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 №2.18**

**дисциплины**

**«Основы кроссплатформенного программирования»**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Пинаева Диана Олеговна  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) | |
|  | | Проверил:  Воронкин Р. А., доцент кафедры инфокоммуникаций  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ставрополь, 2023 г.

**Тема:** Работа с переменными окружения в Python3.

**Цель:** приобретение навыков по работе с переменными окружения с помощью языка программирования Python версии 3.x.

# Ход работы:

Задание 1.Создала общедоступный репозиторий на GitHub, в котором использована лицензий MIT и язык программирования Python, также добавила файл .gitignore с необходимыми правилами. Клонировала свой репозиторий на свой компьютер. Организовала свой репозиторий в соответствие с моделью ветвления git-flow, появилась новая ветка develop в которой буду выполнять дальнейшие задачи.

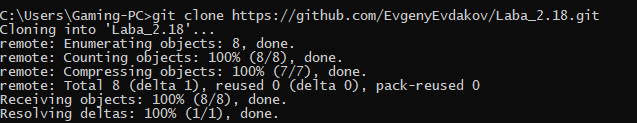


Рисунок 1. Клонирование репозитория

Задание 2.Создала виртуальное окружение conda и активировала его, также установила необходимые пакеты isort, black, flake8, pyinputplus.

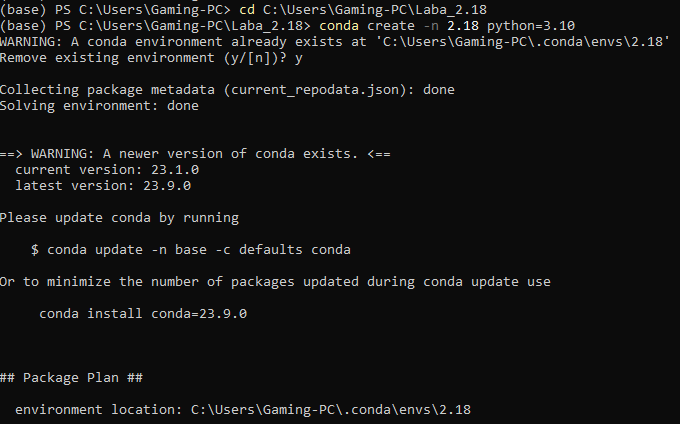


Рисунок 2. Создание виртуального окружения

Задание 3.Создала проект PyCharm в папке репозитория. Приступила к работе с примером. Добавила новый файл primer1.py.

Условие примера: Для примера 1 лабораторной работы 2.17 добавьте возможность получения имени файла данных, используя соответствующую переменную окружения.

Для хранения имени файла данных будем использовать переменную окружения WORKERS\_DATA. При этом сохраним возможность передавать имя файла данных через именной параметр --data. Иными словами, если при запуске программы в командной строке не задан параметр --data, то имя файла данных должно быть взято из переменной окружения WORKERS\_DATA.

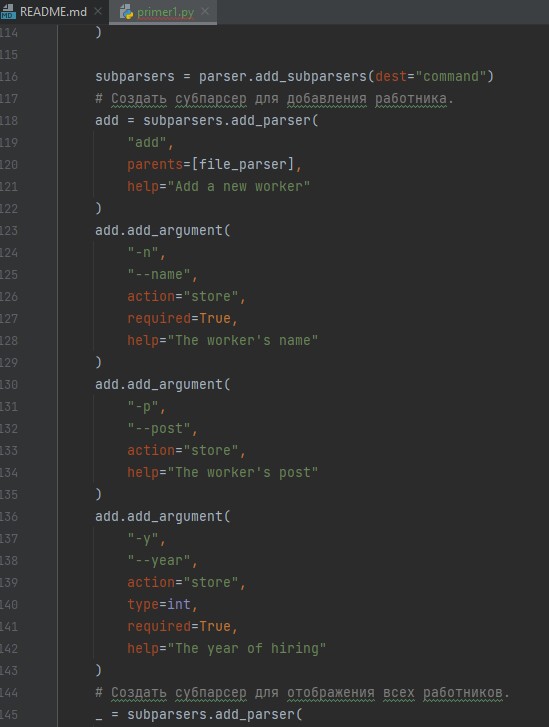


Рисунок 3. Пример 1

**Индивидуальное задание**

**Вариант 10**

Создала новый файл под названием idz.py.

