

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**
**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Институт цифрового развития

ОТЧЁТ

по лабораторной работе №2.18

Дисциплина: «Программирование на Python»

Тема: «Работа с переменными окружения в Python3»

Выполнил: студент 2 курса

группы ИВТ-б-о-21-1

Кочкаров Умар Ахматович

Ставрополь 2022


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

Ход работы:

1. Создать общедоступный репозиторий с лицензией MIT и языком Python.

Owner *

Repository name *


 umarkochkarov ▾

 /


lb2.18 ✓

Great repository names are short and memorable. Need inspiration? How about [miniature-carnival?](#)

Description (optional)

☒  Public

Anyone on the internet can see this repository. You choose who can commit.

☐  Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☒ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore


Choose which files not to track from a list of templates. [Learn more.](#)


.gitignore template: Python ▾

Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

License: MIT License ▾

This will set  main as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

Create repository

Рисунок 1. Создание репозитория

2. Клонировать репозиторий на ПК:

```

erken@LAPTOP-ESTC60GF MINGW64 ~/Desktop/python/Ла62.18
$ git clone https://github.com/umarkochkarov/lb2.18.git
Cloning into 'lb2.18'...
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 2. Клонирование репозитория

3. Организовать репозиторий в соответствии с моделью ветвления git-flow.

```

erken@LAPTOP-ESTC60GF MINGW64 ~/Desktop/python/Ла62.18/lb2.18 (main)
$ git flow init

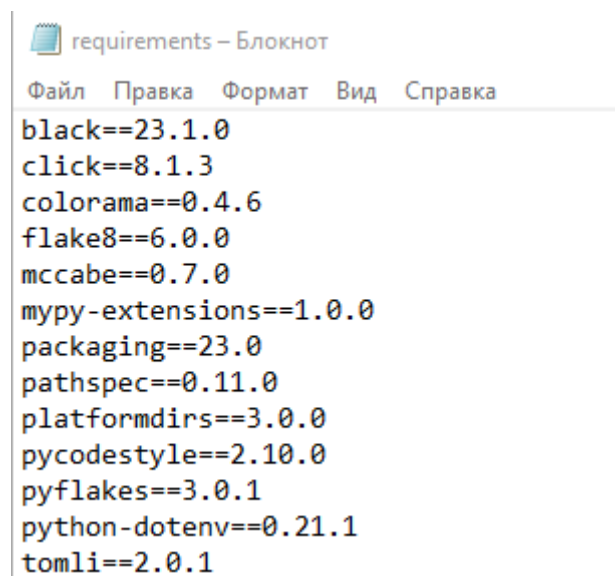
Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

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/erken/Desktop/python/Ла62.18/lb2.18/.git/hooks]

```

Рисунок 3. Организация репозитория в соответствии с моделью git-flow

4. Создание виртуального окружения и формирование файла requirements.txt.



```

requirements - Блокнот
Файл  Правка  Формат  Вид  Справка
black==23.1.0
click==8.1.3
colorama==0.4.6
flake8==6.0.0
mccabe==0.7.0
mypy-extensions==1.0.0
packaging==23.0
pathspec==0.11.0
platformdirs==3.0.0
pycodestyle==2.10.0
pyflakes==3.0.1
python-dotenv==0.21.1
tomli==2.0.1

```

Рисунок 4. Создание виртуального окружения и формирование файла requirements.txt.

5. Проработка примеров из лабораторной работы:

Переменные среды пользователя для erken	
Переменная	Значение
OneDrive	C:\Users\erken\OneDrive
Path	C:\Users\erken\AppData\Local\Microsoft\WindowsApps;C:\Users\erken\AppData\Local\Microsoft\WindowsApps;C:\Users\erken\AppData\Local\Microsoft\WindowsApps
PT6HOME	C:\Program Files (x86)\Cisco Packet Tracer 6.2sv
TEMP	C:\Users\erken\AppData\Local\Temp
TMP	C:\Users\erken\AppData\Local\Temp
WORKERS_DATA	C:\Users\erken\Desktop\python\Ла62.18\lb2.18\prim\prim.py

Рисунок 5. Создание системной переменной WORKERS_DATA

