

РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций
«Взаимодействие с базами данных SQLite3 с помощью языка
программирования Python»

Отчет по лабораторной работе № 2.21
по дисциплине «Программирование на Python»

Выполнил студент группы ИВТ-б-о-21-1

Шайдеров Дмитрий Викторович.

«20» февраля 2023г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2023

Цель работы: исследовать возможности взаимодействия с базами данных SQLite3 с помощью языка программирования Python.

Порядок выполнения работы:

1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python.

Owner * Repository name *

dshayderov / lw_2.21

Great repository names are short and memorable. Need inspiration? How about [didactic-pancake?](#)

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:
Skip this step if you're importing an existing repository.

☒ Add a README file
This is where you can write a long description for your project. [Learn more.](#)

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

.gitignore template: Python

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

License: MIT License

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

Рисунок 1 - Создание репозитория

2. Выполните клонирование созданного репозитория.

```
C:\My projects\4>git clone https://github.com/dshayderov/lw_2.21.git
Cloning into 'lw_2.21'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 11 (delta 2), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (11/11), done.
Resolving deltas: 100% (2/2), done.
```

Рисунок 2 - Клонирование репозитория

3. Организуйте свой репозиторий в соответствии с моделью ветвления git-flow.

```
C:\My projects\4\lw_2.21>git checkout -b develop
Switched to a new branch 'develop'

C:\My projects\4\lw_2.21>
```

Рисунок 3 - Ветвление по модели git-flow

4. Создание виртуального окружения.

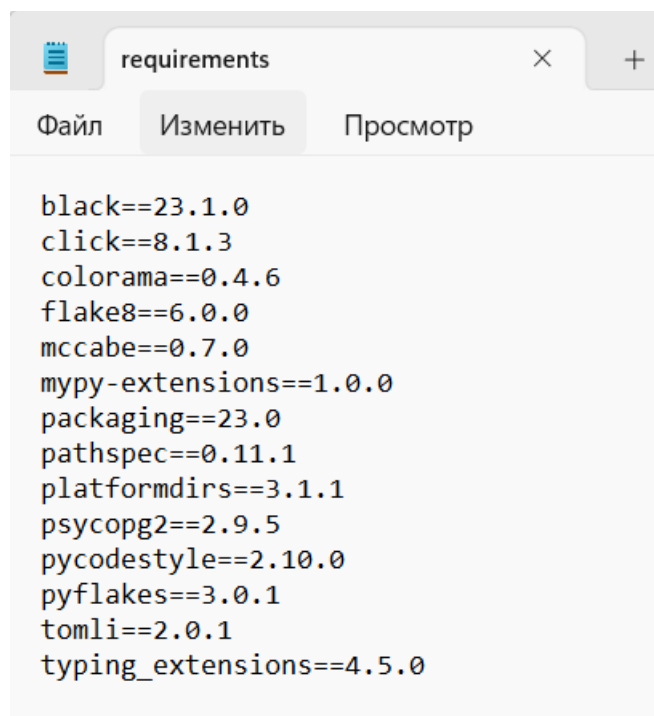
```
C:\My projects\4\lw_2.21>python -m venv .venv

C:\My projects\4\lw_2.21>.venv\Scripts\activate

(.venv) C:\My projects\4\lw_2.21>pip install black flake8 sqlite3
Collecting black
  Downloading black-23.1.0-cp39-cp39-win_amd64.whl (1.2 MB)
    | 1.2 MB 1.3 MB/s
Collecting flake8
  Using cached flake8-6.0.0-py2.py3-none-any.whl (57 kB)
```

Рисунок 4 - Виртуальное окружение

5. Формирование файла requirements.txt.



```
requirements
× +
Файл Изменить Просмотр

black==23.1.0
click==8.1.3
colorama==0.4.6
flake8==6.0.0
mccabe==0.7.0
mypy-extensions==1.0.0
packaging==23.0
pathspec==0.11.1
platformdirs==3.1.1
psycpg2==2.9.5
pycodestyle==2.10.0
pyflakes==3.0.1
tomli==2.0.1
typing_extensions==4.5.0
```

Рисунок 5 - Файл requirements.txt

6. Создайте проект PyCharm в папке репозитория.