Условие задания:Для своего варианта лабораторной работы 2.17 добавьте возможность получения имени файла данных, используя соответствующую переменную окружения.

Для начала необходимо создать переменное окружение:

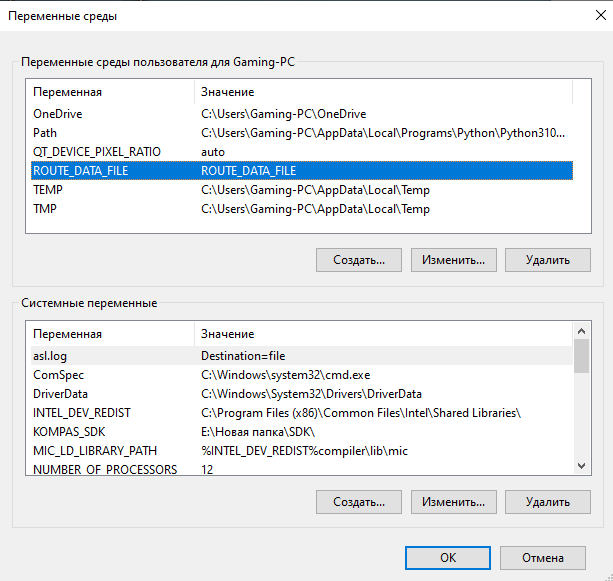


Рисунок 4. Создание переменного окружения

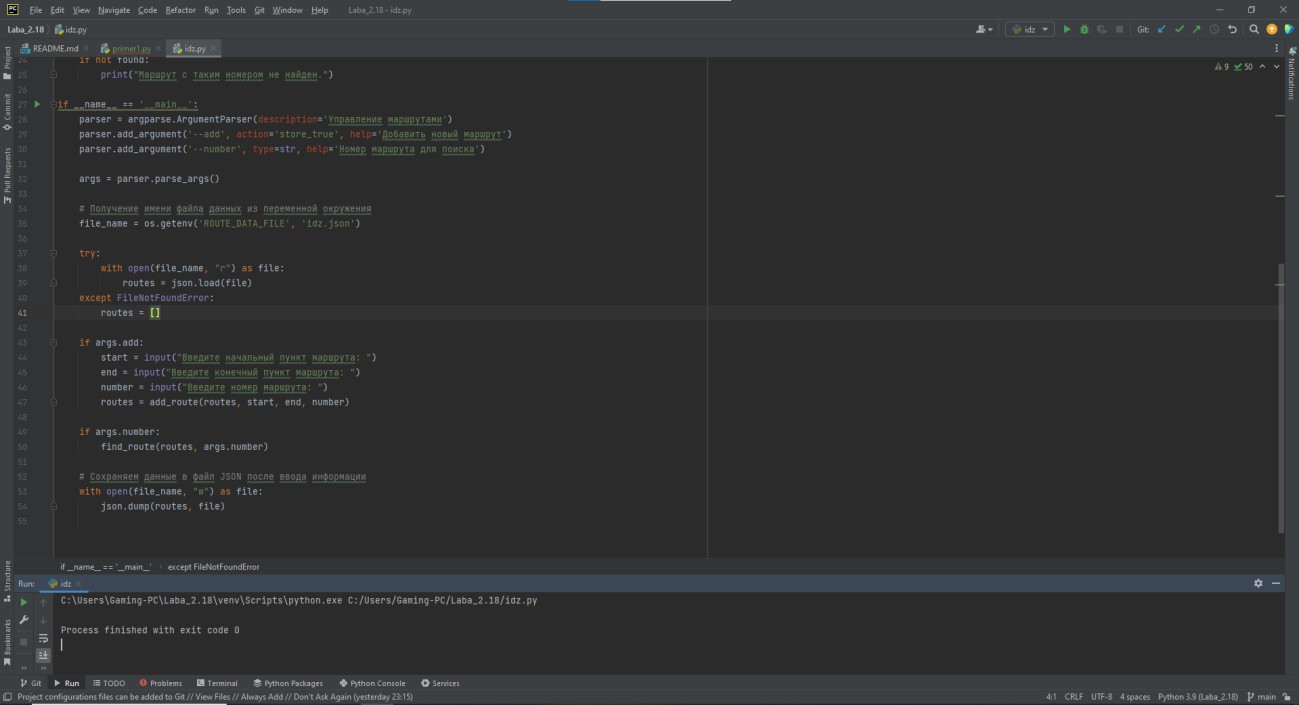


Рисунок 5. Код индивидуального задания

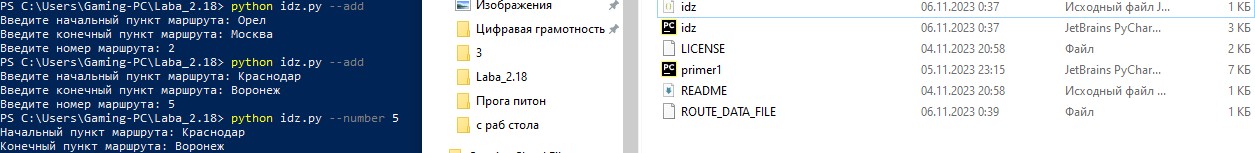


Рисунок 6. Результат индивидуального задания

В результате выполнения кода, если переменная окружения ROUTE\_DATA\_FILE установлена, она будет использоваться в качестве имени файла данных. Если переменная окружения не установлена, будет использоваться значение по умолчанию "data.json" (в данном случае установлена).

