вывести соответствующее сообщение. Оформить каждую команду в виде отдельной функции. Необходимо дополнительно реализовать сохранение и чтение данных из файла формата JSON. Необходимо также проследить за тем, чтобы файлы генерируемый этой программой не попадали в репозиторий лабораторной работы.

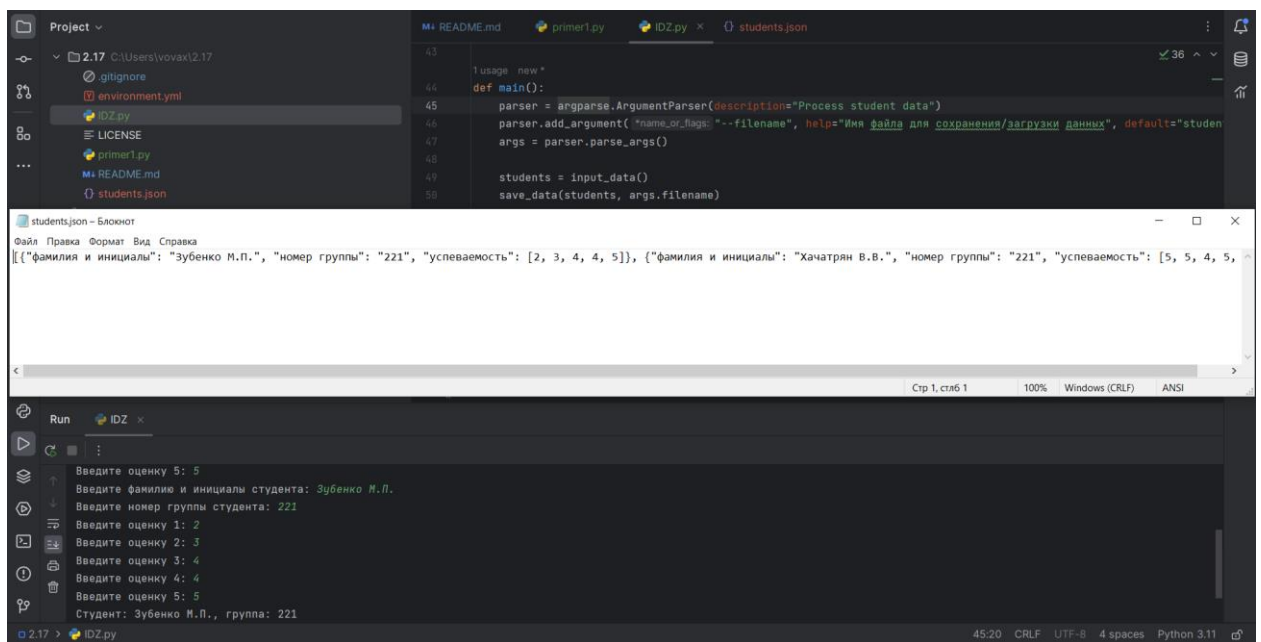


Рисунок 8. Индивидуальное задание

Слил ветку develop с веткой main. Создал файл с зависимостями и деактивировал ВО. Отправил на удаленный сервер.

Ссылка на репозиторий: <https://github.com/GitVolodya/2.17.git>

Ответы на контрольные вопросы:

1. В чем отличие терминала и консоли?

- Терминал: это аппаратное устройство или программа, которая позволяет пользователю взаимодействовать с операционной системой посредством текстового интерфейса. Терминал часто используется для ввода команд и получения вывода от операционной системы.

- Консоль: это текстовый интерфейс операционной системы, который предоставляет пользователю возможность взаимодействия с системой через командную строку. Консоль может быть запущена в терминале или в эмуляторе терминала.

2. Что такое консольное приложение?

Консольное приложение - это программа, которая работает в командной строке или консоли, взаимодействуя с пользователем через команды и выводя результаты на экран. Консольные приложения часто используются для автоматизации задач, работы с файлами, выполнения операций в системе и других операций, не требующих графического интерфейса.

3. Какие существуют средства языка программирования Python для построения приложений командной строки?

- Модуль `sys`: позволяет взаимодействовать с аргументами командной строки, стандартными потоками ввода/вывода и другими системными функциями.

- Модуль `getopt`: предоставляет функции для анализа аргументов командной строки и обработки опций и аргументов.

- Модуль `argparse`: предоставляет более гибкий и мощный способ разбора аргументов командной строки. Он позволяет определять опции, аргументы и подкоманды с различными типами данных и параметрами.

4. Какие особенности построение CLI с использованием модуля sys?

- Модуль `sys` предоставляет список аргументов командной строки через переменную `sys.argv`. Мы можем использовать этот список для анализа и обработки аргументов.

- Модуль `sys` также предлагает стандартные потоки ввода/вывода `sys.stdin`, `sys.stdout` и `sys.stderr` для работы с данными ввода и вывода.

- Модуль `sys` имеет функции для работы с интерпретатором Python, такие как `sys.exit()` для выхода из программы.

- Однако модуль `sys` предоставляет только базовые средства для построения CLI и не имеет встроенных возможностей для определения опций и аргументов.

5. *Какие особенности построение CLI с использованием модуля `getopt`?*

- Модуль `getopt` предоставляет функцию `getopt.getopt()` для разбора опций и аргументов командной строки.

- `getopt.getopt()` позволяет определить опции с короткими и длинными именами соответственно с использованием кортежа аргументов и списков кортежей параметров.

- Модуль `getopt` также обрабатывает ошибки анализа аргументов и генерирует соответствующее сообщение об ошибке.

- Модуль `getopt` не имеет поддержки подкоманд и не предоставляет возможности для работы с различными типами данных опций и аргументов.

6. *Какие особенности построение CLI с использованием модуля `argparse`?*

- Модуль `argparse` позволяет определить аргументы, опции и подкоманды с различными типами данных и параметрами.

- `argparse` автоматически обрабатывает ошибки анализа аргументов и генерирует информативные сообщения об ошибках.

- Модуль `argparse` предоставляет возможности для генерации справочной информации и использования значений аргументов и опций в коде программы.

- `argparse` позволяет гибко контролировать формат вывода и отображение информации о программе и подкомандах.

- Модуль `argparse` предлагает функциональность для работы с произвольными типами данных аргументов и опций, включая проверку вводимых данных.

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