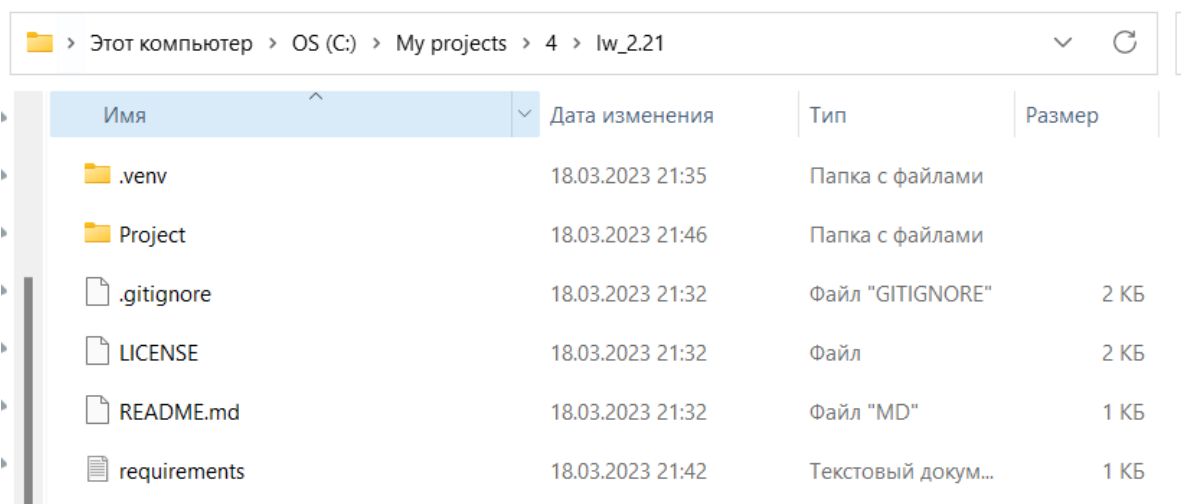


Рисунок 6 - Создание проекта

7. Проработать примеры лабораторной работы.

```
C:\Windows\System32\cmd.exe

(.venv) C:\My projects\4\lw_2.21\Project\Primer>python primer_1.py add --name="Сидоров Сидор"
--post="Главный инженер" --year=2012

(.venv) C:\My projects\4\lw_2.21\Project\Primer>python primer_1.py display
+-----+-----+-----+-----+
| No | Ф.И.О. | Должность | Год |
+-----+-----+-----+-----+
| 1 | Петров Петр | Главный бухгалтер | 2015 |
+-----+-----+-----+-----+
| 2 | Сидоров Сидор | Главный инженер | 2012 |
+-----+-----+-----+-----+

(.venv) C:\My projects\4\lw_2.21\Project\Primer>python primer_1.py select --period=10
+-----+-----+-----+-----+
| No | Ф.И.О. | Должность | Год |
+-----+-----+-----+-----+
| 1 | Сидоров Сидор | Главный инженер | 2012 |
+-----+-----+-----+-----+

(.venv) C:\My projects\4\lw_2.21\Project\Primer>
```

Рисунок 7 – Результат выполнения примера

8. Выполнить индивидуальные задания.

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.

```
C:\Windows\System32\cmd.exe

(.venv) C:\My projects\4\lw_2.21\Project\Индивидуальные задания>python ind.py display

+-----+-----+-----+-----+
| No | Пункт назначения | Номер рейса | Тип самолета |
+-----+-----+-----+-----+
| 1 | Москва | 456 | грузовой |
+-----+-----+-----+-----+
| 2 | Санкт-Петербург | 123 | пассажирский |
+-----+-----+-----+-----+

(.venv) C:\My projects\4\lw_2.21\Project\Индивидуальные задания>python ind.py select
--type="грузовой"

+-----+-----+-----+-----+
| No | Пункт назначения | Номер рейса | Тип самолета |
+-----+-----+-----+-----+
| 1 | Москва | 456 | грузовой |
+-----+-----+-----+-----+

(.venv) C:\My projects\4\lw_2.21\Project\Индивидуальные задания>
```