Создала файл idz2.py

Условия задания: Самостоятельно изучите работу с пакетом python- dotenv. Модифицируйте программу задания 1 таким образом, чтобы значения необходимых переменных окружения считывались из файла .env.

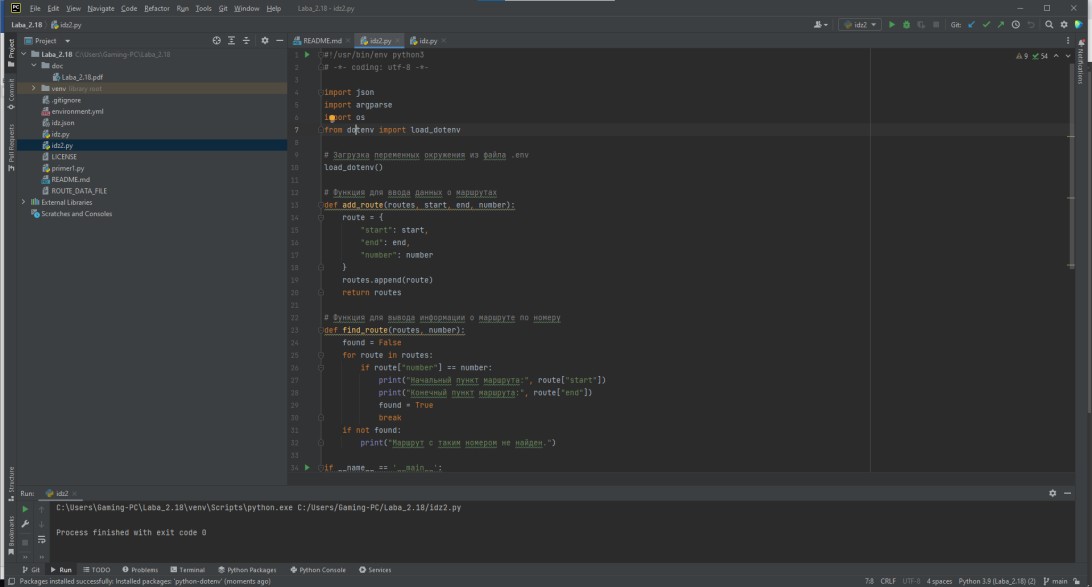


Рисунок 7. Выполнение второго индивидуального задания

# Задание 4.

После выполнения работы на ветке develop, слила ее с веткой main и отправила изменения на удаленный сервер. Создала файл envirement.yml и деактивировала виртуальное окружение.



Рисунок 7. Деактивация ВО

Ссылка: <https://github.com/Diana-372/Lab2.18.git>

# Ответы на контрольные вопросы:

1. Каково назначение переменных окружения?

