

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 «Основы кроссплатформенного программирования»

Выполнил:

Баканов Артем Вадимович

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: работа с данными формата JSON в языке Python.

Цель работы: приобретение навыков по работе с данными формата JSON с помощью языка программирования Python версии 3.x.

Порядок выполнения работы:

Задание 1. 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.

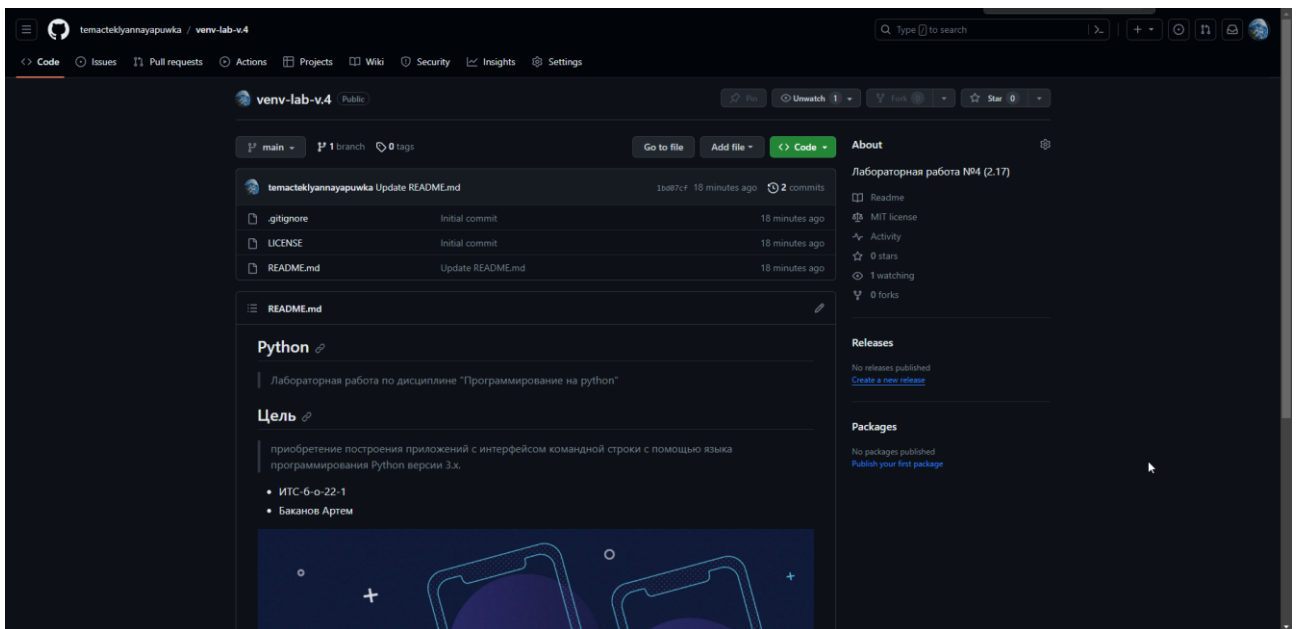


Рисунок 1. Создан новый репозиторий

Клонировал репозиторий на свой компьютер. В ходе данной лабораторной работы работала с моделью ветвления git-flow.

