

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**  
**«Программирование на языке Python»**

Выполнил:  
Быковская Стефания Станиславовна  
2 курс, группа ИТС-б-о-22-1,  
11.03.02 «Инфокоммуникационные  
технологии и системы связи», очная  
форма обучения

---

(подпись)

Проверил:

Воронкин Р. А., доцент кафедры  
инфокоммуникаций

---

(подпись)

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

Ставрополь, 2023 г.

**Тема:** Работа с переменными окружения в Python3.

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

### **Ход работы:**

Задание 1. Создали общедоступный репозиторий на GitHub, в котором использована лицензий MIT и язык программирования Python, также добавила файл .gitignore с необходимыми правилами. Клонировала свой репозиторий на свой компьютер. Организовала свой репозиторий в соответствии с моделью ветвления git-flow, появилась новая ветка develop в которой буду выполнять дальнейшие задачи.

```
Cloning into 'Laba_2.18'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (8/8), done.
Resolving deltas: 100% (1/1), done.
```

Рисунок 1. Клонирование репозитория

Задание 2. Создала виртуальное окружение conda и активировала его, также установила необходимые пакеты isort, black, flake8, pyinputplus.

```
(base) PS C:\Users\Gaming-PC> cd C:\Users\Gaming-PC\Laba_2.18
(base) PS C:\Users\Gaming-PC\Laba_2.18> conda create -n 2.18 python=3.10
WARNING: A conda environment already exists at 'C:\Users\Gaming-PC\.conda\envs\2.18'
Remove existing environment (y/[n])? y

Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 23.1.0
  latest version: 23.9.0

Please update conda by running

    $ conda update -n base -c defaults conda

Or to minimize the number of packages updated during conda update use

    conda install conda=23.9.0

## Package Plan ##

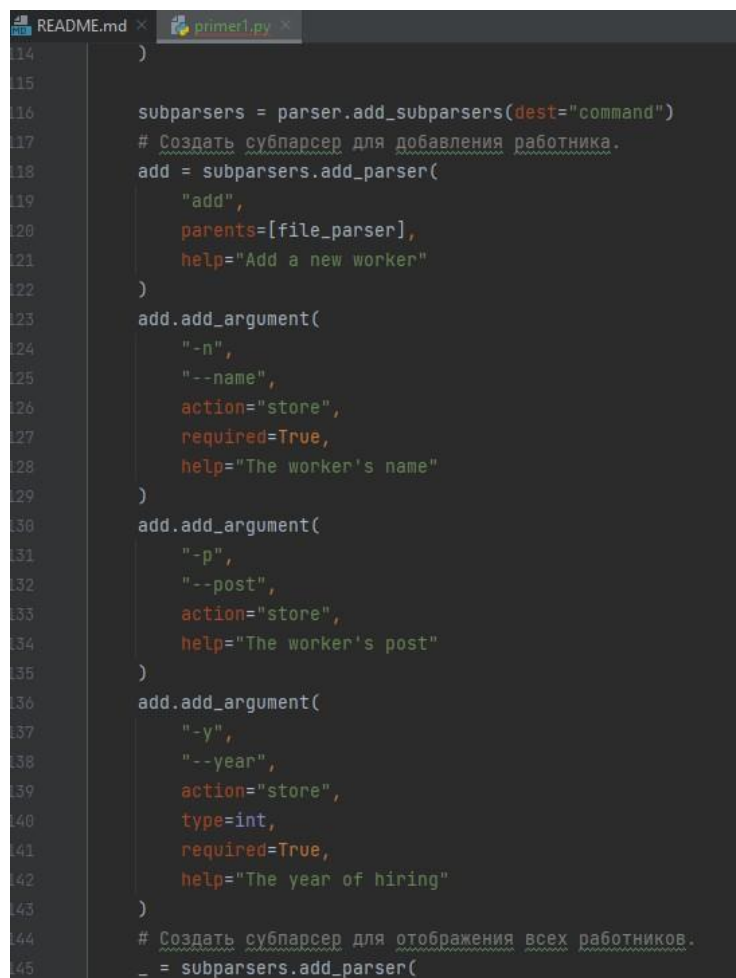
environment location: C:\Users\Gaming-PC\.conda\envs\2.18
```

Рисунок 2. Создание виртуального окружения

Задание 3. Создала проект PyCharm в папке репозитория. Приступила к работе с примером. Добавила новый файл primer1.py.

