

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

Отчет по лабораторной работе №2.18

Тема: «Работа с переменными окружения в Python3»

Выполнила студентка группы
ИВТ-б-о-21-1

Диченко Дина Алексеевна

« » _____ 20__ г.

Подпись студента _____

Работа защищена

« » _____ 20__ г.

Проверил доцент

Кафедры инфокоммуникаций,

старший преподаватель

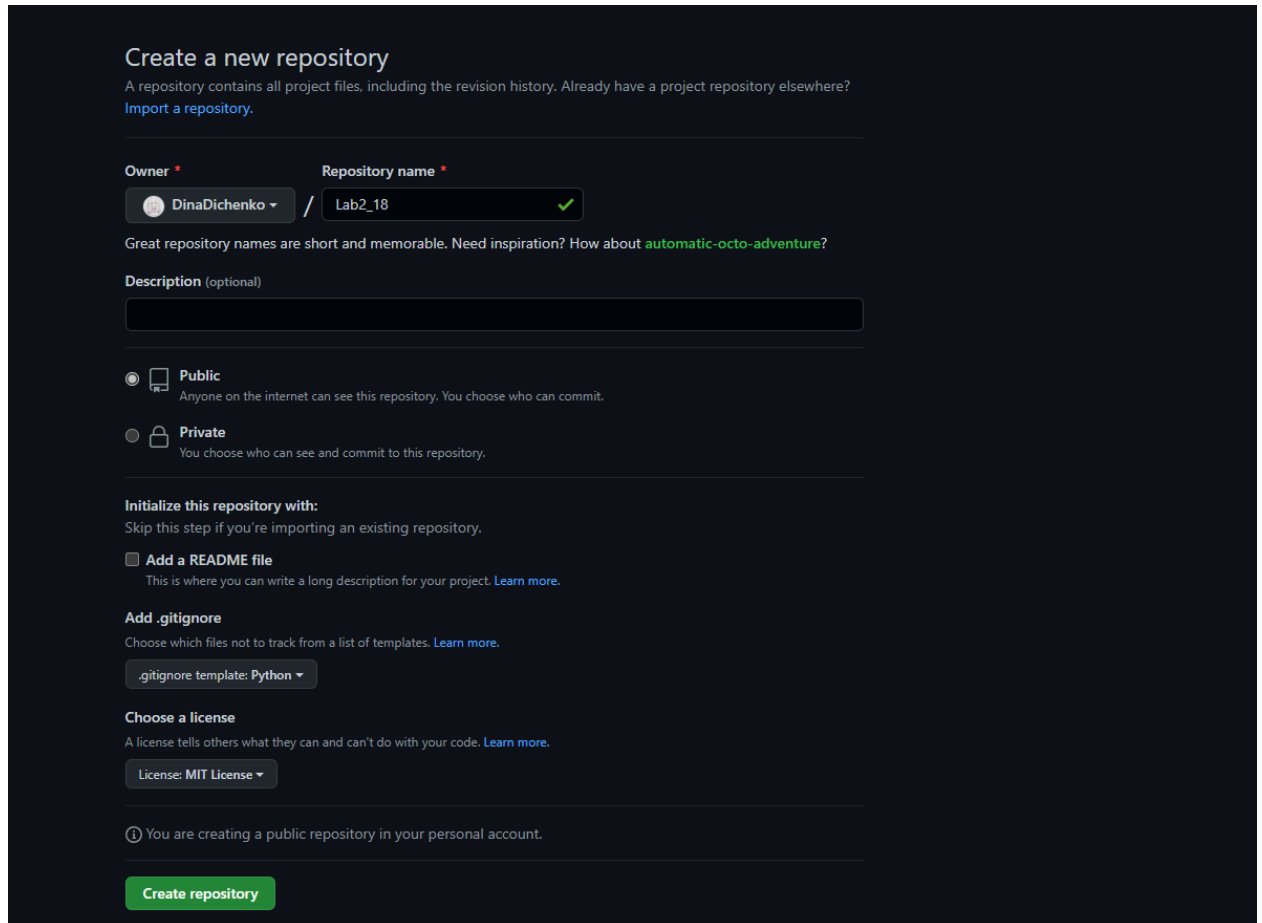
Воронкин Р.А.