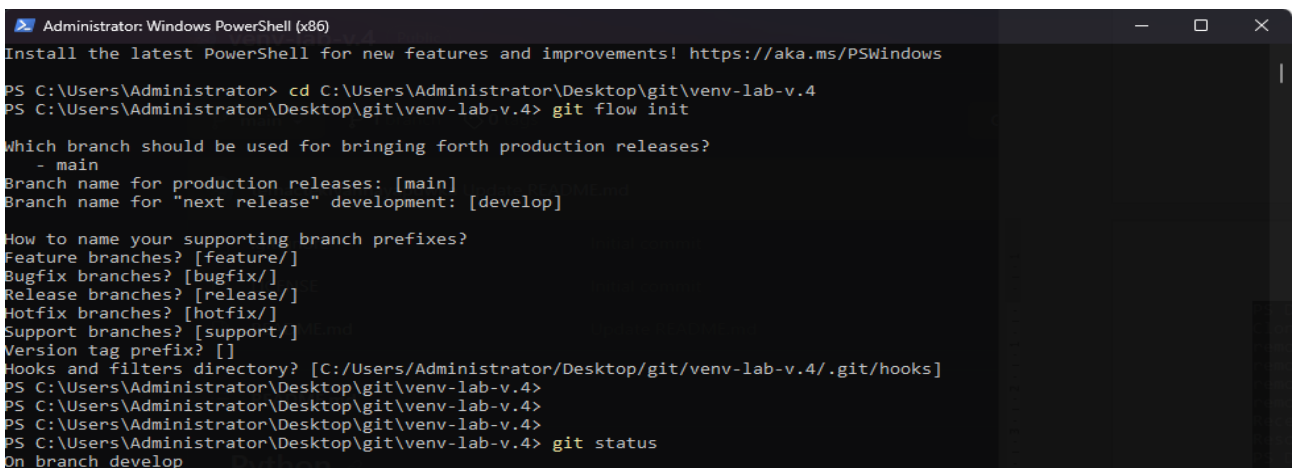


Рисунок 2. Клонирование и модель ветвления git-flow

Создал виртуальное окружение Anaconda

```
(base) PS C:\Users\Administrator> cd C:\Users\Administrator\Desktop\git\venv-lab-v.4
(base) PS C:\Users\Administrator\Desktop\git\venv-lab-v.4> conda create -n 2.17
Retrieving notices: ...working... done
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 23.5.2
  latest version: 23.10.0
Please update conda by running

  $ conda update -n base -c defaults conda

Or to minimize the number of packages updated during conda update use

  conda install conda=23.10.0

## Package Plan ##

  environment location: C:\Users\Administrator\miniconda3\envs\2.17

Proceed ([y]/n)? y
Preparing transaction: done
```

Рисунок 3. Создание

Рисунок 3. Создание виртуального окружения

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

```
# $ conda deactivate

(base) PS C:\Users\Administrator\Desktop\git\venv-lab-v.4> conda activate 2.17
(2.17) PS C:\Users\Administrator\Desktop\git\venv-lab-v.4> conda install isort
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 23.5.2
  latest version: 23.10.0
Please update conda by running

  $ conda update -n base -c defaults conda

Or to minimize the number of packages updated during conda update use

  conda install conda=23.10.0

## Package Plan ##

  environment location: C:\Users\Administrator\miniconda3\envs\2.17

  added / updated specs:
    - isort
```

Цель

- × ИТОГОВЫЙ
- × БАЗОВЫЙ АРХИВ

Рисунок 4. Установка пакета isort

```

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
(2.17) PS C:\Users\Administrator\Desktop\git\venv-lab-v.4> conda install flake8
WARNING conda.core.prefix_data:_load_single_record(215): Ignoring malformed prefix record at: C:\Users\Administrator\miniconda3\envs\2.17\conda-meta\wheel-0.41.2-py312h95532_0.tar.json
Collecting package metadata (current_repodata.json): done
Solving environment: |
The environment is inconsistent, please check the package plan carefully
The following packages are causing the inconsistency:

- defaults/noarch::isort==5.9.3=pyhd3eb1b0_0
- defaults/win-64::pip==23.3=py312h95532_0
- defaults/win-64::python==3.12.0=h1d929f7_0
- defaults/win-64::setuptools==68.0.0=py312h95532_0
done

==> WARNING: A newer version of conda exists. <==
  current version: 23.5.2
  latest version: 23.10.0

Please update conda by running

  $ conda update -n base -c defaults conda

Or to minimize the number of packages updated during conda update use

  conda install conda=23.10.0

