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

Кафедра инфокоммуникаций

Лабораторная работа 2.18

Работа с переменными окружения в Python3

Выполнил студент группы ИВТ-б-о-20-1
Симанский М.Ю « »_____20__г.
Подпись студента_____
Работа защищена « »_____20__г.
Проверил Воронкин Р.А. _____
(подпись)

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

Изменим код для использования переменной окружения, но не дадим ей значения и получим сообщение о том, что файл пустой (рис. 1)

```
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

(base) C:\Users\rac-e\YandexDisk\Лабы 3 семестр\Технологии программирования\Лаб 20(2.18>python example2.py add -n "Сергей" -p "Директор" -y=2021
usage: workers add [-h] -n NAME [-p POST] -y YEAR filename
workers add: error: the following arguments are required: filename

(base) C:\Users\rac-e\YandexDisk\Лабы 3 семестр\Технологии программирования\Лаб 20(2.18>python example2.py add -n "Сергей" -p "Директор" -y=2021
The data file name is absent

(base) C:\Users\rac-e\YandexDisk\Лабы 3 семестр\Технологии программирования\Лаб 20(2.18>
```

Рисунок 1 – Нет данных

Добавим данные и запустим консольную команду, отличие её от обычной версии, без переменных окружения в том что не нужно вводить название файла (рис. 2).

```
(base) C:\Users\rac-e\YandexDisk\Лабы 3 семестр\Технологии программирования\Лаб 20(2.18>python individual1.py add  -n Lenta -p Maslo -pr 353

(base) C:\Users\rac-e\YandexDisk\Лабы 3 семестр\Технологии программирования\Лаб 20(2.18>python individual1.py display

(base) C:\Users\rac-e\YandexDisk\Лабы 3 семестр\Технологии программирования\Лаб 20(2.18>python individual1.py add  -n Lenta -p Maslo -pr 353
Валидация прошла успешно
Данные сохранены
```

Рисунок 2 – Файл получен

Проверим все ли данные на месте (рис. 3).

```
+------+----------------------------------+----------+----------------------+
|  No  |           Название.              |        Товар         |   Цена   |
+------+----------------------------------+----------+----------------------+
|    1 | Magnit                           | Chocolate            |       32 |
+------+----------------------------------+----------+----------------------+
|    2 | Lenta                            | Maslo                |      353 |
+------+----------------------------------+----------+----------------------+

(base) C:\Users\rac-e\YandexDisk\Лабы 3 семестр\Технологии программирования\Лаб 20(2.18>
```

Рисунок 3 – Данные на месте

Для выполнения задания повышенной сложности изменим код и добавим файл. env. Посмотрим добавились ли данные (рис. 4).

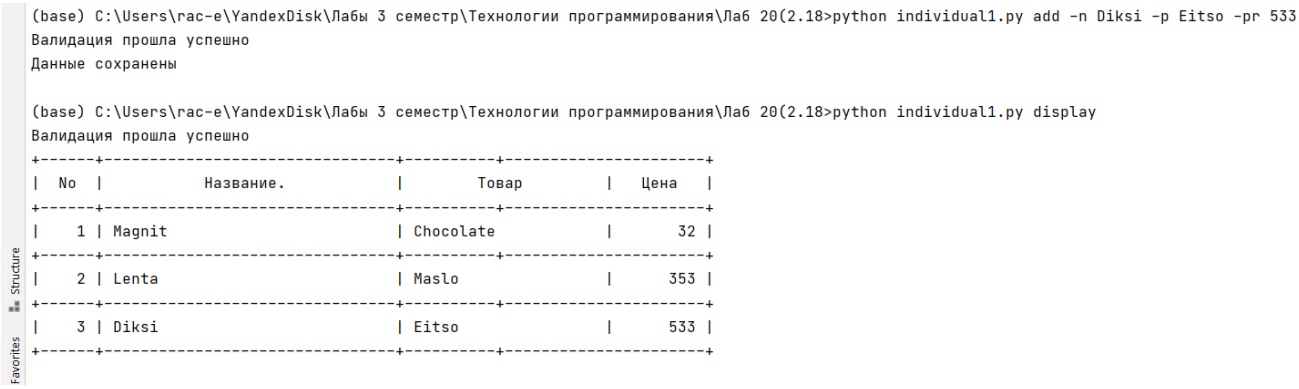

Рисунок 4 – Данные записались

Вывод: в результате выполнения работы были приобретены навыки по работе с переменными окружения с помощью языка

программирования Python версии 3.x.

Ответы на 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?

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

Получить информацию о существующих переменных можно в свойствах системы. Для этого кликаем по ярлыку

4. Каково назначение переменных PATH и PATHEXT?

Если с обычными переменными все понятно (одна ссылка – одно значение), то эти две стоятvособняком. При детальном рассмотрении видно, что они ссылаются сразу на несколько объектов.

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 процесса

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 (локали) 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).

11. Как установить переменные оболочки в Linux?

12. Как установить переменные окружения в Linux?

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?

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

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