

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 №4
дисциплины «Анализ данных»

Выполнила:
Кубанова Ксения Олеговна
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р. А.

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

Тема: работа с переменными окружения

Цель: приобрести навыки работы с переменными окружения

Порядок выполнения работы

Пример 1.

Необходимо повторить сделанный пример в лабораторной работе.

```
(labsad) C:\Users\Сергей\OneDrive\Рабочий стол\ДЗ\2 курс\анализ данных\Lab4_Ad\prog>python prim1.py add data1.json -n Kostya -p Gruzchik -y 2010
```

Рисунок 1 – используемая команда

```
data1.json > {}  
1  [  
2    {  
3      "name": "Kostya",  
4      "post": "Gruzchik",  
5      "year": 2010  
6    }  
7  ]
```

Рисунок 2 – содержимое data2.json после добавления работника через терминал

Код находится в файле prim1.py, его данные в data1.json.

Индивидуальное задание 1.

Необходимо, используя код ИДЗ прошлой лабораторной работы, улучшить его в соответствии с примером.

```
(labsad) C:\Users\Сергей\OneDrive\Рабочий стол\ДЗ\2 курс\анализ данных\Lab4_Ad\prog>python ind1.py select data.json -tm 11  
+-----+-----+-----+-----+  
| № |      Номер поезда      | Пункт назначения | Время отправления |  
+-----+-----+-----+-----+  
| 1 | 5 |      Uruguay      |      11      |  
| 2 | 5 |      Uruguay      |      12      |  
+-----+-----+-----+-----+
```

Рисунок 3 – результат работы кода

```

datajson > {} 0
1  [
2      {
3          "nomer": "5",
4          "punkt": "Urugvai",
5          "time": 11
6      },
7      {
8          "nomer": "5",
9          "punkt": "Urugvai",
10         "time": 12
11     },
12     {
13         "nomer": "5",
14         "punkt": "Urugvai",
15         "time": 6
16     }
17 ]

```

Рисунок 4 – содержимое файла data.json

Код находится в файле ind1.py, его данные в data.json.

Индивидуальное задание 2.

Необходимо изучить пакет python-dotenv и усовершенствовать работу кода ИД31 таким образом, чтобы данные подгружались благодаря использованию переменной .env.

```

import sys

from dotenv import load_dotenv

load_dotenv()

```

Рисунок 5 – импорт пакета dotenv и переменных .env

```

data_file = os.getenv('trains_data')
if not data_file:
    print("The data file name is absent", file=sys.stderr)
    sys.exit(1)

```

Рисунок 6 – изменения в коде (загрузка нужной переменной)

```

(labsad) C:\Users\Сергей\OneDrive\Рабочий стол\ДЗ\2 курс\анализ данных\Lab4_Ad\prog>python ind2.py select trains_data -tm 15

```

№	Номер поезда	Пункт назначения	Время отправления
1	8	Izrael	19
2	2	Gonkong	16

Рисунок 7 – результат работы кода с использованием переменной `trains_data`

```
1  [
2      {
3          "nomer": "4",
4          "punkt": "Moscow",
5          "time": 13
6      },
7      {
8          "nomer": "8",
9          "punkt": "Izrael",
10         "time": 19
11     },
12     {
13         "nomer": "2",
14         "punkt": "Gonkong",
15         "time": 16
16     }
17 ]
```

Рисунок 8 – содержимое файла `data2.json` для ИД32

Код находится в файле `ind2.py`, его данные в `data2.json`.

Контрольные вопросы

1 Каково назначение переменных окружения?

Назначение переменных окружения - это хранение информации о конфигурации системы или пользовательских предпочтениях, доступной для всех процессов в рамках сеанса или сеансов.

2 Какая информация может храниться в переменных окружения?

В переменных окружения может храниться различная информация, такая как пути к исполняемым файлам, настройки локали, аутентификационные данные и другие параметры, необходимые для работы программ или системы.

3 Как получить доступ к переменным окружения в ОС Windows?

Для доступа к переменным окружения в ОС Windows можно использовать команду `set` в командной строке или обратиться к ним через API в различных языках программирования, таких как Python или C#.

4 Каково назначение переменных `PATH` и `PATHEXT`?

Переменная PATH содержит список папок, в которых операционная система ищет исполняемые файлы, а переменная PATHEXT содержит список расширений файлов, которые интерпретируются как исполняемые файлы без указания расширения.

5 Как создать или изменить переменную окружения в Windows?

Для создания или изменения переменной окружения в Windows можно воспользоваться инструментами управления системой в разделе "Свойства системы" или использовать команду set в командной строке.

6 Что представляют собой переменные окружения в ОС Linux?

В ОС Linux переменные окружения представляют собой пары "имя=значение", которые определяют окружение процесса, включая такие вещи, как пути поиска, язык и настройки кодировки.

7 В чем отличие переменных окружения от переменных оболочки?

Переменные окружения отличаются от переменных оболочки тем, что первые доступны для всех процессов в системе, в то время как переменные оболочки существуют только в пределах определенной оболочки и не доступны для других процессов.

8 Как вывести значение переменной окружения в Linux?

Для вывода значения переменной окружения в Linux можно использовать команду echo \$VARIABLE_NAME, где VARIABLE_NAME - это имя переменной окружения.

9 Какие переменные окружения Linux Вам известны?

Некоторые известные переменные окружения в Linux включают PATH, HOME, USER, LANG, LC_ALL и т. д.

10 Какие переменные оболочки Linux Вам известны?

Некоторые известные оболочки Linux включают Bash, Zsh, Fish, Ksh и другие.

11 Как установить переменные оболочки в Linux?

Для установки переменных оболочки в Linux можно использовать команду `export` в оболочке или добавить их в соответствующий файл настройки оболочки, такой как `.bashrc` или `.zshrc`.

12 Как установить переменные окружения в Linux?

Для установки переменных окружения в Linux также можно использовать команду `export` в терминале или добавить их в файлы настройки окружения, такие как `/etc/environment` или файлы в каталоге `/etc/profile.d/`.

13 Для чего необходимо делать переменные окружения Linux постоянными?

Сделать переменные окружения Linux постоянными необходимо для того, чтобы они были доступны при каждом запуске новой сессии или нового процесса. Это особенно полезно для переменных, которые используются широко и часто.

14 Для чего используется переменная окружения PYTHONHOME ?

Переменная окружения `PYTHONHOME` используется для указания директории, в которой находится установка интерпретатора Python.

15 Для чего используется переменная окружения PYTHONPATH ?

Переменная окружения `PYTHONPATH` используется для указания путей поиска модулей Python.

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

Для управления работой интерпретатора Python также используются переменные окружения, такие как `PYTHONSTARTUP`, `PYTHONIOENCODING`, `PYTHONOPTIMIZE` и другие.

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

В программах на языке программирования Python чтение переменных окружения осуществляется через модуль `os` с помощью функции `os.getenv()` или `os.environ`

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

Для проверки, установлено ли значение переменной окружения в программах на языке программирования Python, можно воспользоваться методом `os.getenv()` и проверить, равно ли возвращаемое значение `None`.

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

Для присвоения значения переменной окружения в программах на языке программирования Python можно использовать методы `os.putenv()` или `os.environ`.

Вывод: в ходе выполнения лабораторной работы были приобретены навыки работы с переменными окружения.