```

Рисунок 5. Установка пакета flake8

```

Downloading and Extracting Packages
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
(2.17) PS C:\Users\Administrator\Desktop\git\venv-lab-v.4> conda install black
WARNING conda.core.prefix_data:_load_single_record(215): Ignoring malformed prefix record at: C:\Users\Administrator\miniconda3\envs\2.17\conda-meta\importlib-metadata-6.0.0-py312h95532_0.tar.json
WARNING conda.core.prefix_data:_load_single_record(215): Ignoring malformed prefix record at: C:\Users\Administrator\miniconda3\envs\2.17\conda-meta\wheel-0.41.2-py312h95532_0.tar.json
WARNING conda.core.prefix_data:_load_single_record(215): Ignoring malformed prefix record at: C:\Users\Administrator\miniconda3\envs\2.17\conda-meta\zipp-3.11.0-py312h95532_0.tar.json
Collecting package metadata (current_repodata.json): done
Solving environment: |
The environment is inconsistent, please check the package plan carefully
The following packages are causing the inconsistency:

- defaults/noarch::flake8==4.0.1=pyhd3eb1b0_1
- defaults/noarch::isort==5.9.3=pyhd3eb1b0_0
- defaults/noarch::mccabe==0.7.0=pyhd3eb1b0_0
- defaults/win-64::pip==23.3=py312h95532_0
- defaults/noarch::pycodestyle==2.8.0=pyhd3eb1b0_0
- defaults/noarch::pyflakes==2.4.0=pyhd3eb1b0_0
- defaults/win-64::python==3.12.0=h1d929f7_0
- defaults/win-64::setuptools==68.0.0=py312h95532_0
done

==> WARNING: A newer version of conda exists. <==
  current version: 23.5.2

```

Рисунок 6. Установка пакета black

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.

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

Работа с примером №1.

Для примера 1 лабораторной работы 2.16 разработайте интерфейс командной строки.