Условие примера: Для примера 1 лабораторной работы 2.17 добавьте возможность получения имени файла данных, используя соответствующую переменную окружения.

Для хранения имени файла данных будем использовать переменную окружения WORKERS\_DATA. При этом сохраним возможность передавать имя файла данных через именной параметр --data. Иными словами, если при запуске программы в командной строке не задан параметр --data, то имя файла данных должно быть взято из переменной окружения WORKERS\_DATA.



```
114 )
115
116 subparsers = parser.add_subparsers(dest="command")
117 # Создать субпарсер для добавления работника.
118 add = subparsers.add_parser(
119     "add",
120     parents=[file_parser],
121     help="Add a new worker"
122 )
123 add.add_argument(
124     "-n",
125     "--name",
126     action="store",
127     required=True,
128     help="The worker's name"
129 )
130 add.add_argument(
131     "-p",
132     "--post",
133     action="store",
134     help="The worker's post"
135 )
136 add.add_argument(
137     "-y",
138     "--year",
139     action="store",
140     type=int,
141     required=True,
142     help="The year of hiring"
143 )
144 # Создать субпарсер для отображения всех работников.
145 _ = subparsers.add_parser(
```

Рисунок 3. Пример 1

## Индивидуальное задание

### Вариант 2

Создала новый файл под названием idz.py.

Условие задания: Для своего варианта лабораторной работы 2.17 добавьте возможность получения имени файла данных, используя

соответствующую переменную окружения.

Для начала необходимо создать переменное окружение:

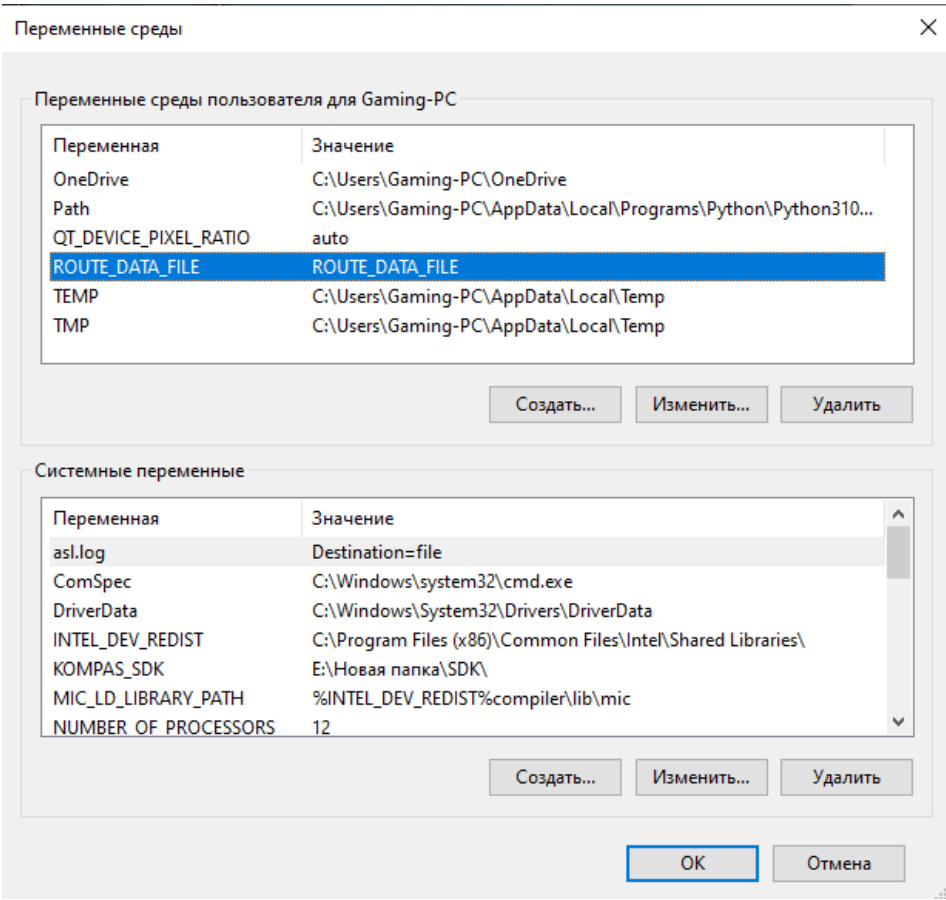


Рисунок 4. Создание переменного окружения

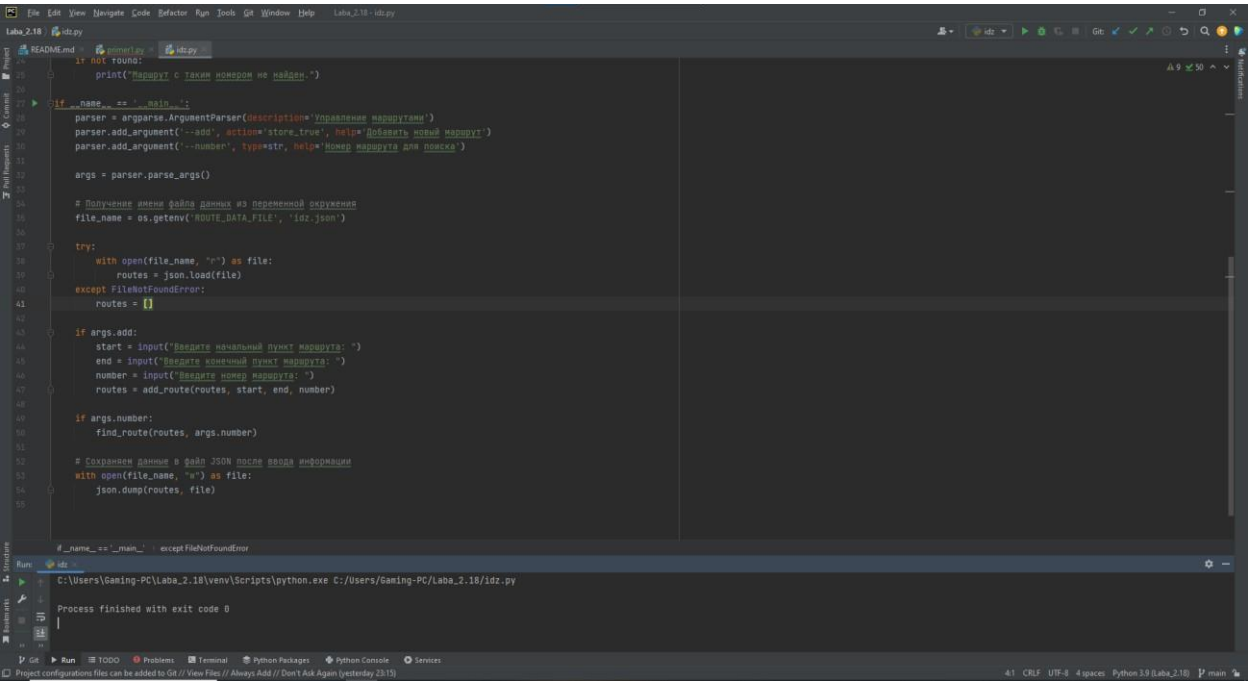


Рисунок 5. Код индивидуального задания

```

PS C:\Users\Gaming-PC\Laba_2.18> python idz.py --add
Введите начальный пункт маршрута: Орел
Введите конечный пункт маршрута: Москва
Введите номер маршрута: 2
PS C:\Users\Gaming-PC\Laba_2.18> python idz.py --add
Введите начальный пункт маршрута: Краснодар
Введите конечный пункт маршрута: Воронеж
Введите номер маршрута: 5
PS C:\Users\Gaming-PC\Laba_2.18> python idz.py --number 5
Начальный пункт маршрута: Краснодар
Конечный пункт маршрута: Воронеж

