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 окружениями в Python3

по дисциплине «Технологии программирования и алгоритмизация»

Выполнил студент группы ИВТ-б-о-20-1

Бобров Н. В. « »_____20__г.

Подпись студента_____

Работа защищена « »_____20__г.

Проверил Воронкин Р. А. _____

(подпись)

Ставрополь 2021

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

Ход работы:

1. Создал общедоступный репозиторий, клонировал его на локальный сервер.

2. Создал новое виртуальное окружение и установит необходимые пакеты, затем сделал список зависимостей.

```
PS C:\Users\DNS\PycharmProjects\pythonProject6\tasks> conda env export > environment.yml
PS C:\Users\DNS\PycharmProjects\pythonProject6\tasks> pip freeze > requirements.txt
```

Рисунок 1 – Получение списка зависимостей

3. Изучив теоретический материал, приступил к выполнению примера. Для этого создал переменное окружение, где будет хранится файл json.

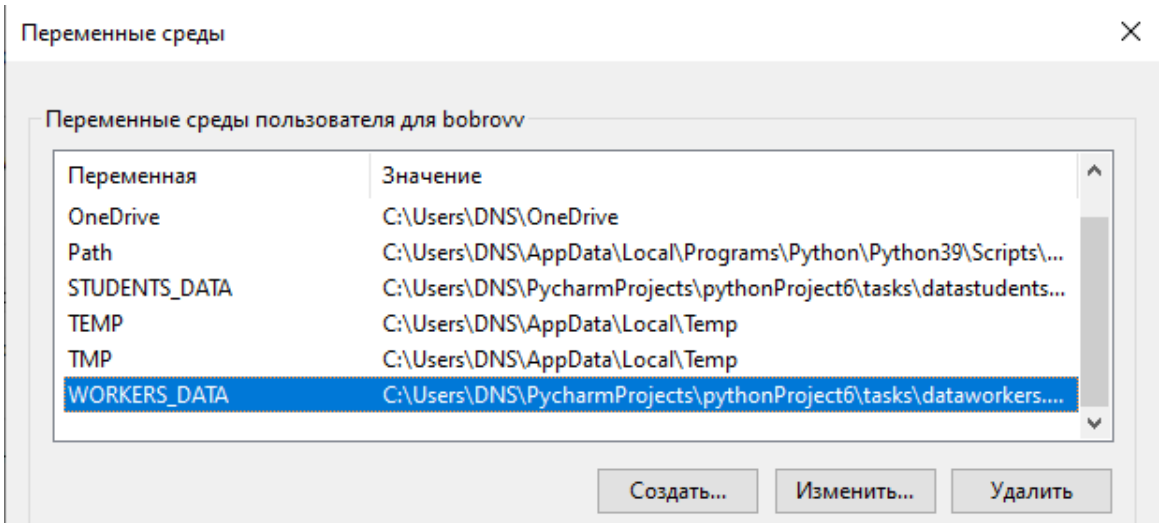

Рисунок 2 – Переменные среды

4. Дополнил код примера из лабораторной 2.17 и сделал проверку, не указывая файл json.

```
    # Выполнить разбор аргументов командной строки.
    args = parser.parse_args(command_line)

    # Получить имя файла.
    data_file = args.data
    if not data_file:
        data_file = os.environ.get("WORKERS_DATA")
    if not data_file:
        print("The data file name is absent", file=sys.stderr)
        sys.exit(1)

    # Загрузить всех работников из файла, если файл существует.
    is_dirty = False
    if os.path.exists(data_file):
        workers = load_workers(data_file)
    else:
        workers = []
```

Рисунок 3 – Код из примера

```
PS C:\Users\DNS\PycharmProjects\pythonProjectó\tasks> python example.py add -n="Arsen Wenger" -p="Couch" -y=2000
PS C:\Users\DNS\PycharmProjects\pythonProjectó\tasks> python example.py display
+------+--------------------------------+----------------------+----------+
|  No  |            Ф.И.О.              |      Должность       |   Год    |
+------+--------------------------------+----------------------+----------+
|    1 | Bobrov Nikolay                 | Student              |     2019 |
|    2 | Sergeev Sergey                 | Developer            |     2010 |
|    3 | Arsen Wenger                   | Couch                |     2000 |
+------+--------------------------------+----------------------+----------+
PS C:\Users\DNS\PycharmProjects\pythonProjectó\tasks> python example.py select --period=6
Список работников пуст.
```