```
(env) C:\Users\erken\Desktop\python\Ла62.18\lb2.18\prim>python prim.py display
+-----+-----+-----+-----+
| No | Ф.И.О. | Должность | Год |
+-----+-----+-----+-----+
| 1 | Иван Иванов | Главный инженер | 2012 |
+-----+-----+-----+-----+
```

Рисунок 6. Результат выполнения

6. Выполнение индивидуальных заданий:

Задание 1 Для своего варианта лабораторной работы 2.17 добавьте возможность получения имени файла данных, используя соответствующую переменную окружения

Переменные среды пользователя для erken	
Переменная	Значение
OneDrive	C:\Users\erken\OneDrive
Path	C:\Users\erken\AppData\Local\Microsoft\WindowsAp
POEZD_DATA	C:\Users\erken\Desktop\python\Ла62.18\lb2.18\ind\ir
PT6HOME	C:\Program Files (x86)\Cisco Packet Tracer 6.2sv
TEMP	C:\Users\erken\AppData\Local\Temp
TMP	C:\Users\erken\AppData\Local\Temp
WORKERS_DATA	C:\Users\erken\Desktop\python\Ла62.18\lb2.18\prim\

Рисунок 7. Переменная

```
+-----+-----+-----+
| No | Название | Время |
+-----+-----+-----+
| 1 | one | 12:21:00 |
+-----+-----+-----+
```

Рисунок 8. Результат выполнения

Задание 2. Самостоятельно изучите работу с пакетом python-dotenv. Модифицируйте программу задания 1 таким образом, чтобы значения необходимых переменных окружения считывались из файла .env.

```
: \Users\Dina\Desktop\Вуз\Lab2_18\prog>py ind2.py display
-----+-----+-----+
      No      |      Название      |      Время      |
-----+-----+-----+
          1    |      one            |      12:21:00    |
-----+-----+-----+

.\Users\Dina\Desktop\Вуз\Lab2_18\prog>
```

Рисунок 9. Результат выполнения

Контрольные вопросы:

1. Каково назначение переменных окружения?

Переменные окружения используются для передачи информации процессам, которые запущены в оболочке.

2. Какая информация может храниться в переменных окружения?

Переменные среды хранят информацию о среде операционной системы. Эта информация включает такие сведения, как путь к операционной системе, количество процессоров, используемых операционной системой, и

расположение временных папок.

3. Как получить доступ к переменным окружения в ОС Windows?

Нужно открыть окно свойства системы и нажать на кнопку “Переменные среды”.

4. Каково назначение переменных PATH и PATHNEXT?

PATH позволяет запускать исполняемые файлы и скрипты, «лежащие» в определенных каталогах, без указания их точного местоположения.

PATHNEXT дает возможность не указывать даже расширение файла, если оно прописано в ее значениях.

5. Как создать или изменить переменную окружения в Windows?

В окне “Переменные среды” нужно нажать на кнопку “Создать”, затем ввести имя переменной и путь.

6. Что представляют собой переменные окружения в ОС Linux?

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

7. В чем отличие переменных окружения от переменных оболочки?

Переменные окружения (или «переменные среды») – это переменные, доступные в масштабах 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.

8. Как вывести значение переменной окружения в Linux?

Наиболее часто используемая команда для вывода переменных окружения – `printenv`.

9. Какие переменные окружения Linux Вам известны?

USER — текущий пользователь.

PWD – текущая директория.

HOME – домашняя директория текущего пользователя.

SHELL – путь к оболочке текущего пользователя.

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.

10. 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.

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.

11. Как установить переменные оболочки в Linux?

Чтобы создать новую переменную оболочки с именем, нужно ввести Имя этой переменной потом знак равенства и указать значение новой переменной

12. Как установить переменные окружения в Linux?

Команда export используется для задания переменных окружения.

С помощью данной команды мы экспортируем указанную переменную,

в результате чего она будет видна во всех вновь запускаемых дочерних командных оболочках.

13. Для чего необходимо делать переменные окружения Linux постоянными?

Чтобы переменная сохранялась после закрытия сеанса оболочки.

14. Для чего используется переменная окружения PYTHONHOME? Переменная среды PYTHONHOME изменяет расположение стандартных библиотек Python.

15. Для чего используется переменная окружения PYTHONPATH? Переменная среды PYTHONPATH изменяет путь поиска по умолчанию для файлов модуля.

16. 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

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

Путём использования модуля os, при помощи которого программист может получить и изменить значения всех переменных среды.

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

При помощи модуля os можно просмотреть все переменные окружения, у которых есть значение.

19. Как присвоить значение переменной окружения в программах

на языке программирования Python?

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

Вывод: в результате выполнения работы были приобретены навыки по работе с переменными окружения с помощью языка программирования Python версии 3.x