

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2.18
дисциплины «Анализ данных»

Выполнил:
Кенесбаев Хилол Куат улы
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

Тема: Работа с переменными окружения в Python3

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

Порядок выполнения работы:

1. Создал новый репозиторий, клонировал его, в нем создал ветку developer и перешел на нее.

2. Проработал пример лабораторной работы:

```
(lab_2.18) PS C:\Users\HP-PC\PycharmProjects\lab_2.18> cd программы
(lab_2.18) PS C:\Users\HP-PC\PycharmProjects\lab_2.18\программы> python primer.py display
+-----+-----+-----+-----+
| № |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+
|  1 | Кеесбаев                | Студент              | 2022 |
+-----+-----+-----+-----+
|  2 | Петренко                | Директор              | 2008 |
+-----+-----+-----+-----+
|  3 | Озаров                  | Зам.директора        | 2010 |
+-----+-----+-----+-----+
(lab_2.18) PS C:\Users\HP-PC\PycharmProjects\lab_2.18\программы> python primer.py add -n Магдаев -p Студент -y 2022
(lab_2.18) PS C:\Users\HP-PC\PycharmProjects\lab_2.18\программы> python primer.py select -P 5
+-----+-----+-----+-----+
| № |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+
|  1 | Петренко                | Директор              | 2008 |
+-----+-----+-----+-----+
|  2 | Озаров                  | Зам.директора        | 2010 |
+-----+-----+-----+-----+
(lab_2.18) PS C:\Users\HP-PC\PycharmProjects\lab_2.18\программы>
```

Рисунок 1. Ввод, вывод и выбор работников в консоли

3. **Выполнил индивидуальное задание** : Для своего варианта лабораторной работы 2.17 добавьте возможность получения имени файла данных, используя соответствующую переменную окружения.

```
(lab_2.18) PS C:\Users\HP-PC\PycharmProjects\lab_2.18\программы> python ind.py
>>> help
Список команд:

add - добавить знак зодиака;
list - вывести список;
select <список знаков зодиака> - запросить данные о зодиаке;
help - отобразить справку;
load - загрузить данные из файла;
save - сохранить данные в файл;
exit - завершить работу с программой.
>>> load data_individ.json
JSON валиден по схеме.
>>> list
+-----+-----+-----+-----+
| № |          Фамилия, имя          |   Знак Зодиака   |   Дата рождения   |
+-----+-----+-----+-----+
|  1 | Kenesbayev Hilol              | Kozerog          | 28.12.2002        |
|  2 | Sidorov Ivan                  | Edinorog         | 12.12.2012        |
+-----+-----+-----+-----+
>>>
```

Рисунок 2. Страницы руководства и результат работы программы

Код индивидуального задания №1:

```
import json
import os
import sys
from jsonschema import validate, ValidationError

def get_zodiac():
    """
    Запросить данные о списке
    """
    start = input("Ведите фамилию, имя ")
    finish = input("Введите знак Зодиака ")
    zodiac = (input("Введите дату рождения "))

    return {
        'start': start,
        'finish': finish,
        'zodiac': zodiac,
    }

def display_zodiac(zodiacs):
    """
    Отобразить список
    """
    if zodiacs:
        line = '+-{}--{}--{}--{}--{}--+'.format(
            '-' * 4,
            '-' * 30,
            '-' * 20,
            '-' * 14
        )
        print(line)
        print(
            '| {:^4} | {:^30} | {:^20} | {:^14} |'.format(
                "№",
                "Фамилия, имя",

```

```

        "Знак Зодиака",
        "Дата рождения"
    )
)
print(line)

for idx, worker in enumerate(zodiacs, 1):
    print(
        '| {:>4} | {:<30} | {:<20} | {:>14} |'.format(
            idx,
            worker.get('start', ''),
            worker.get('finish', ''),
            worker.get('zodiac', '')
        )
    )
    print(line)
else:
    print("Список пуст")

def select_zodiacs(zodiacs, period):
    """
    Выбрать зодиак
    """
    result = []
    for employee in zodiacs:
        if employee.get('finish') == period:
            result.append(employee)

    return result

def save_zodiacs(file_name, staff):
    """
    Сохранить данные в файл JSON
    """
    with open(file_name, "w", encoding="utf-8") as fout:
        json.dump(staff, fout, ensure_ascii=False, indent=4)

def load_zodiacs(file_name=None):
    """
    Загрузить данные из файла JSON
    """
    if file_name is None:
        file_name = os.environ.get('data_individ.json') # Get file name from
        environment variable

    if file_name is None:
        print("Файл не указан. Укажите имя файла или установите переменную
        окружения ZODIAC_FILE.", file=sys.stderr)
        return []

    schema = {
        "type": "array",
        "items": {
            "type": "object",
            "properties": {
                "start": {"type": "string"},
                "finish": {"type": "string"},
                "zodiac": {"type": "string"},
            },
            "required": [
                "start",
                "finish",
                "zodiac",
            ],
        },
    },

