# РОССИЙСКОЙ ФЕДЕРАЦИИ

**Федеральное государственное автономное образовательное учреждение высшего образования**

# «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

**Кафедра инфокоммуникаций**

# «Работа с переменными окружения в Python3»

**Отчет по лабораторной работе № 2.18**

# по дисциплине «Программирование на Python»

|  |
| --- |
| Выполнил студент группы ИВТ-б-о-21-1 |
| Урусов Максим Андреевич.  «19» декабря 2022г. |
| Подпись студента |
| Работа защищена « » 20 г. |
| Проверил Воронкин Р.А.  (подпись) |

Ставрополь 2022

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

# Порядок выполнения работы:

1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python.

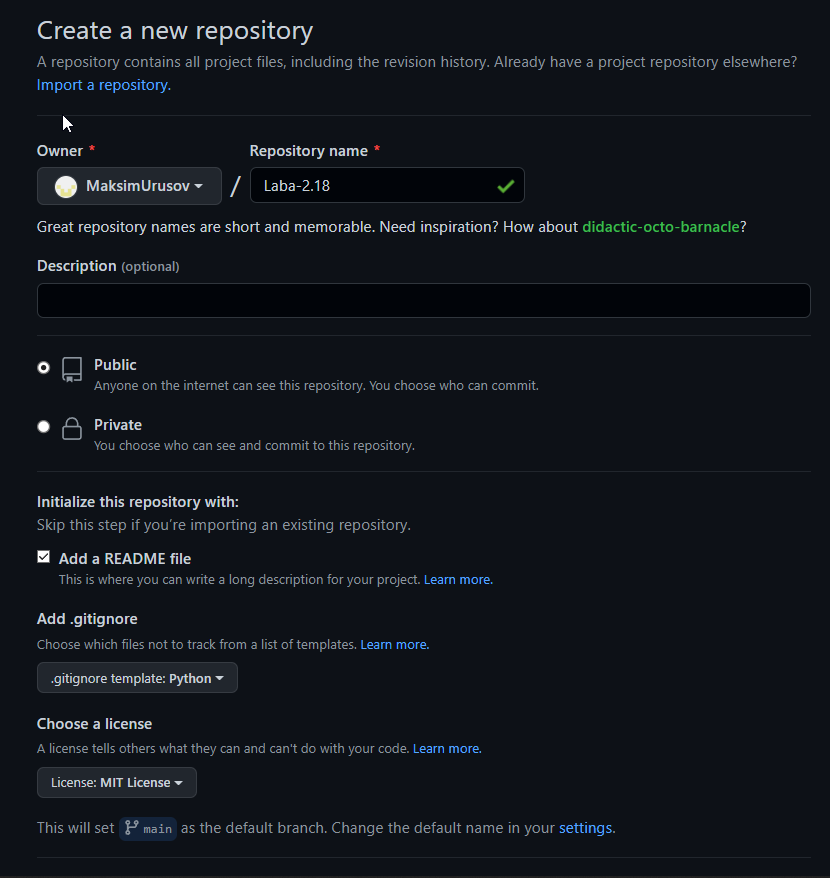


Рисунок 1 - Создание репозитория

1. Выполните клонирование созданного репозитория.

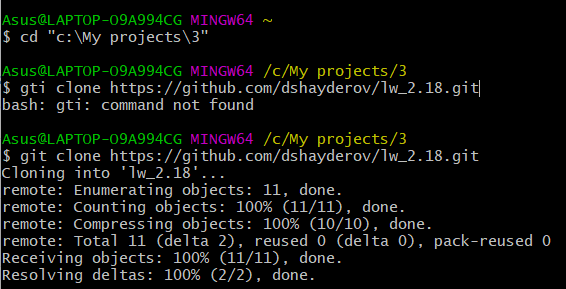


Рисунок 2 - Клонирование репозитория

1. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.

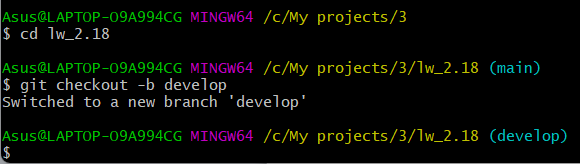


Рисунок 3 - Ветвление по модели git-flow

1. Создание виртуального окружения.