Рисунок 8 - Результат выполнения индивидуального задания

Задание повышенной сложности

Самостоятельно изучите работу с пакетом python-psycopg2 для работы с базами данных PostgreSQL. Для своего варианта лабораторной работы 2.17 необходимо реализовать возможность хранения данных в базе данных СУБД PostgreSQL. Информация в базе данных должна храниться не менее чем в двух таблицах.

```
(.venv) C:\My projects\4\lw_2.21\Project\Индивидуальные задания>python ind_hard.py add --destination="Санкт-Петербург" --type="пассажирский" --num=123

(.venv) C:\My projects\4\lw_2.21\Project\Индивидуальные задания>python ind_hard.py display

+-----+-----+-----+-----+
| No | Пункт назначения | Номер рейса | Тип самолета |
+-----+-----+-----+-----+
| 1 | Москва | 456 | грузовой |
+-----+-----+-----+-----+
| 2 | Санкт-Петербург | 123 | пассажирский |
+-----+-----+-----+-----+
| 3 | Санкт-Петербург | 123 | пассажирский |
+-----+-----+-----+-----+
| 4 | Санкт-Петербург | 123 | пассажирский |
+-----+-----+-----+-----+

(.venv) C:\My projects\4\lw_2.21\Project\Индивидуальные задания>python ind_hard.py select --type="грузовой"

+-----+-----+-----+-----+
| No | Пункт назначения | Номер рейса | Тип самолета |
+-----+-----+-----+-----+
| 1 | Москва | 456 | грузовой |
+-----+-----+-----+-----+
```

Рисунок 9 - Результат выполнения задания повышенной сложности

Контрольные вопросы:

1. Каково назначение модуля sqlite3?

Модуль sqlite3 предназначен для взаимодействия с СУБД SQLite.

2. Как выполняется соединение с базой данных SQLite3? Что такое курсор базы данных?

Объект соединения создается с помощью функции `connect()`. Курсор SQLite3 – это метод объекта соединения. Для выполнения инструкций SQLite3 сначала устанавливается соединение, а затем создается объект курсора с использованием объекта соединения.

3. Как подключиться к базе данных SQLite3, находящейся в оперативной памяти компьютера?

С помощью функции `:memory:`

4. Как корректно завершить работу с базой данных SQLite3?

Сначала импортируется модуль `sqlite3`, а затем определяется функция с именем `sql_connection`. Внутри функции у нас есть блок `try`, где функция `connect()` возвращает объект соединения после установления соединения. В случае возникновения ошибок при установке соединения с базой данных выполняются операторы блока `except`, в котором в данном случае просто печатается содержимое объекта ошибки. После этого вне зависимости от того возникло или нет исключение по работе с базой данных, выполняются операторы блока `finally`, в котором соединение закрывается. Заккрытие соединения необязательно, но это хорошая практика программирования, поэтому вы освобождаете память от любых неиспользуемых ресурсов.

5. Как осуществляется вставка данных в таблицу базы данных SQLite3?

Чтобы вставить данные в таблицу, используется оператор `INSERT INTO`.

6. Как осуществляется обновление данных таблицы базы данных SQLite3?

Чтобы обновить данные в таблице, просто создайте соединение, затем создайте объект курсора с помощью соединения и, наконец, используйте оператор `UPDATE` в методе `execute()`.

7. Как осуществляется выборка данных из базы данных SQLite3?

Оператор SELECT используется для выбора данных из определенной таблицы. Если вы хотите выбрать все столбцы данных из таблицы, вы можете использовать звездочку (*).

8. Каково назначение метода rowcount?

SQLite3 rowcount используется для возврата количества строк, которые были затронуты или выбраны последним выполненным SQL-запросом.

9. Как получить список всех таблиц базы данных SQLite3?

Чтобы перечислить все таблицы в базе данных SQLite3, вы должны запросить данные из таблицы sqlite_master, а затем использовать fetchall() для получения результатов из инструкции SELECT .

10. Как выполнить проверку существования таблицы как при ее добавлении, так и при ее удалении?

Чтобы проверить, не существует ли таблица уже, мы используем IFNOT EXISTS с оператором CREATE TABLE

11. Как выполнить массовую вставку данных в базу данных SQLite3?

Метод executemany можно использовать для вставки нескольких строк одновременно.

12. Как осуществляется работа с датой и временем при работе с базами данных SQLite3?

В базе данных Python SQLite3 мы можем легко хранить дату или время, импортируя модуль datetime.

Вывод: были исследованы возможности взаимодействия с базами данных SQLite3 с помощью языка программирования Python.