```

Рисунок 6. Результат индивидуального задания

В результате выполнения кода, если переменная окружения `ROUTE_DATA_FILE` установлена, она будет использоваться в качестве имени файла данных. Если переменная окружения не установлена, будет использоваться значение по умолчанию `"data.json"` (в данном случае установлена).

Создаем файл `idz2.py`

Условие задания: Самостоятельно изучите работу с пакетом `python-dotenv`. Модифицируйте программу задания 1 таким образом, чтобы значения необходимых переменных окружения считывались из файла `.env`.

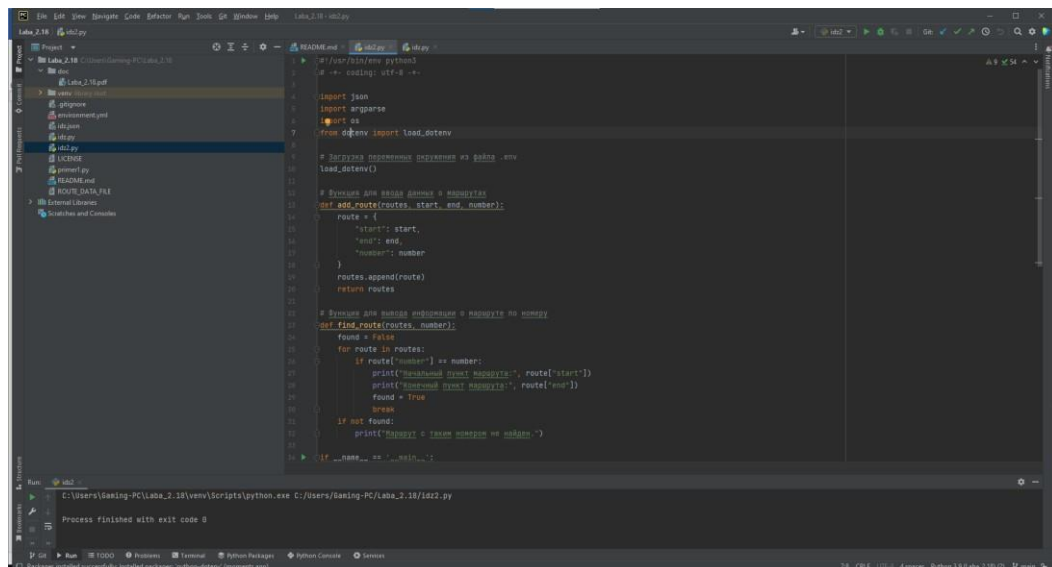


Рисунок 7. Выполнение второго индивидуального задания

#### Задание 4.

После выполнения работы на ветке `develop`, слила ее с веткой `main` и отправила изменения на удаленный сервер. Создала файл `envirement.yml` и деактивировала виртуальное окружение.

```

(2.18) PS C:\Users\Gaming-PC> conda env export > envirement.yml
(2.18) PS C:\Users\Gaming-PC> conda deactivate

```

Рисунок 7. Деактивация ВО

### Ответы на контрольные вопросы:

1. Каково назначение переменных окружения?

Переменные окружения используются для хранения информации, доступной для всех процессов, запущенных в операционной системе. Они предоставляют программам и системе информацию о конфигурации, путях поиска файлов, доступе к ресурсам, языковых настройках и многом другом.

2. Какая информация может храниться в переменных окружения?