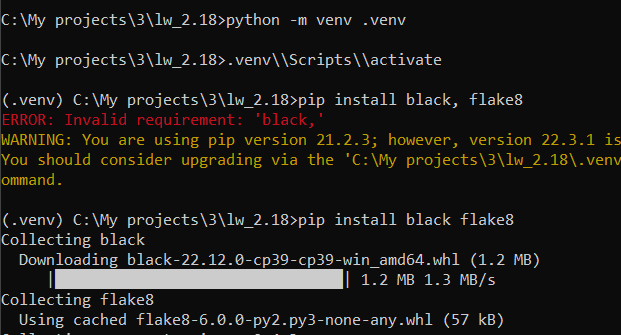


Рисунок 4 - Виртуальное окружение

1. Формирование файла requirements.txt.

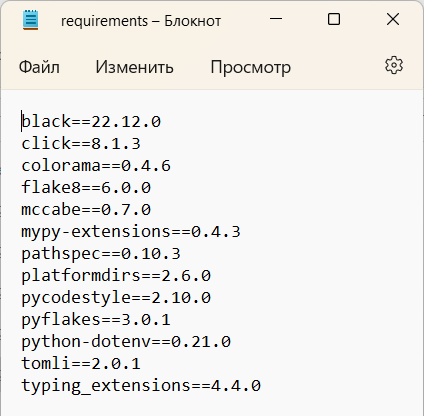


Рисунок 5 - Файл requirements.txt

1. Создайте проект PyCharm в папке репозитория.

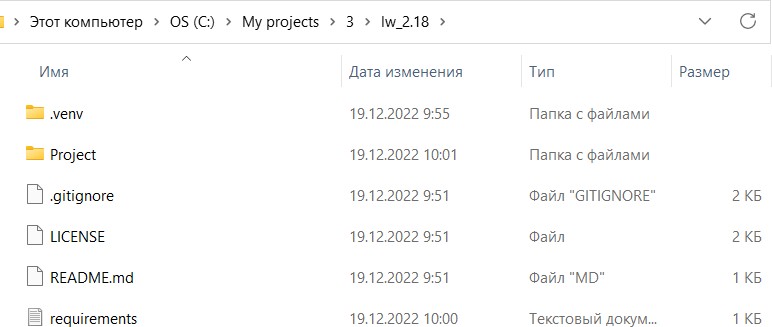


Рисунок 6 - Создание проекта

1. Проработать примеры лабораторной работы.

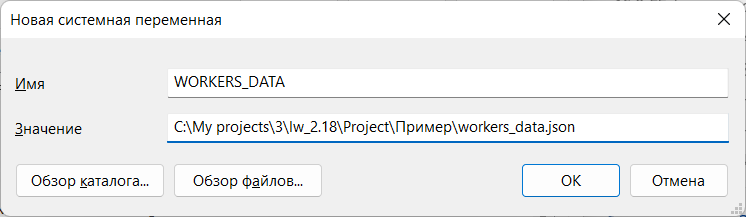


Рисунок 7 - Создание системной переменной WORKERS\_DATA

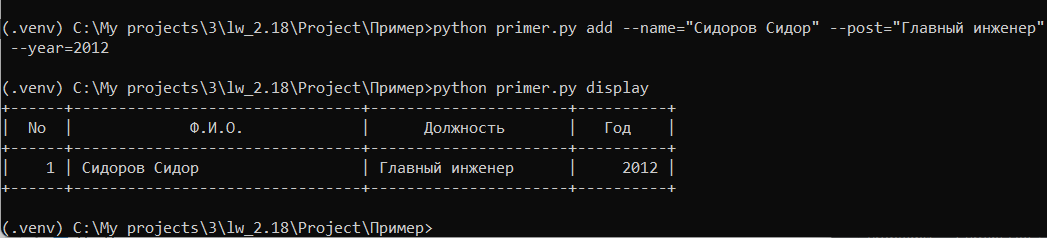


Рисунок 8 - Результат выполнения примера

1. Выполнить индивидуальные задания.