(подпись)

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

Практическая часть:



1. Создала репозиторий.



Create a new repository


A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Owner ^{*} Repository name ^{*}

 DinaDichenko / Lab2_18 

Great repository names are short and memorable. Need inspiration? How about [automatic-octo-adventure?](#)

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 

 You are creating a public repository in your personal account.

[Create repository](#)

Рисунок 1. Создание репозитория

2. Клонировала репозиторий.

```
C:\Users\Dina\Desktop\Вуз>git clone https://github.com/DinaDichenko/Lab2_18.git
Cloning into 'Lab2_18'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.

C:\Users\Dina\Desktop\Вуз>
```

Рисунок 2. Клонирование репозитория

3. Изменила файл .gitignore

```

ENV/
env.bak/
venv.bak/

# Spyder project settings
.spyderproject
.spyproject

# Rope project settings
.ropeproject

# mkdocs documentation
/site

# mypy
.mypy_cache/
.dmypy.json
dmypy.json

# Pyre type checker
.pyre/

# pytype static type analyzer
.pytype/

# Cython debug symbols
cython_debug/

# PyCharm
# JetBrains specific template is maintained in a separate JetBrains.gitignore that can
# be found at https://github.com/github/gitignore/blob/main/Global/JetBrains.gitignore
..

```

Рисунок 3. Изменение файла .gitignore

4. Организовала репозиторий в соответствии с моделью ветвления git-flow.

```

C:\Users\Dina\Desktop\Быз\Lab2_18>git branch develop
C:\Users\Dina\Desktop\Быз\Lab2_18>git checkout develop
Switched to branch 'develop'

C:\Users\Dina\Desktop\Быз\Lab2_18>git flow init

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

Which branch should be used for integration of the "next release"?
- develop
Branch name for "next release" development: [develop] develop

How to name your supporting branch prefixes?
Feature branches? [feature/] fea
Bugfix branches? [bugfix/] bug
Release branches? [release/] branch
Hotfix branches? [hotfix/] hot
Support branches? [support/] sup
Version tag prefix? [] tag
Hooks and filters directory? [C:/Users/Dina/Desktop/Быз/Lab2_18/.git/hooks] hooks

```

Рисунок 4. Организация репозитория в соответствии с git-flow

5. Проработала примеры.

