

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»**  
**Вариант 9**

Выполнил:  
Дудкин Константин Александрович  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника»  
направление «Программное  
обеспечение средств вычислительной  
техники и автоматизированных  
систем»,  
очная форма обучения

---

(подпись)

Руководитель практики:  
Кандидат технических наук, доцент  
кафедры инфокоммуникаций, доцент  
Воронкин Роман Александрович

---

(подпись)

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

Ставрополь, 2024 г.

Цель: Приобрести навыки по работе с переменными окружения с помощью языка программирования Python версии 3.x.

```

(base) [code_raider@archlinux ~]$ printenv
SHELL=/usr/bin/bash
SESSION_MANAGER=local/archlinux:0/tmp/.ICE-unix/3751:unix/archlinux:/tmp/.ICE-unix/3751
COLORTERM=truecolor
XDG_MENU_PREFIX=gnome-
TERM_PROGRAM_VERSION=46.0
CONDA_EXE=/home/code_raider/anaconda3/bin/conda
_CVE_M=
SSH_AUTH_SOCK=/run/user/1000/gcr/ssh
MEMORY_PRESSURE_WRITE=c20125AYhDAwMDAgHjAWMDAwMAA=
DESKTOP_SESSION=gnome
PWD=/home/code_raider
LOGNAME=code_raider
XDG_SESSION_DESKTOP=gnome
XDG_SESSION_TYPE=wayland
CONDA_PREFIX=/home/code_raider/anaconda3
SYSTEMD_EXEC_PID=54415
XAUTORETTY=/run/user/1000/.mutter-Xwaylandauth.I4VND2
NOTO_SHOWNIPAM
GDM_LANG=ru_RU.UTF-8
HOME=/home/code_raider
USERNAME=code_raider
LANG=en_US.UTF-8
XDG_CURRENT_DESKTOP=GNOME
MEMORY_PRESSURE_WATCH=sys/fs/cgroup/user.slice/user-1000.slice/user@1000.service/app.slice/app-dbus+2d:1.5+xdorg.gnome.Console.slice/dbus-1:1.5-org.gnome.Console@0.service/memory.pressure
VTE_VERSION=7601
WAYLAND_DISPLAY=wayland-0
CONDA_PROMPT_MODIFIER=(base)
INVOCATION_ID=bbf3cd7ee2b407ad7bfbf807b9af13c
MANAGERPID=3636
GNOME_SETUP_DISPLAY=:1
XDG_SESSION_CLASS=user
TERM=xterm-256color
_CVE_CONDA=
ASDF_DIR=/home/code_raider/.asdf
VIER=code_raider
CONDA_SHLVL=1
DISPLAY=:0
SHLVL=1
LC_MESSAGES=C
WINEDEBUG=-all
CONDA_PYTHON_EXE=/home/code_raider/anaconda3/bin/python
LC_CTYPE=en_US.UTF-8
XDG_RUNTIME_DIR=/run/user/1000
HAXE_STD_PATH=/usr/share/haxe/std
CONDA_DEFAULT_ENV=base
DEBUGINFOD_URLS=https://debuginfod.archlinux.org
JOURNAL_STREAM=8:368374
XDG_DATA_DIRS=/home/code_raider/.local/share/flatpak/exports/share:/var/lib/flatpak/exports/share:/usr/local/share:/usr/share
PATH=/home/code_raider/anaconda3/bin:/home/code_raider/.asdf/shims:/home/code_raider/.asdf/bin:/usr/concabins:/local/bin:/usr/bin:/usr/local/sbin:/var/lib/flatpak/exports/bin:/usr/lib/jvm/default/bin:/usr/bin/site_perl:/usr/bin/vendor_perl:/usr/bin/core_perl:/home/code_raider/.clearnweep
GMSSESSION=gnome
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
WALL=/var/spool/mail/code_raider
OLDPWD=/home/code_raider/.zshrc

```

```
TERM_PROGRAM=k9x
_=/usr/bin/printenv
```

Проработал пример 1:

```
(base) [code_raider@archlinux Progs]$ python example.py add -d="example.json" -n="Сидоров Сидор" -p="Главный инженер" -y=2012
(base) [code_raider@archlinux Progs]$ ~
```

```
[
  {
    "name": "Сидоров Сидор",
    "post": "Главный инженер",
    "year": 2012
  }
]
```