# Задание 1

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

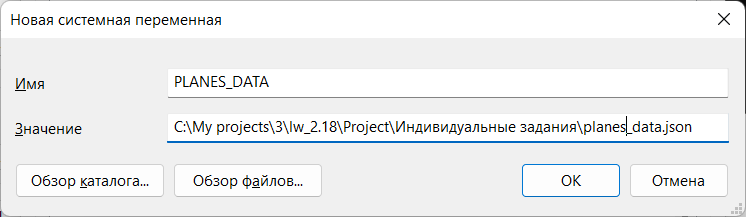


Рисунок 9 - Создание переменной окружения PLANES\_DATA

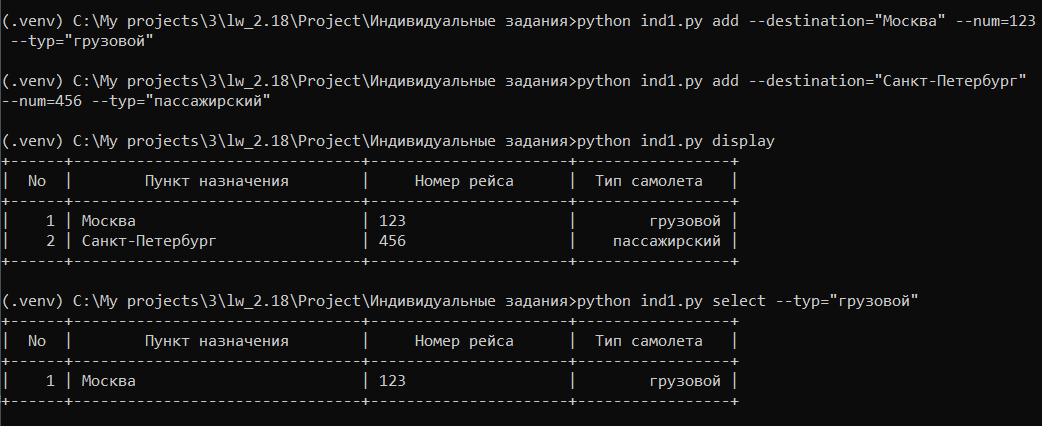


Рисунок 10 - Результат выполнения задания 1

# Задание 2

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

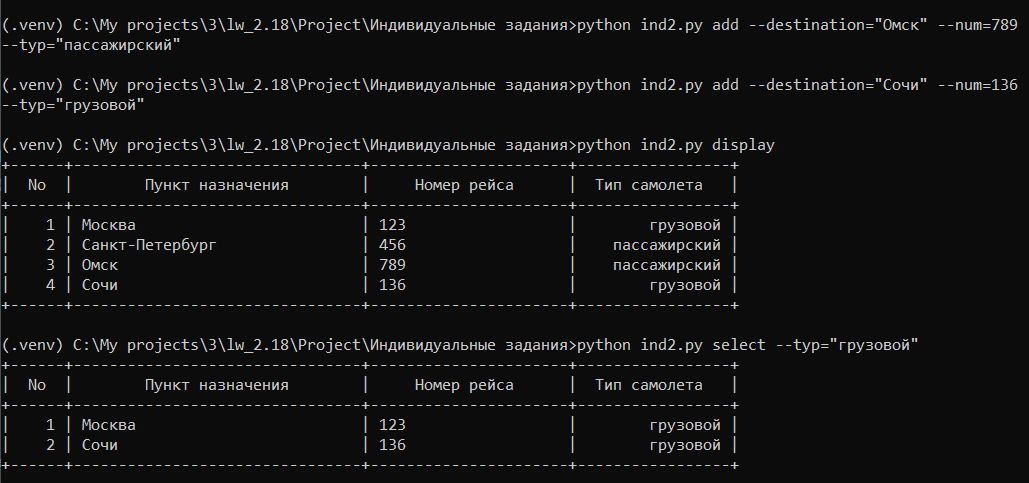


Рисунок 11 - Результат выполнения задания 2

# Контрольные вопросы:

1. **Каково назначение переменных окружения?**

Переменная среды (переменная окружения) – это короткая ссылка на какой-либо объект в системе. С помощью таких сокращений, например, можно создавать универсальные пути для приложений, которые будут работать на любых ПК, независимо от имен пользователей и других параметров.

# Какая информация может храниться в переменных окружения?

Переменная окружения может хранить информацию о путях к исполняемым файлам, заданном по умолчанию текстовом редакторе, браузере, языковых параметрах (локали) системы или настройках раскладки клавиатуры.

# Как получить доступ к переменным окружения в ОС Windows?

Получить информацию о существующих переменных можно в свойствах системы. Для этого кликаем по ярлыку Компьютера на рабочем столе правой кнопкой мыши и выбираем соответствующий пункт.

Переходим в «Дополнительные параметры».

В открывшемся окне с вкладкой «Дополнительно» нажимаем кнопку

«Переменные среды».

# Каково назначение переменных PATH и PATHEXT?

«PATH» позволяет запускать исполняемые файлы и скрипты,

«лежащие» в определенных каталогах, без указания их точного местоположения.

PATHEXT, в свою очередь, дает возможность не указывать даже расширение файла, если оно прописано в ее значениях.

# Как создать или изменить переменную окружения в Windows?

Нажимаем кнопку Создать. Сделать это можно как в пользовательском разделе, так и в системном.

Вводим имя, например, desktop. Обратите внимание на то, чтобы такое название еще не было использовано (просмотрите списки).

В поле Значение указываем путь.

