МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Отчет по лабораторной работе №13

Работа с переменными окружения в Python3

По дисциплине «Технологии программирования и алгоритмизация»

Выполнил студент группы ИВТ-б-о-20-1

Галяс Д. И. « »_____20__г.

Подпись студента_____

Работа защищена « »_____20__г.

Проверил Воронкин Р. А. _____

(подпись)

Ставрополь 2021

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

Ход работы:

Ссылка на репозиторий: https://github.com/DIMITRY-GALYAS1/Laba-2.18.git

1. Создал новый репозиторий на github, после клонировал его и создал в папке репозитория новый проект PyCharm.

2. Выполнил пример.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import argparse
import json
import os.path
import sys
from datetime import date


def add_worker(staff, name, post, year):
    """
    Добавить данные о работнике.
    """
    staff.append(
        {
            "name": name,
            "post": post,
            "year": year
        }
    )
    return staff


def display_workers(staff):
    """
    Отобразить список работников.
    """
    # Проверить, что список работников не пуст.
    if staff:
        # Заголовок таблицы
```

Рисунок 1. Код примера

```
PS C:\Users\lizeq\PycharmProjects\pythonProject13> python primer_1.py add --name="Сидоров Сидр" --post="главный инженер" --year=2012
PS C:\Users\lizeq\PycharmProjects\pythonProject13> python primer_1.py display
+------+--------------------------------+----------------------+----------+
|  No  |            Ф.И.О.              |      Должность       |   Год    |
+------+--------------------------------+----------------------+----------+
|    1 | Сидоров Сидр                   | главный инженер      |     2012 |
+------+--------------------------------+----------------------+----------+
Список работников пуст.
PS C:\Users\lizeq\PycharmProjects\pythonProject13> python primer_1.py select --period=12
PS C:\Users\lizeq\PycharmProjects\pythonProject13> python primer_1.py select --period=8
+------+--------------------------------+----------------------+----------+
|  No  |            Ф.И.О.              |      Должность       |   Год    |
+------+--------------------------------+----------------------+----------+
|    1 | Сидоров Сидр                   | главный инженер      |     2012 |
+------+--------------------------------+----------------------+----------+
```

Рисунок 2. Выполнение примера

3. Выполнил индивидуальное задание.

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import argparse
import json
import os.path
import sys


def add_student(students, name, group, progress):
    """
    Запросить данные о студенте.
    """
    students.append(
        {
            'name': name,
            'group': group,
            'progress': progress,
        }
    )
    return students


def display_students(students):
    """
    Отобразить список студентов.
    """
    # Проверить, что список студентов не пуст.
    if students:
        # Заголовок таблицы.
        line = '+-{}-+-{}-+-{}-+-{}-+'.format(
```

Рисунок 3. Код индивидуального задания

```
PS C:\Users\lizeq\PycharmProjects\pythonProject13> python individual_1.py add --name="Сидоров Сидр" --group="2" --progress=23444
Данные сохранены
PS C:\Users\lizeq\PycharmProjects\pythonProject13> python primer_1.py display
PS C:\Users\lizeq\PycharmProjects\pythonProject13> python individual_1.py display
Нет ошибок валидации
+------+--------------------------------+----------------------+-----------------+
|  No  |            Ф.И.О.              |        Группа        |  Успеваемость   |
+------+--------------------------------+----------------------+-----------------+
|    1 | Сидоров Сидр                   | 2                    |           23444 |
+------+--------------------------------+----------------------+-----------------+
PS C:\Users\lizeq\PycharmProjects\pythonProject13> python individual_1.py select
Нет ошибок валидации
+------+--------------------------------+----------------------+-----------------+
|  No  |            Ф.И.О.              |        Группа        |  Успеваемость   |
+------+--------------------------------+----------------------+-----------------+
|    1 | Сидоров Сидр                   | 2                    |           23444 |
+------+--------------------------------+----------------------+-----------------+
PS C:\Users\lizeq\PycharmProjects\pythonProject13>
```

Рисунок 4. Работа кода

4. Затем выполнил второе индивидуальное задание.

