

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

Институт цифрового развития

Кафедра инфо коммуникаций

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №2.17**  
**Дисциплины «Программирование на Python»**

Выполнил:  
Волошин Алексей Вадимович  
2 курс, группа ИТС-б-о-22-1,  
11.03.02 «Инфокоммуникационные  
технологии и системы связи»,  
направленность (профиль)  
«Инфокоммуникационные системы и  
сети», очная форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Р. А, канд. техн. наук, доцент  
кафедры инфокоммуникаций

---

(подпись)

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

Ставрополь, 2023 г.

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

### Ход работы:

1. Создал репозиторий в GitHub «<https://github.com/Alexander-its/laba-2.18>» в который добавил .gitignore, который дополнил правила для работы с IDE

PyCharm с ЯП Python, выбрал лицензию MIT, клонировал его на лок. сервер и организовал в соответствии с моделью ветвления git-flow.

2. Создал проект PyCharm в папке репозитория, проработал примеры:

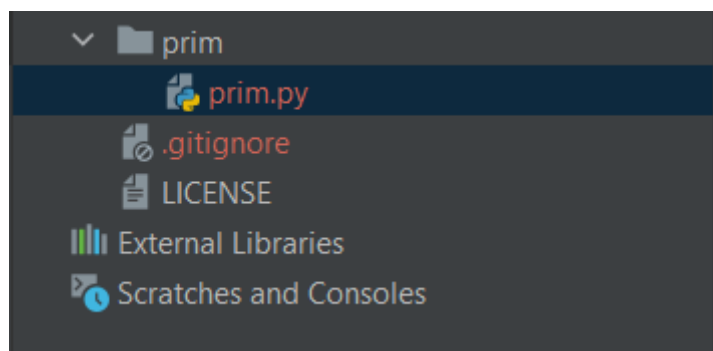


Рисунок 1 – Создание проекта в PyCharm

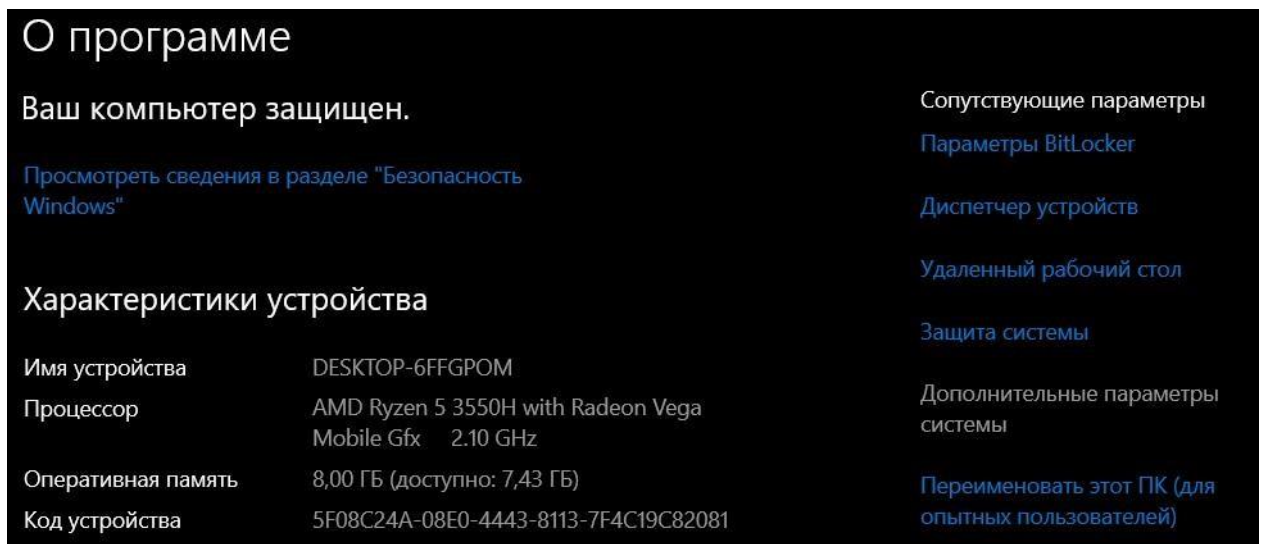


Рисунок 2 – Свойства «Этот компьютер»