# Что представляют собой переменные окружения в ОС LInux?

Переменные окружения в Linux представляют собой набор именованных значений, используемых другими приложениями.

# В чем отличие переменных окружения от переменных оболочки?

Переменные окружения (или «переменные среды») — это переменные, доступные вимасштабах всей системы и наследуемые всеми дочерними процессами и оболочками.

Переменные оболочки — это переменные, которые применяются только к текущему экземпляру оболочки. Каждая оболочка, например, bash или zsh, имеет свой собственный набор внутренних переменных.

# Как вывести значение переменной окружения в Linux?

Наиболее часто используемая команда для вывода переменных окружения — printenv.

# Какие переменные окружения Linux Вам известны?

USER — текущий пользователь. PWD — текущая директория.

OLDPWD — предыдущая рабочая директория. Используется оболочкой для того, чтобы вернуться в предыдущий каталог при выполнении команды cd - .

HOME — домашняя директория текущего пользователя.

SHELL — путь к оболочке текущего пользователя (например, bash или

zsh).

EDITOR — заданный по умолчанию редактор. Этот редактор будет

вызываться в ответ на команду edit .

LOGNAME — имя пользователя, используемое для входа в систему.

PATH — пути к каталогам, в которых будет производиться поиск вызываемых команд. При выполнении команды система будет проходить по данным каталогам в указанном порядке и выберет первый из них, в котором будет находиться исполняемый файл искомой команды.

LANG — текущие настройки языка и кодировки. TERM — тип текущего эмулятора терминала.

MAIL — место хранения почты текущего пользователя.

LS\_COLORS — задает цвета, используемые для выделения объектов (например, различные типы файлов в выводе команды ls будут выделены разными цветами).

# Какие переменные оболочки Linux Вам известны?

BASHOPTS — список задействованных параметров оболочки, разделенных двоеточием.

BASH\_VERSION — версия запущенной оболочки bash.

COLUMNS — количество столбцов, которые используются для отображения выходных данных.

DIRSTACK — стек директорий, к которому можно применять команды pushd и popd .

HISTFILESIZE — максимальное количество строк для файла истории команд.

HISTSIZE — количество строк из файла истории команд, которые можно хранить в памяти.

HOSTNAME — имя текущего хоста.

IFS — внутренний разделитель поля в командной строке (по умолчанию используется пробел).

PS1 — определяет внешний вид строки приглашения ввода новых команд.

PS2 — вторичная строка приглашения.

SHELLOPTS — параметры оболочки, которые можно устанавливать с помощью команды set .

UID — идентификатор текущего пользователя.

1. Как установить переменные оболочки в Linux?

Чтобы создать новую переменную оболочки с именем, например, NEW\_VAR и значением Ravesli.com , просто введите:

$ NEW\_VAR='Ravesli.com'

# Как установить переменные окружения в Linux?

Команда export используется для задания переменных окружения.

# Для чего необходимо делать переменные окружения Linux постоянными?

Чтобы переменная сохранялась после закрытия сеанса оболочки.

# Для чего используется переменная окружения PYTHONHOME?

Переменная среды PYTHONHOME изменяет расположение стандартных библиотек Python.

# Для чего используется переменная окружения PYTHONPATH?

Переменная среды PYTHONPATH изменяет путь поиска по умолчанию для файлов модуля.

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

PYTHONSTARTUP, PYTHONOPTIMIZE, PYTHONBREAKPOINT, PYTHONDEBUG, PYTHONINSPECT, PYTHONUNBUFFERED, PYTHONVERBOSE, PYTHONCASEOK, PYTHONDONTWRITEBYTECODE, PYTHONPYCACHEPREFIX, PYTHONHASHSEED, PYTHONIOENCODING, PYTHONNOUSERSITE, PYTHONUSERBASE, PYTHONWARNINGS, PYTHONFAULTHANDLER, PYTHONTRACEMALLOC, PYTHONPROFILEIMPORTTIME, PYTHONASYNCIODEBUG, PYTHONMALLOC, PYTHONMALLOCSTATS,

PYTHONLEGACYWINDOWSFSENCODING, PYTHONLEGACYWINDOWSSTDIO, PYTHONCOERCECLOCALE, PYTHONDEVMODE,PYTHONUTF8, PYTHONWARNDEFAULTENCODING, PYTHONTHREADDEBUG, PYTHONDUMPREFS.

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

Для доступа к переменным среды в Python используется объект os.environ.

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

Для чтения значений переменных мы используем модуль os, а модуль sys — для прекращения работы приложения.

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

Для присвоения значения любой переменной среды используется функция setdefault().

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