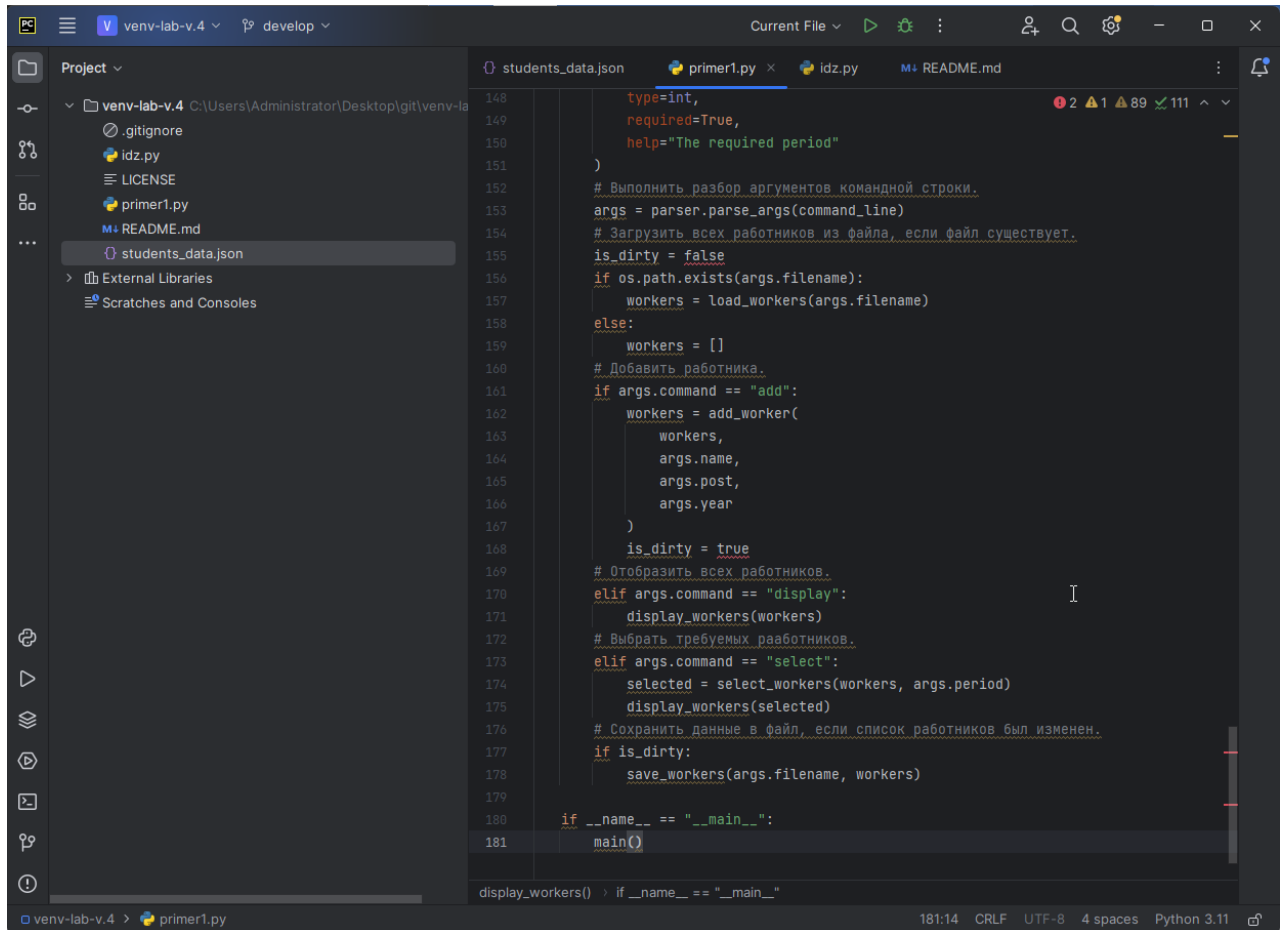


Рисунок 7. Работа с примером №1.

Выполнение индивидуального задания.

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.