Переменные окружения используются для хранения информации, доступной для всех процессов, запущенных в операционной системе. Они предоставляют программам и системе информацию о конфигурации, путях поиска файлов, доступе к ресурсам, языковых настройках и многом другом.

# Какая информация может храниться в переменных окружения?

* Пути к исполняемым файлам (например, переменная PATH).
* Конфигурационные настройки программ.
* Языковые параметры (например, LANG, LC\_ALL).
* Данные о временных директориях, пользователях и системе.
* Параметры, управляющие поведением операционной системы и программ.

# Как получить доступ к переменным окружения в ОС Windows?

* Для получения доступа к переменным окружения в Windows можно использовать команду echo %VARIABLE\_NAME% в командной строке, где VARIABLE\_NAME - имя переменной.
* В окне "Свойства системы" можно просмотреть и изменить переменные окружения через панель управления.

# Каково назначение переменных PATH и PATHEXT?

* PATH: Переменная, хранящая пути к исполняемым файлам. Она определяет, где операционная система будет искать исполняемые файлы, когда команда вводится в командной строке.
* PATHEXT: Список расширений файлов, который интерпретируется как исполняемые файлы в Windows.

# Как создать или изменить переменную окружения в Windows?

Для создания или изменения переменной окружения в Windows можно использовать "Свойства системы" -> "Дополнительные параметры системы" -

* "Переменные окружения". Можно добавить новую переменную или изменить значение существующей.

# Что представляют собой переменные окружения в ОС LInux?

В Linux переменные окружения представляют собой параметры, хранящиеся в системе, доступные для всех процессов. Они определяют окружение, в котором запускаются процессы, включая пути поиска, языковые настройки и другие параметры.

# В чем отличие переменных окружения от переменных оболочки?

* + Переменные оболочки (shell variables) - это переменные, специфичные для конкретной оболочки и доступные только для этой оболочки.
  + Переменные окружения (environment variables) - это переменные, доступные для всех процессов, запущенных в операционной системе, их значения наследуются от родительских процессов.

# Как вывести значение переменной окружения в Linux?

В командной строке Linux можно использовать команду echo

$VARIABLE\_NAME, где VARIABLE\_NAME - имя переменной.

# Какие переменные окружения Linux Вам известны?

PATH, HOME, USER, LANG, SHELL, PWD и другие.

# Какие переменные оболочки Linux Вам известны?

PS1, PS2, HISTSIZE, HISTFILE и другие, специфичные для определенных оболочек (например, BASH, Zsh).

# Как установить переменные оболочки в Linux?

Для установки переменных оболочки в Linux используются команды экспорта переменной с ключевым словом export (например, export VARIABLE\_NAME=value).

# Как установить переменные окружения в Linux?

Переменные окружения устанавливаются в Linux также, как и переменные оболочки, но они будут доступны для всех процессов. Эти переменные часто устанавливаются в файлах конфигурации системы, таких как .bashrc, .bash\_profile, /etc/environment, и т. д.

# Для чего необходимо делать переменные окружения Linux постоянными?

Переменные окружения могут быть установлены постоянно, добавив их в файлы инициализации оболочки, такие как .bashrc или .bash\_profile в домашнем каталоге пользователя.

# Для чего используется переменная окружения PYTHONHOME?

PYTHONHOME - это переменная окружения Python, которая определяет базовый каталог установки Python.

# Для чего используется переменная окружения PYTHONPATH?

PYTHONPATH - это переменная окружения Python, определяющая пути поиска Python для модулей.

# Какие еще переменные окружения используются для управления работой интерпретатора Python?

PYTHONHOME, PYTHONPATH, PYTHONSTARTUP, PYTHONCASEOK, PYTHONIOENCODING и другие.

# Как осуществляется чтение переменных окружения в программах на языке программирования Python?

В Python переменные окружения можно читать с помощью модуля os с функцией os.getenv().

# Как проверить, установлено или нет значение переменной окружения в программах на языке программирования Python?

Для проверки установленного значения переменной окружения в Python используйте функцию os.getenv('VARIABLE\_NAME').

# Как присвоить значение переменной окружения в программах на языке программирования Python?

Для установки значения переменной окружения в Python используйте

`os.putenv('VARIABLE

**Вывод:** приобрела навыки по работе с переменными окружения с помощью языка программирования Python версии 3.x.