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import click
import json
import os
import sys
from dotenv import load_dotenv


@click.group()
def cli():
    pass


@cli.command()
@click.argument('data')
@click.option("-n", "--name")
@click.option("-g", "--group")
@click.option("-p", "--progress")
def add(data, name, group, progress):
    """
    Добавление нового студента
    """
    if os.path.exists(data):
        load_dotenv()
        dotenv_path = os.getenv("STUDENTS_DATA")
        if not dotenv_path:
            click.secho('Файла нет', fg='red')
            sys.exit(1)
        if os.path.exists(dotenv_path):
            students = open_file(dotenv_path)
```

Рисунок 5. Код задания

```
1  STUDENTS_DATA
```

Рисунок 6. Содержимое файла .env

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.

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?

Получить информацию о существующих переменных можно в свойствах системы. Для этого кликаем по ярлыку компьютер и переходим в пункт свойства системы. Далее дополнительные параметры системы и нажимаем 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, в свою очередь, дает возможность не указывать даже расширение файла, если оно прописано в ее значениях.

5. Как создать или изменить переменную окружения в Windows?

Чтобы добавить или изменить переменные среды, пользователь выбирает пункт система на панели управления, а затем выбирает вкладку Среда. Пользователь также может добавлять или изменять переменные среды в командной строке с помощью команды Set. Переменные среды, созданные с помощью команды Set, применяются только к командному окну, в котором они заданы, и к его дочерним процессам.

6. Что представляют собой переменные окружения в ОС LInux?

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

Переменные окружения применяются для настройки поведения приложений и работы самой системы. Например, переменная окружения может хранить информацию о путях к исполняемым файлам, заданном по

умолчанию текстовом редакторе, браузере, языковых параметрах системы или настройках раскладки клавиатуры.

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?

Значение каждой переменной окружения изначально представляет собой строковую константу (строку). Интерпретация значений переменных полностью возлагается на программу. Иными словами, все переменные окружения имеют тип char*, а само окружение имеет тип char**. Чтобы вывести на экран значение какой-нибудь переменной окружения, достаточно набрать echo $ИМЯ_ПЕРЕМЕННОЙ.

9. 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).

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 (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.

11. Как установить переменные оболочки в Linux?

Команда set — устанавливает переменные оболочки. При использовании без аргумента выведет список всех переменных, включая переменные окружения и переменные оболочки, а также функции оболочки.

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. Переменные такого типа принято называть внешними.

13. Для чего необходимо делать переменные окружения Linux постоянными?

Если вы хотите, чтобы переменная сохранялась после закрытия сеанса оболочки, то необходимо прописать её в специальном файле. Прописать переменную можно как для текущего пользователя, так и для всех пользователей.

14. Для чего используется переменная окружения PYTHONHOME ?

Переменная среды PYTHONHOME изменяет расположение стандартных библиотек Python. По умолчанию библиотеки ищутся в prefix/lib/pythonversion и exec_prefix/lib/pythonversion, где prefix и exec_prefix - это каталоги, зависящие от установки, оба каталога по умолчанию - /usr/local.

15. Для чего используется переменная окружения PYTHONPATH?

Переменная среды PYTHONPATH изменяет путь поиска по умолчанию для файлов модуля. Формат такой же, как для оболочки PATH: один или несколько путей к каталогам, разделенных os.pathsep (например, двоеточие в Unix или точка с запятой в Windows). Несуществующие каталоги игнорируются.

16. Какие еще переменные окружения используются для управления работой интерпретатора Python?

PYTHONSTARTUP:

Если переменная среды PYTHONSTARTUP это имя файла, то команды Python в этом файле выполняются до отображения первого приглашения в

интерактивном режиме. Файл выполняется в том же пространстве имен, в котором выполняются интерактивные команды, так что определенные или

импортированные в нем объекты можно использовать без квалификации в интерактивном сеансе. При запуске вызывает событие аудита cpython.run_startup с именем файла в качестве аргумента.

PYTHONOPTIMIZE:

Если в переменной среды PYTHONOPTIMIZE задана непустая строка, это эквивалентно указанию параметра -O . Если установлено целое число, то это эквивалентно указанию -OO.

PYTHONBREAKPOINT:

Если переменная среды PYTHONBREAKPOINT установлена, то она определяет вызываемый объект с помощью точечной нотации.

Модуль, содержащий вызываемый объект, будет импортирован, а затем вызываемый объект будет запущен реализацией по умолчанию

sys.breakpointhook(), которая сама вызывается встроенной функцией breakpoint() . Если PYTHONBREAKPOINT не задан или установлен в пустую строку, то это эквивалентно значению pdb.set_trace . Установка этого значения в строку 0 приводит к тому, что стандартная реализация sys.breakpointhook() ничего не делает, кроме немедленного возврата.

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

Значения переменных окружения можно считывать и изменять при помощи объекта environ модуля os либо путем использования функций setdefault() и get(). В качестве ключа, по которому можно обратиться и получить либо присвоить значение переменной, в environ используется имя переменной окружения.

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

С помощью следующего кода:

if os.environ[key_value]:

```
print(
"The value of",
key_value,
" is ",
os.environ[key_value]
)
```

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

С помощью следующего кода:

```
os.environ.setdefault('DEBUG', 'True')
```

Вывод: в ходе выполнения лабораторной работы приобрел навыки по работе с переменными окружения с помощью языка программирования Python версии 3.x.