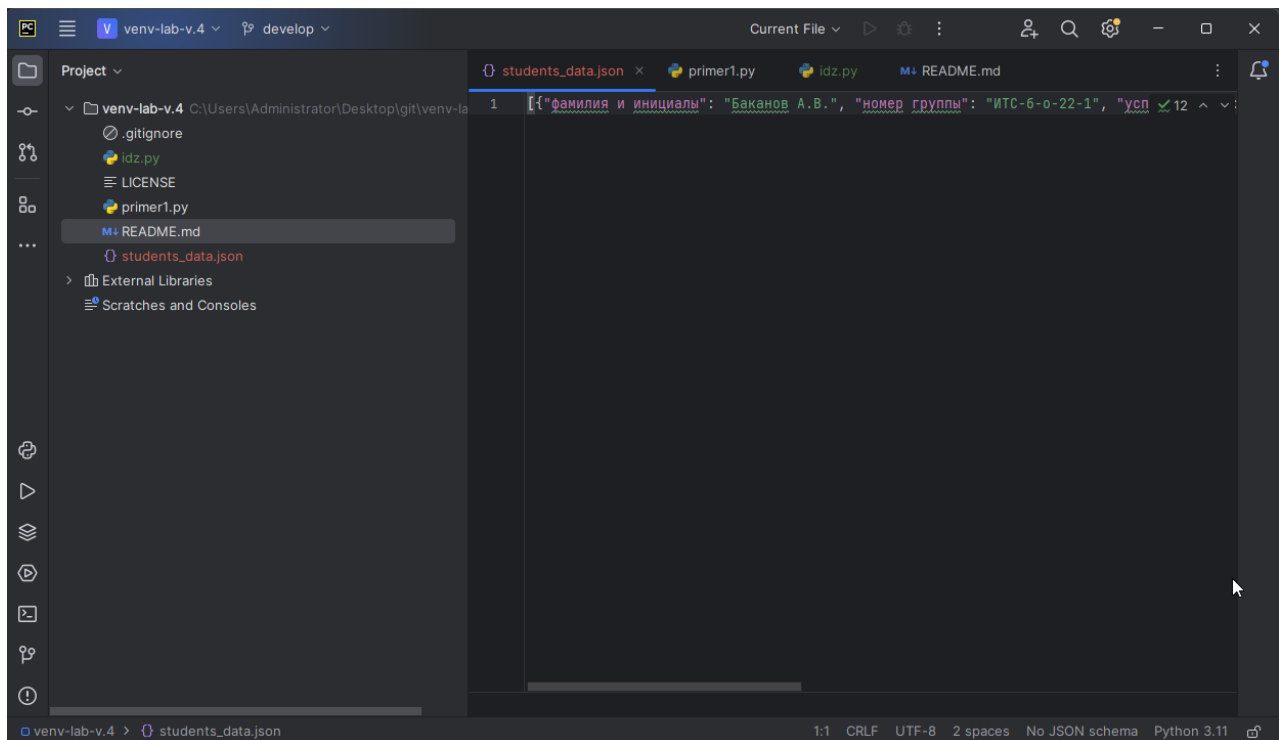


Рисунок 8. Выполнение индивидуального задания

Деактивировал виртуальное окружение.

```
(2.17) PS C:\Users\Administrator\Desktop\git\venv-lab-v.4> pip freeze > requirements.txt
(2.17) PS C:\Users\Administrator\Desktop\git\venv-lab-v.4> conda env export > environmenr.yml
WARNING conda.core.prefix_data:load_single_record(215): Ignoring malformed prefix record at: C:\Users\Administrator\mi
i conda3\envs\2.17\conda-meta\black-23.3.0-py312haa95532_0.tar.json
WARNING conda.core.prefix_data:load_single_record(215): Ignoring malformed prefix record at: C:\Users\Administrator\mi
i conda3\envs\2.17\conda-meta\click-8.1.7-py312haa95532_0.tar.json
WARNING conda.core.prefix_data:load_single_record(215): Ignoring malformed prefix record at: C:\Users\Administrator\mi
i conda3\envs\2.17\conda-meta\colorama-0.4.6-py312haa95532_0.tar.json
WARNING conda.core.prefix_data:load_single_record(215): Ignoring malformed prefix record at: C:\Users\Administrator\mi
i conda3\envs\2.17\conda-meta\importlib-metadata-6.0.0-py312haa95532_0.tar.json
WARNING conda.core.prefix_data:load_single_record(215): Ignoring malformed prefix record at: C:\Users\Administrator\mi
i conda3\envs\2.17\conda-meta\mypy_extensions-1.0.0-py312haa95532_0.tar.json
WARNING conda.core.prefix_data:load_single_record(215): Ignoring malformed prefix record at: C:\Users\Administrator\mi
i conda3\envs\2.17\conda-meta\packaging-23.1-py312haa95532_0.tar.json
WARNING conda.core.prefix_data:load_single_record(215): Ignoring malformed prefix record at: C:\Users\Administrator\mi
i conda3\envs\2.17\conda-meta\pathspec-0.10.3-py312haa95532_0.tar.json
WARNING conda.core.prefix_data:load_single_record(215): Ignoring malformed prefix record at: C:\Users\Administrator\mi
i conda3\envs\2.17\conda-meta\wheel-0.41.2-py312haa95532_0.tar.json
WARNING conda.core.prefix_data:load_single_record(215): Ignoring malformed prefix record at: C:\Users\Administrator\mi
i conda3\envs\2.17\conda-meta\zipp-3.11.0-py312haa95532_0.tar.json
(2.17) PS C:\Users\Administrator\Desktop\git\venv-lab-v.4> conda deactivate
(base) PS C:\Users\Administrator\Desktop\git\venv-lab-v.4>
```

Рисунок 9. Деактивация ВО

Слил ветку develop с веткой main и отправил на удаленный сервер

```
Your branch is up to date with 'origin/main'.
PS C:\Users\Administrator\Desktop\git\venv-lab-v.4> git merge develop
Updating 1bd07cf..7be93f5
Fast-forward
 .idea/.gitignore          | 3 +
 .idea/inspectionProfiles/profiles_settings.xml | 6 +
 .idea/misc.xml            | 4 +
 .idea/modules.xml         | 8 ++
 .idea/vcs.xml             | 6 +
 .idea/venv-lab-v.4.iml    | 8 ++
 environmenr.yml          | Bin 0 -> 2024 bytes
 idz.py                   | 77 ++++++++
 primer1.py               | 181 ++++++++
 requirements.txt         | Bin 0 -> 2544 bytes
10 files changed, 293 insertions(+)
create mode 100644 .idea/.gitignore
create mode 100644 .idea/inspectionProfiles/profiles_settings.xml
create mode 100644 .idea/misc.xml
create mode 100644 .idea/modules.xml
create mode 100644 .idea/vcs.xml
create mode 100644 .idea/venv-lab-v.4.iml
create mode 100644 environmenr.yml
create mode 100644 idz.py
create mode 100644 primer1.py
create mode 100644 requirements.txt
PS C:\Users\Administrator\Desktop\git\venv-lab-v.4>
```

