

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

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЁТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №4**  
**дисциплины «Анализ данных»**  
**Вариант 28**

Выполнил:  
Репкин Александр Павлович  
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., канд. техн. наук,  
доцент кафедры инфокоммуникаций

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2024 г.

Рисунок 3. Получаемый файл с переменными среды

2. Выполнен пример лабораторной работы – модифицирован пример №1 из Лабораторной работы №3 дисциплины “Анализ данных”: в примере создаётся словарь, хранящий информацию о сотрудниках – ФИО, должность, год поступления на работу. Обработываемые команды: вывод всех сотрудников (list), вывод сводки о имеющихся командах (help), добавление сотрудника (add), вывод сотрудников относительно требуемой продолжительности работы (select “срок”), выход из программы (exit), сохранение данных о сотрудниках в файл (save), загрузка данных о сотрудниках из файла (load). Помимо интерфейса командной строки, к данной программе также добавлена возможность получения имени файла данных при помощи переменной окружения.

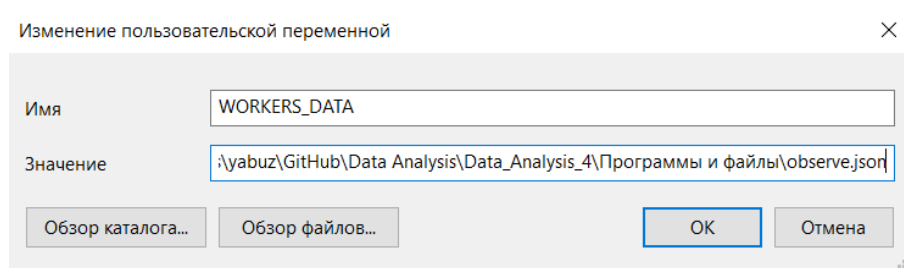


Рисунок 4. Переменная окружения WORKERS\_DATA

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import argparse
5  import json
6  import os
7  import sys
8  from datetime import date
9
10
11 def add_worker(staff, name, post, year):
12     """
13     Добавить данные о работнике.
14     """
15     staff.append({
16         "name": name,
17         "post": post,
18         "year": year
19     })
20     return staff
21
22
23 def display_workers(staff):
24     """
25     Отобразить список работников.
26     """
27     # Проверить, что список работников не пуст.
28     if staff:
29         # Заголовок таблицы.
30         line = '+-{}-+-{}-+-{}-+-{}-+'.format(
31             '-' * 4,
32             '-' * 30,
33             '-' * 20,
34             '-' * 8
35         )
36         print(line)
37         print('{:^4} | {:^30} | {:^20} | {:^8}'.format(
```

Рисунок 5. Код примера

```

(Data Analysis) C:\Users\yabuz\GitHub\Data Analysis\Data_Analysis_4\Программы и файлы>python Example_1.py
Traceback (most recent call last):
  File "C:\Users\yabuz\GitHub\Data Analysis\Data_Analysis_4\Программы и файлы\Example_1.py", line 196, in <module>
    main()
  File "C:\Users\yabuz\GitHub\Data Analysis\Data_Analysis_4\Программы и файлы\Example_1.py", line 162, in main
    data_file = args.data
    ^^^^^^^^^
AttributeError: 'Namespace' object has no attribute 'data'

(Data Analysis) C:\Users\yabuz\GitHub\Data Analysis\Data_Analysis_4\Программы и файлы>python Example_1.py -h
usage: workers [-h] [--version] {add,display,select} ...

positional arguments:
  {add,display,select}
    add                Add a new worker
    display            Display all workers
    select            Select the workers

options:
  -h, --help            show this help message and exit
  --version            show program's version number and exit

(Data Analysis) C:\Users\yabuz\GitHub\Data Analysis\Data_Analysis_4\Программы и файлы>python Example_1.py display

```

№	Ф.И.О.	Должность	Год
1	Кишки К.К.	Доставщик	1989
2	Хаку К.	Спасатель	2001
3	Бенедикт Б.	Почтальон	1910

```

(Data Analysis) C:\Users\yabuz\GitHub\Data Analysis\Data_Analysis_4\Программы и файлы>

```

Рисунок 6. Выполнение примера не требует указания имени файла

3. Выполнено первое индивидуальное задание – модифицировано задание из Лабораторной работы №3 дисциплины “Анализ Данных”: в задании создаётся словарь, хранящий информацию о людях – фамилия, имя, телефонный номер, дата рождения. Требовалось обрабатывать несколько команд – вывод всех людей (list), вывод сводки о имеющихся командах (help), добавление человека (add), вывод людей относительно требуемого месяца рождения (select “месяц”), выход из программы (exit). Данные о внесённых в список людях сохраняются и могут быть считаны в файл формата JSON (Команды save и load соответственно). Убрана необходимость обязательного ввода имени файла путём добавления переменной окружения.

```

67
68 def main(command_line=None):
69     """Главная функция программы."""
70     # Создать родительский парсер для определения имени файла.
71     file_parser = argparse.ArgumentParser(add_help=False)
72     file_parser.add_argument(
73         "-f",
74         "--filename",
75         required=False,
76         action="store",
77         help="The name of data file."
78     )
79     # Создать основной парсер командной строки.
80     parser = argparse.ArgumentParser("people")
81     parser.add_argument(
82         "--version",
83         action="version",
84         version=f"%(prog)s 0.1.0"
85     )
86     subparsers = parser.add_subparsers(dest="command")
87     # Создать субпарсер для добавления человека.
88     add = subparsers.add_parser(
89         "add",
90         parents=[file_parser],
91         help="Add a record about new human."
92     )
93     add.add_argument(
94         "-n",
95         "--name",
96         action="store",
97         required=True,
98         help="The human's name."
99     )
100     add.add_argument(
101         "-s",
102         "--surname",
103         action="store",

```

Рисунок 7. Код индивидуального задания №1



```
(Data_Analysis) C:\Users\yabuz\GitHub\Data_Analysis\Data_Analysis_4\Программы и файлы>python Indiv_Task_2.py
Traceback (most recent call last):
  File "C:\Users\yabuz\GitHub\Data_Analysis\Data_Analysis_4\Программы и файлы\Indiv_Task_2.py", line 181, in <module>
    main()
  File "C:\Users\yabuz\GitHub\Data_Analysis\Data_Analysis_4\Программы и файлы\Indiv_Task_2.py", line 146, in main
    file_name = args.filename
                ^^^^^^^^^^^^^
AttributeError: 'Namespace' object has no attribute 'filename'

(Data_Analysis) C:\Users\yabuz\GitHub\Data_Analysis\Data_Analysis_4\Программы и файлы>python Indiv_Task_2.py -h
usage: people [-h] [--version] {add,display,select} ...

positional arguments:
  {add,display,select}
    add                Add a record about new human.
    display             Display all people.
    select             Select people according to their birth's month.

options:
  -h, --help            show this help message and exit
  --version             show program's version number and exit

(Data_Analysis) C:\Users\yabuz\GitHub\Data_Analysis\Data_Analysis_4\Программы и файлы>python Indiv_Task_2.py display
```

№	Name	Surname	Telephone	Birthday
1	Suzuki	Satoru	4000000000	07.07.2015
2	Alebrije	Wisdom	9999999999	01.11.2007
3	Angus	Bambi	30403040304	14.06.2011

Рисунок 11. Пример выполнения программы

## 5. Ответы на 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.

`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.

`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?

`$ NEW_VAR='значение'`

**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?

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

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

```
if os.environ[key_value]:
```

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

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.