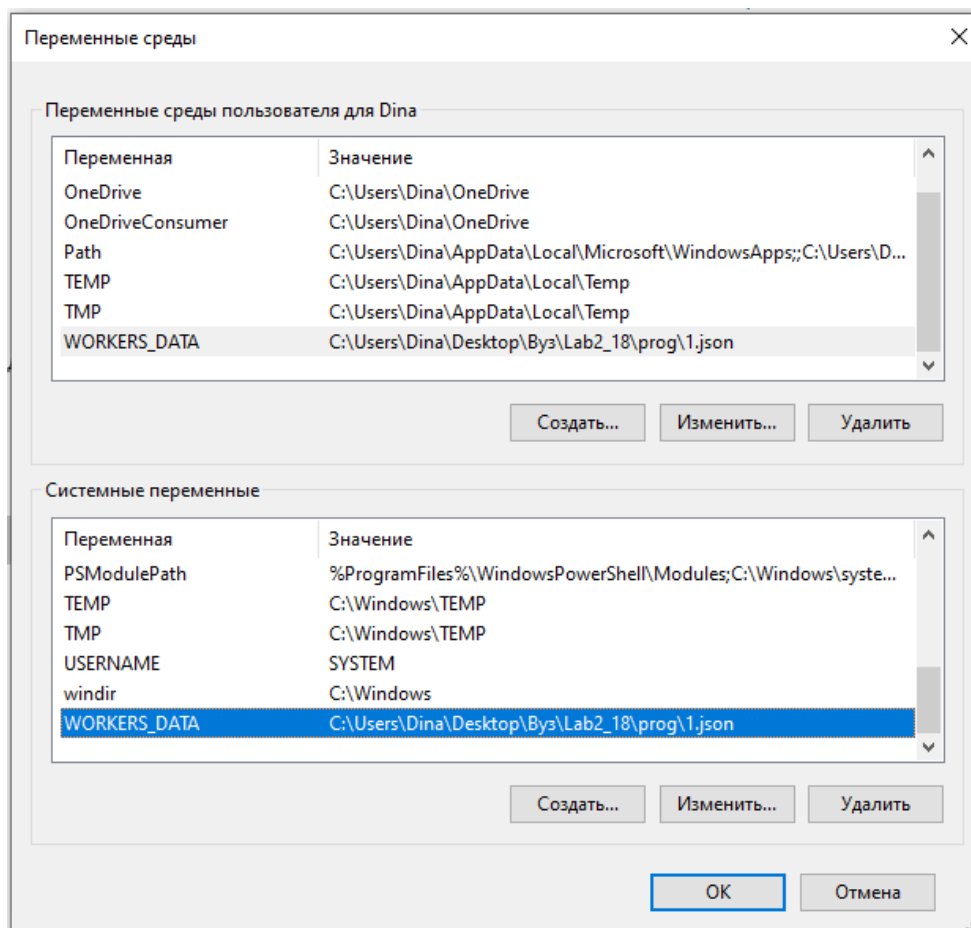


Рисунок 5. Создание переменных

```
C:\Users\Dina\Desktop\Быз\Lab2_18\prog>python 1.py add --name="Сидоров Сидор" --post="Главный инженер" --year=2012
C:\Users\Dina\Desktop\Быз\Lab2_18\prog>python 1.py display
```

No	Ф.И.О.	Должность	Год
1	Сидоров Сидор	Главный инженер	2012

Рисунок 6. Результат выполнения программы

6. Выполнила индивидуальные задания.

Задание 1

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

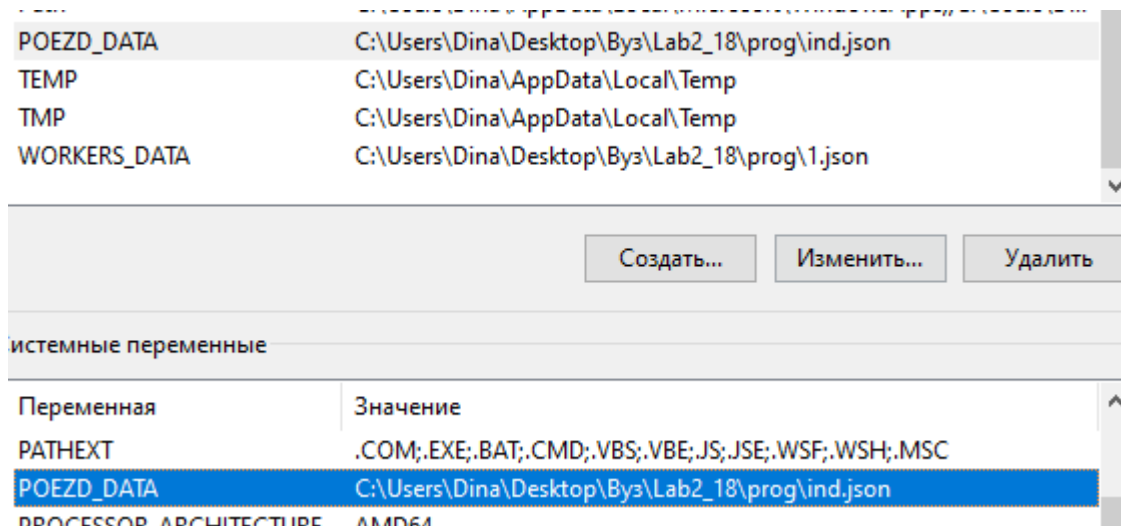


Рисунок 7. Создание переменных

```
C:\Users\Dina\Desktop\Byz\Lab2_18\prog>py ind.py add --name="one" --no=1 --time="12:21:00"
C:\Users\Dina\Desktop\Byz\Lab2_18\prog>py ind.py display
```

No	Название	Время
1	one	12:21:00

Рисунок 8. Результат выполнения программы

Задание 2

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

```
C:\Users\Dina\Desktop\Byz\Lab2_18\prog>py ind2.py display
```

No	Название	Время
1	one	12:21:00

```
C:\Users\Dina\Desktop\Byz\Lab2_18\prog>
```

Рисунок 8. Результат выполнения программы

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 PATHEXT?

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

PATHEXT 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

находится 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.