

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 ЛАБОРАТОРНОЙ РАБОТЕ №2.17
дисциплины «Основы кроссплатформенного программирования»

Выполнил:
Евдаков Евгений 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., доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

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

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

Ход работы:

Задание 1. Создал общедоступный репозиторий на GitHub, в котором использована лицензий MIT и язык программирования Python, также добавил файл .gitignore с необходимыми правилами. Клонировал свой репозиторий на свой компьютер.

```
C:\Users\Gaming-PC>git clone https://github.com/EvgenyEvdakov/Laba_2.17.git
Cloning into 'Laba_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.
```

Рисунок 1. Клонирование репозитория

Задание 2. Организовал свой репозиторий в соответствии с моделью ветвления git-flow, появилась новая ветка develop в которой буду выполнять дальнейшие задачи.

```
C:\Users\Gaming-PC\Laba_2.17>git flow init
0 [sh] sh 1426 dofork: child -1 - forked process 19768 died unexpectedly, retry 0, exit code 0xc0000005, errno 11
C:/Program Files/Git/usr/bin/gitflow-shFlags: fork: retry: Resource temporarily unavailable
Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]
How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/Gaming-PC/Laba_2.17/.git/hooks]
```

Рисунок 2. Модель ветвления git-flow

Задание 3. Создал виртуальное окружение conda и активировал его, также установил необходимые пакеты isort, black, flake8.

```

(base) PS C:\Users\Gaming-PC> cd C:\Users\Gaming-PC\Laba_2.17
(base) PS C:\Users\Gaming-PC\Laba_2.17> conda create -n 2.17 python=3.7
Retrieving notices: ...working... done
Collecting package metadata (current_repodata.json): done
Solving environment: failed with repodata from current_repodata.json, will retry with next repodata source.
Collecting package metadata (repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 23.1.0
  latest version: 23.9.0

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=23.9.0

## Package Plan ##

environment location: C:\Users\Gaming-PC\.conda\envs\2.17

added / updated specs:
- python=3.7

The following packages will be downloaded:



| package        | build      |        |
|----------------|------------|--------|
| openssl-1.1.1w | h2bbff1b_0 | 5.5 MB |
| Total:         |            | 5.5 MB |


```

Рисунок 3. Создание виртуального окружения

```

(base) PS C:\Users\Gaming-PC\Laba_2.17> conda install -c conda-forge black
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 23.1.0
  latest version: 23.9.0

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=23.9.0

```

Рисунок 4. Добавление пакета black

```
(base) PS C:\Users\Gaming-PC\Laba_2.17> conda install -c conda-forge flace8
Collecting package metadata (current_repodata.json): done
Solving environment: failed with initial frozen solve. Retrying with flexible solve.
Collecting package metadata (repodata.json): done
Solving environment: failed with initial frozen solve. Retrying with flexible solve.

PackagesNotFoundError: The following packages are not available from current channels:

- flace8

Current channels:

- https://conda.anaconda.org/conda-forge/win-64
- https://conda.anaconda.org/conda-forge/noarch
- https://repo.anaconda.com/pkgs/main/win-64
- https://repo.anaconda.com/pkgs/main/noarch
- https://repo.anaconda.com/pkgs/r/win-64
- https://repo.anaconda.com/pkgs/r/noarch
- https://repo.anaconda.com/pkgs/msys2/win-64
- https://repo.anaconda.com/pkgs/msys2/noarch

To search for alternate channels that may provide the conda package you're
looking for, navigate to

    https://anaconda.org

and use the search bar at the top of the page.
```

Рисунок 5. Добавление пакета flace8

```
(base) PS C:\Users\Gaming-PC\Laba_2.17> conda install -c conda-forge isort
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
    current version: 23.1.0
    latest version: 23.9.0

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=23.9.0
```

Рисунок 6. Добавление пакета isort

Задание 4. Создал проект PyCharm в папке репозитория. Приступил к работе с примером. Добавил новый файл primer1.py.

