

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

Кафедра инфокоммуникаций

Отчет по лабораторной работе № 20(2.18)

Работа с переменными окружения

по дисциплине «Технологии программирования и алгоритмизации»

Выполнил студент группы ИВТ-б-о-
20-1

Злыгостев И.С. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____

(подпись)

Ставрополь 2021

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

Ход работы:

1. Изучив методические указания, приступил к разбору примера лабораторной работы: разобрал код задания и посмотрел результат его выполнения.

```
(venv) C:\Users\zligo\git\demo-2.18\Zadanie>python primer.py add -d primer.json -n="Егоров Егор" -p="Инженер" -y=2012
```

Рисунок 20.1 – Ввод данных

```
[
  {
    "name": "Злыгостев",
    "post": "Студент",
    "year": 2020
  },
  {
    "name": "Сидоров Сидор",
    "post": "Главный инженер",
    "year": 2012
  },
  {
    "name": "Илларион Ким",
    "post": null,
    "year": 2019
  },
  {
    "name": "Егоров Егор",
    "post": "Инженер",
    "year": 2012
  }
]
```

Рисунок 20.2 – Результат выполнения

2. Затем приступил к выполнению индивидуальных заданий.
3. По примеру лабораторной работы написал код к первому заданию и проверил его работоспособность.

```
demo-2.18\Zadanie>python zadanie.py add -d zadanie.json -s Washington -v Boeing -n US-512333
```

Рисунок 20.3 – Введённые данные

```
[
  {
    "stay": "Moscow",
    "number": "RF-86231",
    "value": "Airbus"
  },
  {
    "stay": "London",
    "number": "RF-862131",
    "value": "Boeing"
  },
  {
    "stay": "Lissabon",
    "number": "RF-862357",
    "value": "Airbus"
  },
  {
    "stay": "Washington",
    "number": "US-512333",
    "value": "Boeing"
  }
]
```

Рисунок 20.4 – Результат выполнения скрипта

4. Ко второму индивидуальному заданию самостоятельно изучил библиотеку python-dotenv и приступил к написанию кода.

```

def main(command, data, stay, value, number, typing):
    if os.path.exists(data):
        load_dotenv()
        dotenv_path = os.getenv("WORKERS_DATA")
        if not dotenv_path:
            click.secho('Такого файла нет', fg='red')
            sys.exit(1)
        if os.path.exists(dotenv_path):
            flights = opening(dotenv_path)

```

Рисунок 20.5 – Использование библиотеки

```

(venv) C:\Users\zligo\git\demo-2.18\Zadanie>python zadanie2.py -c add -d .env -s Rostov -v Airbus -n EN-652131
Рейс добавлен

```

Рисунок 20.6 – Введённые данные

```

    "value": "Boeing"
  },
  {
    "stay": "Stavropol",
    "number": "FN-821333",
    "value": "Boeing"
  },
  {
    "stay": "Rostov",
    "number": "EN-652131",
    "value": "Airbus"
  }
]

```

Рисунок 20.7 – Результат выполнения кода

```

zadanie2.json x .env x zadanie2.py x
WORKERS_DATA=zadanie2.json

```

Рисунок 20.8 – Содержимое файла .env

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.

`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 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 ходе выполнения лабораторной работы были изучены основные методы по работе с переменными окружения при помощи языка программирования Python.