```

```

}
# Открыть файл с заданным именем и прочитать его содержимое.
with open(file_name, "r") as file_in:
    data = json.load(file_in) # Прочитать данные из файла

try:
    # Валидация
    validate(instance=data, schema=schema)
    print("JSON валиден по схеме.")
except ValidationError as e:
    print(f"Ошибка валидации: {e.message}")

return data # Вернуть загруженные и проверенные данные

def main():
    """
    Главная функция программы
    """
    zodiacs = []

    while True:
        command = input(">>> ").lower()
        if command == 'exit':
            break

        elif command == 'add':
            zodiac = get_zodiac()
            zodiacs.append(zodiac)
            zodiacs.sort(key=lambda item: int(item.get('zodiac',
'').split('.')[2]))

        elif command == 'list':
            display_zodiac(zodiacs)

        elif command.startswith('select'):
            parts = command.split(' ', maxsplit=1)
            period = parts[1].strip() # Получаем название знака Зодиака
            selected = select_zodiacs(zodiacs, period)
            if selected:
                display_zodiac(selected)
            else:
                print("Нет людей с таким знаком Зодиака.")

        elif command.startswith("save "):
            parts = command.split(maxsplit=1)
            file_name = parts[1]
            save_zodiacs(file_name, zodiacs)

        elif command.startswith("load "):
            parts = command.split(maxsplit=1)
            file_name = parts[1]
            zodiacs = load_zodiacs(file_name)

        elif command == 'help':
            print("Список команд:\n")
            print("add - добавить знак зодиака;")
            print("list - вывести список;")
            print("select <список знаков зодиака> - запросить данные о зодиаке;")
            print("help - отобразить справку;")
            print("load - загрузить данные из файла;")
            print("save - сохранить данные в файл;")
            print("exit - завершить работу с программой.")
        else:
            print(f"Неизвестная команда {command}", file=sys.stderr)

```

```
if __name__ == '__main__':  
    main()
```

Ответы на контрольные вопросы:

1) Каково назначение переменных окружения?

Ответ: 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.

Ответ: 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?

Ответ: Нужно открыть окно свойства системы и нажать на кнопку “Переменные среды”.

4) Каково назначение переменных PATH и PATHNEXT?

Ответ: PATH позволяет запускать исполняемые файлы и скрипты, «лежащие» в определенных каталогах, без указания их точного местоположения. PATHNEXT дает возможность не указывать даже расширение

файла, если оно прописано в ее значениях.

5) Как создать или изменить переменную окружения в Windows?

Ответ: В окне “Переменные среды” нужно нажать на кнопку “Создать”, затем ввести имя переменной и путь.

6) Что представляют собой переменные окружения в ОС Linux?

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

7) В чем отличие переменных окружения от переменных оболочки?

Ответ: Переменные окружения (или «переменные среды») – это переменные, доступные в масштабах всей системы и наследуемые всеми дочерними процессами и оболочками.

Ответ: 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?

Ответ: Наиболее часто используемая команда для вывода переменных окружения – `printenv`.

9) Какие переменные окружения Linux Вам известны? USER – текущий пользователь.

Ответ: 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 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?

Ответ: BASHOPTS – список задействованных параметров оболочки, разделенных двоеточием;

BASH_VERSION – версия запущенной оболочки bash;

COLUMNS – количество столбцов, которые используются для отображения выходных данных;

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.

UID – идентификатор текущего пользователя.

11) Как установить переменные оболочки в Linux?

Ответ: \$ NEW_VAR='значение'

12) Как установить переменные окружения в Linux?

Ответ: 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?

Ответ: Чтобы переменная сохранялась после закрытия сеанса оболочки.

14) Для чего используется переменная окружения PYTHONHOME?

Ответ: Переменная среды PYTHONHOME изменяет расположение стандартных библиотек Python.

15) Для чего используется переменная окружения PYTHONPATH?

Ответ: Переменная среды PYTHONPATH изменяет путь поиска по умолчанию для файлов модуля.

16) Как осуществляется чтение переменных окружения в программах на языке программирования Python?

Ответ: `value = os.environ.get('MY_ENV_VARIABLE')`

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

Ответ: `if os.environ[key_value]:`

18) Как присвоить значение переменной окружения в программах на языке программирования Python?

Ответ: Для присвоения значения любой переменной среды используется функция `os.environ.setdefault(«Переменная», «Значение»)`.

Вывод: в ходе выполнения лабораторной работы, приобретены навыки построения приложений с переменными окружения с помощью языка программирования Python версии 3.x.