Условие примера:

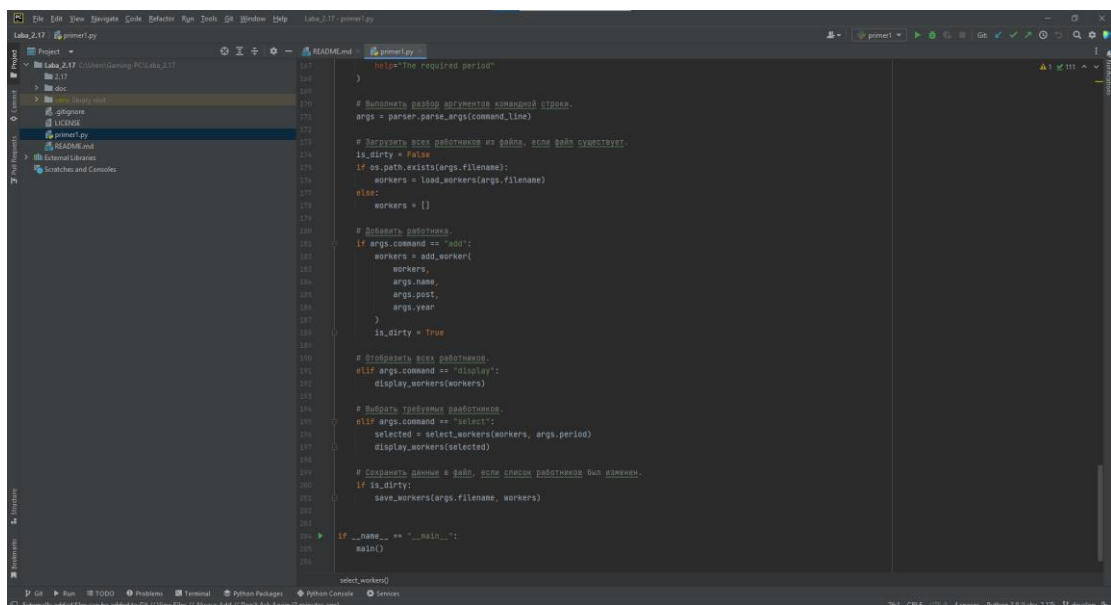


Рисунок 7. Пример 1

Задание 5.

Индивидуальное задание

Вариант 10

Создал новый файл под названием idz.py.

