

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ  
ФЕДЕРАЦИИ**  
**Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**Программирование на Python**

**Отчет по лабораторной работе**

Тема: «Взаимодействие с базами данных DuckDB с помощью  
языка программирования Python»

Выполнил студент группы ИВТ-б-о-21-1

Кочкаров Умар Ахматович

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверил Воронкин Р.А. \_\_\_\_\_  
(подпись)

Ставрополь 2023

**Цель работы:** исследовать возможности взаимодействия с базами данных DuckDB с помощью языка программирования Python.


**Ход работы:**

1. Создать общедоступный репозиторий с лицензией MIT и языком Python.

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \*

 umarkochkarov ▾

Repository name \*

/ duckdb\_2

✔ duckdb\_2 is available.

Great repository names are short and memorable. Need inspiration? How about [refactored-octo-palm-tree?](#)

Description (optional)

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore


.gitignore template: Python ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: MIT License ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set  main as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

Create repository

Рисунок 1. Создание репозитория

2. Клонировать репозиторий на ПК:

```
erken@LAPTOP-ESTC60GF MINGW64 ~/Desktop/python/duckdb2
$ git clone https://github.com/umarkochkarov/duckdb_2.git
Cloning into 'duckdb_2'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

Рисунок 2. Клонирование репозитория

3. Организовать репозиторий в соответствии с моделью ветвления git-flow.

```
erken@LAPTOP-ESTC60GF MINGW64 ~/Desktop/python/duckdb2/duckdb_2 (main)
$ git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/erken/Desktop/python/duckdb2/duckdb_2/.git/hooks]
```

Рисунок 3. Организация репозитория в соответствии с моделью git-flow

4. Проработка примеров из лабораторной работы:

```
(env) C:\Users\erken\Desktop\python\duckdb2\duckdb_2\prim>python prim.py add -n="Иван Иванов" -p="Директор" -y=2015
(env) C:\Users\erken\Desktop\python\duckdb2\duckdb_2\prim>python prim.py display
```

No	Ф.И.О.	Должность	Год
1	Иван Иванов	Директор	2015

Рисунок 4. Пример

5. Индивидуальное задание. Для своего варианта лабораторной работы 2.17 необходимо реализовать хранение данных в базе данных SQLite3. Информация в базе данных должна храниться не менее чем в двух таблицах.

```
(env) C:\Users\erken\Desktop\python\duckdb2\duckdb_2\ind>python ind.py add -d="Saint-P" -n=123 -t="Скоростной"
(env) C:\Users\erken\Desktop\python\duckdb2\duckdb_2\ind>python ind.py add -d="Moscow" -n=456 -t="Грузовой"
(env) C:\Users\erken\Desktop\python\duckdb2\duckdb_2\ind>python ind.py display
```

No	Пункт назначения	Номер поезда	Тип поезда
1	Saint-P	123	Скоростной
2	Moscow	456	Грузовой

Рисунок 5. Индивидуальное задание

**Вывод:** исследованы возможности взаимодействия с базами данных DuckDB с помощью языка программирования Python.