- Пути к исполняемым файлам (например, переменная PATH).
- Конфигурационные настройки программ.
- Языковые параметры (например, LANG, LC\_ALL).
- Данные о временных директориях, пользователях и системе.
- Параметры, управляющие поведением операционной системы и программ.

3. Как получить доступ к переменным окружения в ОС Windows?

- Для получения доступа к переменным окружения в Windows можно использовать команду `echo %VARIABLE_NAME%` в командной строке, где `VARIABLE_NAME` - имя переменной.

- В окне "Свойства системы" можно просмотреть и изменить переменные окружения через панель управления.

4. Каково назначение переменных PATH и PATHEXT?

- PATH: Переменная, хранящая пути к исполняемым файлам. Она определяет, где операционная система будет искать исполняемые файлы, когда команда вводится в командной строке.
- PATHEXT: Список расширений файлов, который интерпретируется как исполняемые файлы в Windows.

5. Как создать или изменить переменную окружения в Windows?

Для создания или изменения переменной окружения в Windows можно использовать "Свойства системы" -> "Дополнительные параметры системы" -> "Переменные окружения". Можно добавить новую переменную или изменить значение существующей.

6. Что представляют собой переменные окружения в ОС Linux?

В Linux переменные окружения представляют собой параметры, хранящиеся в системе, доступные для всех процессов. Они определяют окружение, в котором запускаются процессы, включая пути поиска, языковые настройки и другие параметры.

7. В чем отличие переменных окружения от переменных оболочки?

- Переменные оболочки (shell variables) - это переменные, специфичные для конкретной оболочки и доступные только для этой оболочки.

- Переменные окружения (environment variables) - это переменные, доступные для всех процессов, запущенных в операционной системе, их значения наследуются от родительских процессов.

8. Как вывести значение переменной окружения в Linux?

В командной строке Linux можно использовать команду `echo $VARIABLE_NAME`, где `VARIABLE_NAME` - имя переменной.

9. Какие переменные окружения Linux Вам известны?

`PATH`, `HOME`, `USER`, `LANG`, `SHELL`, `PWD` и другие.

10. Какие переменные оболочки Linux Вам известны?

`PS1`, `PS2`, `HISTSIZE`, `HISTFILE` и другие, специфичные для определенных оболочек (например, `BASH`, `Zsh`).

11. Как установить переменные оболочки в Linux?

Для установки переменных оболочки в Linux используются команды экспорта переменной с ключевым словом `export` (например, `export VARIABLE_NAME=value`).

12. Как установить переменные окружения в Linux?

Переменные окружения устанавливаются в Linux также, как и переменные оболочки, но они будут доступны для всех процессов. Эти переменные часто устанавливаются в файлах конфигурации системы, таких как `.bashrc`, `.bash_profile`, `/etc/environment`, и т. д.

13. Для чего необходимо делать переменные окружения Linux постоянными?

Переменные окружения могут быть установлены постоянно, добавив их

в файлы инициализации оболочки, такие как `.bashrc` или `.bash_profile` в домашнем каталоге пользователя.

14. Для чего используется переменная окружения `PYTHONHOME`?

`PYTHONHOME` - это переменная окружения Python, которая определяет базовый каталог установки Python.

15. Для чего используется переменная окружения `PYTHONPATH`?

`PYTHONPATH` - это переменная окружения Python, определяющая пути поиска Python для модулей.

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

`PYTHONHOME`, `PYTHONPATH`, `PYTHONSTARTUP`, `PYTHONCASEOK`, `PYTHONIOENCODING` и другие.

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

В Python переменные окружения можно читать с помощью модуля `os` с функцией `os.getenv()`.

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

Для проверки установленного значения переменной окружения в Python используйте функцию `os.getenv('VARIABLE_NAME')`.

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

Для установки значения переменной окружения в Python используйте `os.putenv('VARIABLE`

**Вывод:** приобрела навыки по работе с переменными окружения с помощью языка программирования Python версии 3.x.