Рисунок 8. Слияние веток

Ссылка на репозиторий: <https://github.com/temacteklyannayapuwka/venv-lab-v.4>

Ответы на контрольные вопросы

1. В чем отличие терминала и консоли?

Терминал (Terminal):

В общем смысле, терминал - это программа, предоставляющая текстовый интерфейс для взаимодействия с операционной системой. Это может быть командная строка в графической среде, также известная как терминал в Unix-подобных системах.

В контексте Python, "терминал" может означать окно командной строки, в котором вы запускаете скрипты Python.

Консоль (Console):

В Python термин "консоль" часто используется для обозначения интерактивной оболочки Python (REPL - Read-Eval-Print Loop), где вы можете вводить команды Python непосредственно и видеть результаты выполнения.

В графической среде Windows, "консоль" может также относиться к окну командной строки (Command Prompt) или PowerShell.

2. Что такое консольное приложение?

Консольное приложение (или текстовое приложение) — это приложение, взаимодействие с пользователем которого осуществляется через текстовый интерфейс командной строки. В отличие от графических приложений, консольные приложения не используют графический пользовательский интерфейс (GUI) и обычно работают в текстовом режиме.

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

Модуль `argparse`: Этот модуль предоставляет инструменты для анализа аргументов командной строки. Он позволяет определять, какие аргументы ожидаются при запуске программы и как они должны обрабатываться.

Модуль `sys`: Модуль `sys` предоставляет доступ к некоторым переменным и функциям, связанным с интерпретатором Python. Например, `sys.argv` содержит список аргументов командной строки, переданных скрипту.

Модуль `click`: Это библиотека, которая облегчает создание красивых и удобных в использовании интерфейсов командной строки.

Модуль `subprocess`: Этот модуль позволяет запускать другие программы из Python и взаимодействовать с ними. Он может быть полезен для выполнения внешних команд из командной строки.

4. Какие особенности построение CLI с использованием модуля `sys` ?

Построение интерфейса командной строки (CLI) с использованием модуля `sys` включает в себя работу с аргументами командной строки, переданными скрипту. Основные шаги включают в себя:

Импорт модуля `sys`:

Обработка аргументов: Аргументы командной строки доступны в списке `sys.argv`. Этот список содержит имя скрипта (индекс 0) и все переданные аргументы.

Обработка флагов и значений: Модуль `sys` не предоставляет специальных инструментов для обработки флагов и значений.

Запуск из командной строки: Ваш скрипт может быть запущен из командной строки с аргументами

5. Какие особенности построение CLI с использованием модуля getopt ?

Модуль getopt - это более старый и менее удобный способ обработки аргументов командной строки по сравнению с более современным модулем argparse. Однако, если вам нужна простая и легкая в использовании альтернатива, getopt может быть полезным.

1. Импорт модуля getopt
2. Определение параметров командной строки
3. Обработка ошибок
4. Запуск из командной строки

6. Какие особенности построение CLI с использованием модуля argparse

Модуль argparse предоставляет более мощные и удобные средства для построения интерфейса командной строки (CLI) по сравнению с getopt или простым использованием sys.argv. Вот основные особенности построения CLI с использованием модуля argparse:

1. Импорт модуля argparse
2. Создание объекта парсера
3. Добавление аргументов
4. Парсинг аргументов
5. Использование аргументов
6. Запуск из командной строки
7. Поддержка справки и документации

Вывод: приобрел навыки построения приложений с интерфейсом командной строки с помощью языка программирования Python версии 3.x.