Рисунок 4 – Выполнение кода

5. Приступил к выполнению индивидуального задания, для этого создал новую переменную среду и дополнил код.

```python
    select.add_argument(
        "-s",
        "--select",
        action="store",
        required=True,
        help="The required select"
    )

    # Выполнить разбор аргументов командной строки.
    args = parser.parse_args(command_line)

    data_file = args.data
    if not data_file:
        data_file = os.environ.get("STUDENTS_DATA")
    if not data_file:
        print("The data file name is absent", file=sys.stderr)
        sys.exit(1)

    # Загрузить всех студентов из файла, если файл существует.
    is_dirty = False
    if os.path.exists(data_file):
        students = load_students(data_file)
    else:
        students = []
```

Рисунок 5 – Дополненный код

```
PS C:\Users\DNS\PycharmProjects\pythonProject6\tasks> python individ.py add --name="Nikolay Bobrov" --group="1" --grade="5 5 4 5 4"
PS C:\Users\DNS\PycharmProjects\pythonProject6\tasks> python individ.py add --name="Oleg Ivanov" --group="17" --grade="5 4 3 2 5"
PS C:\Users\DNS\PycharmProjects\pythonProject6\tasks> python individ.py display
+------+------------------------------+---------------------+-----------------+
|  №   |            Ф.И.О.            |        Группа       |  Успеваемость   |
+------+------------------------------+---------------------+-----------------+
|    1 | Nikolay Bobrov               | 1                   |       5 5 4 5 4 |
|    2 | Oleg Ivanov                  | 17                  |       5 4 3 2 5 |
+------+------------------------------+---------------------+-----------------+
PS C:\Users\DNS\PycharmProjects\pythonProject6\tasks> python individ.py select -s=1
   * Nikolay Bobrov группа № 1
```

Рисунок 6 – Результат выполнения кода

6. Затем приступил к выполнению второго индивидуального задания с применением пакета dotenv.

7. Взял за основу код из лабораторной работы 2.17 с модулем click. Далее, изучив документацию о пакете dotenv, приступил к выполнению задания.

Рисунок 7 – Код к заданию с применением dotenv

8. Создал файл .env и добавил в него информацию о переменном окружении.

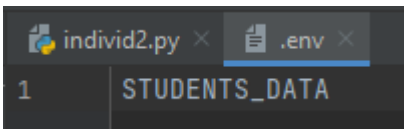

Рисунок 8 – Содержимое файла env

9. Воспользовавшись терминалом, проверил работоспособность кода.

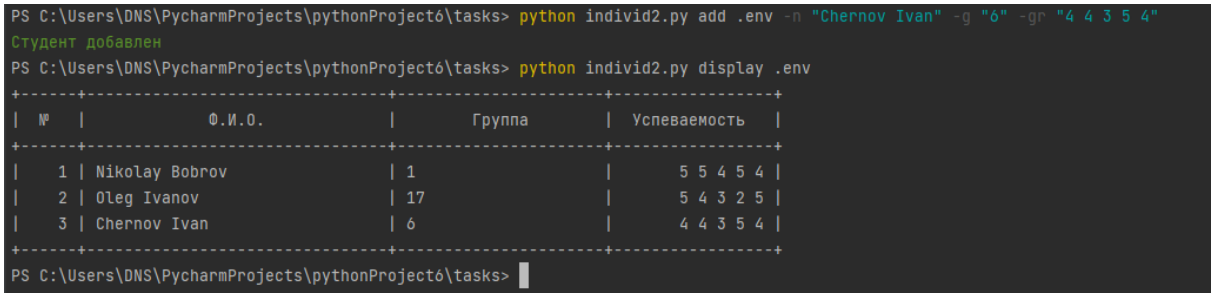

Рисунок 9 – Результат выполнения программы

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, еслионо прописано в ее значениях.

5. Как создать или изменить переменную окружения в Windows?

В окне “Переменные среды” нужно нажать на кнопку “Создать”, затемввести 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 дочернимипроцессами и оболочками.

Переменные оболочки — это переменные, которые применяются только к текущемуэкземпляру 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.

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.

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 спомощью команды set.

UID – идентификатор текущего пользователя.

11. Как установить переменные оболочки в Linux?

Чтобы создать новую переменную оболочки с именем, нужно ввести имяэтой 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 умолчаниюдля файлов модуля.

16. Какие еще переменные окружения используются для

управленияработой 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 программахна 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 переменнойокружения 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 ходе выполнения лабораторной работы приобретены навыки по работе с переменными окружениями с помощью языка программирования Python версии 3.x