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

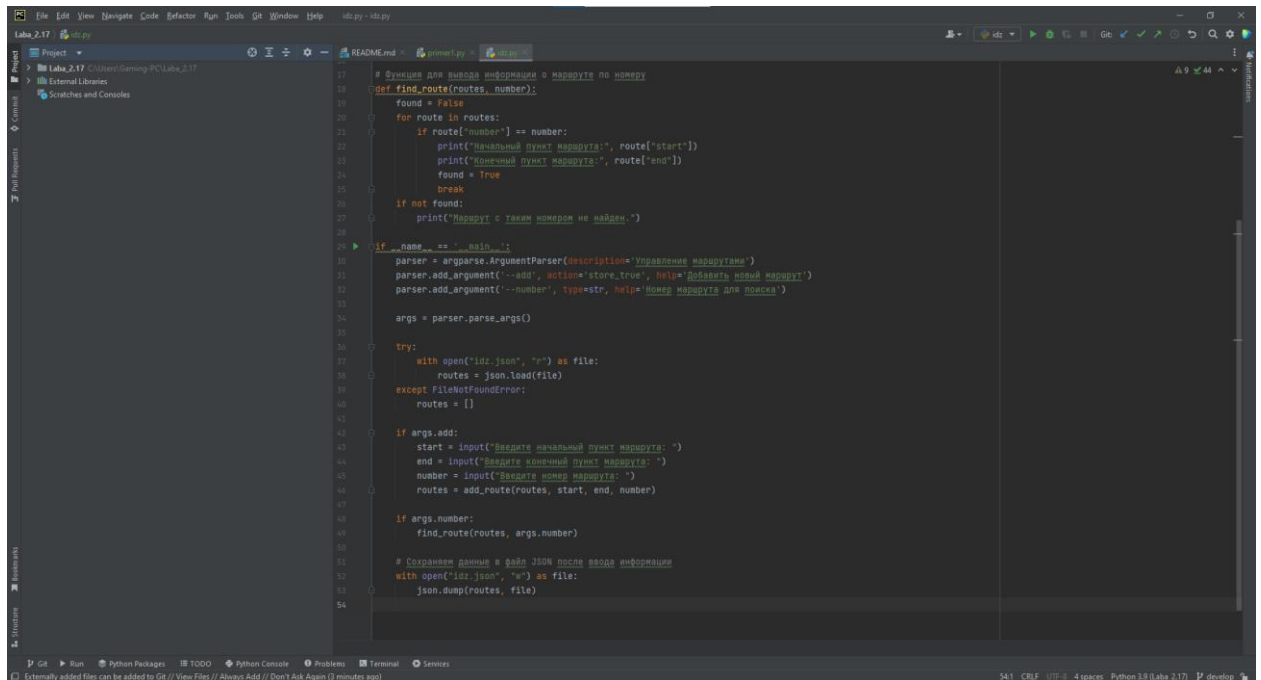


Рисунок 8. Код программы

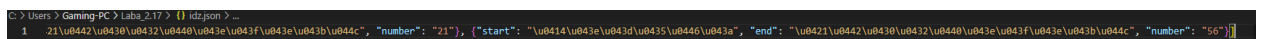


Рисунок 9. Содержимое json файла

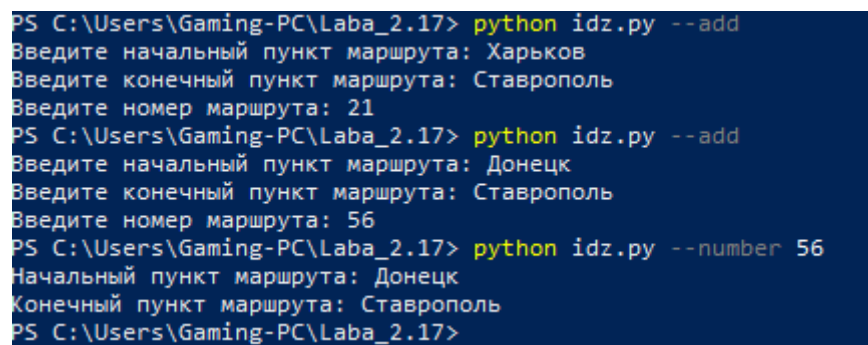


Рисунок 10. Реализация программы

Задание 6.

После выполнения работы на ветке develop, слил ее с веткой main и отправил изменения на удаленный сервер. Создал файл envirement.yml и деактивировал виртуальное окружение.

```
(base) PS C:\Users\Gaming-PC\Laba_2.17> conda env export > envirement.yml
(base) PS C:\Users\Gaming-PC\Laba_2.17> conda deactivate
```

Рисунок 11. Деактивация виртуального окружения

Ссылка: https://github.com/EvgenyEvdakov/Laba_2.17

Ответы на контрольные вопросы:

1. В чем отличие терминала и консоли?

Терминал - это программное обеспечение или устройство, позволяющее пользователю взаимодействовать с операционной системой. Он предоставляет текстовый интерфейс для ввода команд и получения вывода. Обычно, терминал используется для запуска команд и управления системой.

Консоль - это окно, где пользователь может вводить команды, запускать приложения и видеть вывод этих программ. Это графическое представление терминала.

2. Что такое консольное приложение?

Консольное приложение - это программа, предназначенная для выполнения в командной строке (консоли). Такие приложения обрабатывают ввод пользователя и выводят результат на консоль без графического интерфейса.

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

В Python существует несколько встроенных модулей для создания CLI-приложений: `sys`, `argparse`, `getopt` и другие.

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

Модуль `sys` предоставляет доступ к некоторым переменным и функциям, связанным с интерпретатором Python. Он обеспечивает доступ к аргументам командной строки через `sys.argv`, что позволяет обрабатывать аргументы при запуске скрипта.

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

Модуль `getopt` предоставляет функции для парсинга аргументов командной строки. Он позволяет более гибко управлять аргументами командной строки и их опциями.

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

Модуль `argparse` является более мощным и гибким инструментом для создания интерфейса командной строки в Python. Он позволяет определять аргументы, их типы, флаги и даже создавать справочную информацию для пользователей. `argparse` автоматически генерирует справку о том, как использовать ваше CLI-приложение.

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