Рисунок 3. Результат работы примера

Выполнил индивидуальное задание 1: Необходимо в программу из прошлой лабораторной работы добавить возможность получения имени файла данных, используя соответствующую переменную окружения

```
# Список маршрутов
routes = []

filename = os.getenv('ROUTES_FILE', args.filename)

# Загрузить маршруты из файла, если файл существует.
if os.path.exists(args.filename):
    routes = import_json(args.filename)

# Добавить маршрут.
if args.command == "add":
    add_route(routes, args.first, args.second)
    if filename:
        export_to_json(filename, routes)

# Показать список маршрутов.
elif args.command == "list":
    list_of_routes(routes)
```

Рисунок 4. Измененная часть кода

```
(base) [code_raider@archlinux Progs]$ python individual.py add --first="Кисловодск" --second="Пятигорск" individual.json
(base) [code_raider@archlinux Progs]$ python individual.py list individual.json
```

Номер маршрута	Место отправки	Место прибытия
1	Ставрополь	Ессентуки
2	Пятигорск	Москва
3	Кисловодск	Пятигорск

Рисунок 5. Результат работы команды

Выполнил индивидуальное задание 2: Нужно в программе индивидуального задания 1 добавить библиотеку dotenv

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import argparse
import json
import os.path
from dotenv import load_dotenv

# Необходимо добавить в данную работу
# библиотеку dotenv и сделать так, чтобы программа
# находила и считывала соответствующие данные с переменных окружения

load_dotenv()
```

Рисунок 6. Измененная часть кода

### Ответы на вопросы

1. Переменные окружения необходимы для передачи информации процессам, запущенным в оболочке. Они обычно нужны для настройки поведения программ и назначения им путей к исполняемым файлам
2. Эти переменные обычно хранят в себе данные о среде ОС. Таковой, например, является системная информация и расположение системных каталогов
3. Для этого в окне «Свойства системы» необходимо открыть «Переменные среды»
4. PATH — переменная, позволяющая запускать программы без указания точного расположения
- PATHNEXT — переменная, позволяющая не указывать расширение файла при запуске какой-либо программы или при открытии файла
5. Для этого в окне «Переменные среды» есть кнопка «Создать» или «Изменить»
6. Они представляют собой набор значений, используемый для настройки ОС и конфигурации программ

7. Переменные окружения и оболочки отличаются их областью использования — первые могут передаваться по всей системе, в то время как вторые локальны и могут быть использованы только внутри оболочки

8. Воспользоваться командой `printenv` с соответствующей переменной

9. `PWD` — переменная текущей директории

`USER` — переменная текущего пользователя

`HOME` — переменная домашней директории

`PATH` — переменная пути к каталогам

`LANG` — переменная настроек языка

`MAIL` — переменная местонахождения почты

`SHELL` — переменная оболочки пользователя

10. `HOSTNAME` — переменная имени текущего хоста

`UID` — переменная User ID

`BASH_VERSION` — переменная текущей версии `bash`

`BASHOPTS` — переменная задействованных `bash` параметров

11. Для этого можно использовать команду `NEW_VAR="value"`

12. Для этого необходимо экспортировать переменную оболочки командой `export VAR="value"`

13. В отличие от переменных, даваемые программами, переменные окружения не стираются после использования или перезапуска ОС, что позволяет их использовать для настройки системы

14. `PYTHONHOME` используется для указания расположения компонентов и стандартных библиотек Python

15. `PYTHONPATH` используется для указания пути к дополнительным библиотекам Python

16. `PYTHONSTARTUP`, `PYTHONUSERBASE`, `PYTHONFAULTHANDLER`, `PYTHONDEBUG`, `PYTHONBREAKPOINT` и т. д.

17. С помощью команды `os.environ.get()`

18. Для этого можно просто сделать программу для проверки:

```
import os
if „VAR“ in os.environ:
    print(„VAR exists“)
else:
    print(„VAR doesnt exist“)
```

19. Для этого можно воспользоваться `os.environ[„VAR“] = 'value'`

Вывод: В ходе данной лабораторной работы были приобретены навыки работы с переменными окружения на Python 3.x.