No	Ф.И.О.	Должность	Год
1	asd	ert	0

Рисунок 3 – Вызов команды display без указания файла с данными

### 3. (8 вариант). Выполнил индивидуальные задания.

```
172     # Выполнить разбор аргументов командной строки.
173     args = parser.parse_args(command_line)
174
175     data_file = args.data
176     if not data_file:
177         data_file = os.environ.get("STUDENTS_DATA")
178     if not data_file:
179         print("The data file name is absent", file=sys.stderr)
180         sys.exit(1)
181
182     # Получить имя файла.
183     data_file = args.data
```

Рисунок 4 – Изменения в коде программы

№	Фамилия и имя		Знак Зодиака	Дата рождения
1	qwe	asd	ert	11/11/1111111
2	Skalette	Vito	Oven	22/11/1943
3	Bsdsaaa	ASdqw	Asdwq	02/01/1995

Рисунок 5 – Вывод программы индивидуального задания №1

```
172     # Выполнить разбор аргументов командной строки.
173     args = parser.parse_args(command_line)
174
175     data_file = args.data
176     dotenv_path = os.path.join(os.path.dirname(__file__), ".env")
177     if os.path.exists(dotenv_path):
178         load_dotenv(dotenv_path)
179     if not data_file:
180         data_file = os.getenv("HUMANS_DATA")
181     if not data_file:
182         print("The data file name is absent", file=sys.stderr)
183         sys.exit(1)
184
185     # Загрузить всех людей из файла, если файл существует.
```

Рисунок 6 – Изменения в коде программы

```
1 HUMANS_DATA|
```

Рисунок 7 – Содержимое файла .env

№	Фамилия и имя		Знак Зодиака	Дата рождения
1	qwe	asd	ert	11/11/1111111
2	Skalette	Vito	Oven	22/11/1943
3	Bsdsaaa	ASdqw	Asdwq	02/01/1995

Рисунок 8 – Вывод программы индивидуального задания №2

5. Сделал коммит, выполнил слияние с веткой main, и запустил изменения в уд. репозиторий.

```
C:\rep_2.6>git commit -m "added programs + modidied .gitignore"
[develop 2582c62] added programs + modidied .gitignore
4 files changed, 379 insertions(+), 3 deletions(-)
create mode 100644 ind.py
create mode 100644 prim.py
create mode 100644 zadaniya.py

C:\rep_2.6>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

```
C:\rep_2.6>git push
Everything up-to-date
```

Рисунок 9 – коммит и пуш изменений и переход на ветку main

```
C:\rep_2.6>git merge develop
Updating 5d4b8d1..2582c62
Fast-forward
 .gitignore | 157 ++++++
--
 ind.py      | 105 ++++++
 prim.py     | 99  ++++++
 zadaniya.py | 21  ++++++
4 files changed, 379 insertions(+), 3 deletions(-)
create mode 100644 ind.py
create mode 100644 prim.py
create mode 100644 zadaniya.py

C:\rep_2.6>_
```

Рисунок 10 – Слияние ветки main с develop

```
C:\rep_2.6>git push
info: please complete authentication in your browser...
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 5.16 KiB | 2.58 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
```

Рисунок 11 – Пуш изменений на удаленный сервер

### **Ответы на контрольные 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.

DIRSTACK – стек директорий, к которому можно применять команды pushd и popd.

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.

SHELLOPTS – параметры оболочки, которые можно устанавливать с помощью команды set.

UID – идентификатор текущего пользователя.

### **11. Как установить переменные оболочки в Linux?**

Чтобы создать новую переменную оболочки с именем, нужно ввести имя этой переменной потом знак равенства и указать значение новой переменной

### **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?**

Путём использования модуля `os`, при помощи которого программист может получить и изменить значения всех переменных среды.

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

При помощи модуля `os` можно просмотреть все переменные окружения, у которых есть значение.

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

Для присвоения значения любой переменной среды используется функция `setdefault()`.



