

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 №2.21
дисциплины «Основы кроссплатформенного программирования»

Выполнил:
Наумов Никита Викторович
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., доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

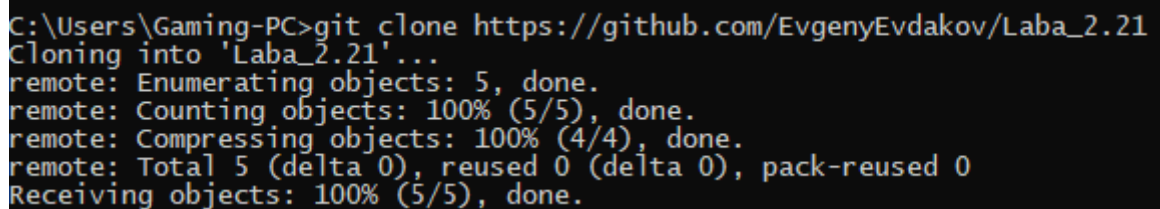
Ставрополь, 2023 г.

Тема: Взаимодействие с базами данных SQLite3 с помощью языка программирования Python

Цель: исследовать базовые возможности системы управления базами данных SQLite3 в языке Python.

Ход работы:

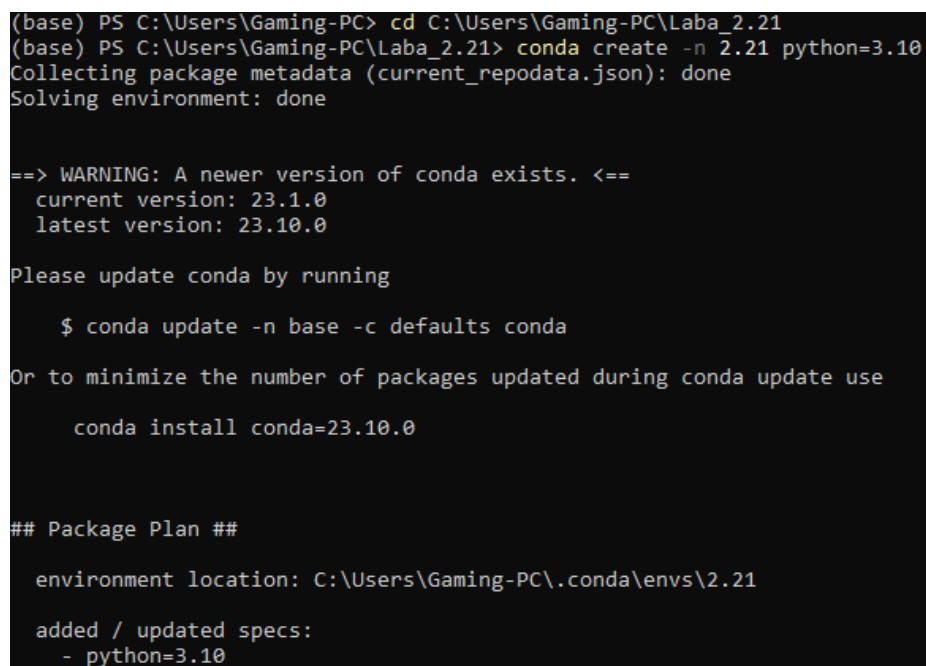
Задание 1. Создал общедоступный репозиторий на GitHub, в котором использована лицензий MIT и язык программирования Python, также добавил файл .gitignore с необходимыми правилами. Клонировал свой репозиторий на свой компьютер. Организовал свой репозиторий в соответствии с моделью ветвления git-flow, появилась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



```
C:\Users\Gaming-PC>git clone https://github.com/EvgenyEvdakov/Laba_2.21
Cloning into 'Laba_2.21'...
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 1. Клонирование репозитория

Задание 2. Создал виртуальное окружение conda и активировал его, также установил необходимые пакеты isort, black, flake8.



```
(base) PS C:\Users\Gaming-PC> cd C:\Users\Gaming-PC\Laba_2.21
(base) PS C:\Users\Gaming-PC\Laba_2.21> conda create -n 2.21 python=3.10
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 23.1.0
  latest version: 23.10.0

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=23.10.0

## Package Plan ##

  environment location: C:\Users\Gaming-PC\.conda\envs\2.21

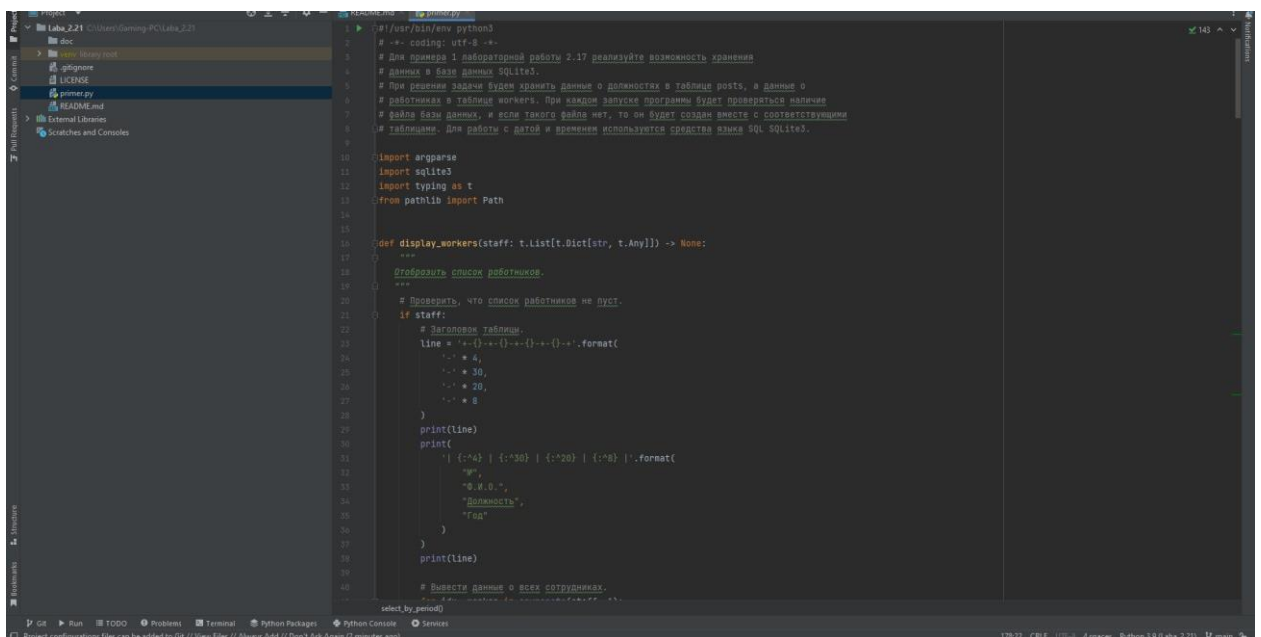
  added / updated specs:
    - python=3.10
```

Рисунок 2. Создание виртуального окружения

Задание 3. Создал проект PyCharm в папке репозитория. Приступил к работе с примером. Добавил новый файл primer.py.

Условие примера: Для примера лабораторной работы 2.17 реализуйте возможность хранения данных в базе данных SQLite3.

Решение: При решении задачи будем хранить данные о должностях в таблице posts, а данные о работниках в таблице workers. При каждом запуске программы будет проверяться наличие файла базы данных, и если такого файла нет, то он будет создан вместе с соответствующими таблицами. Для работы с датой и временем используются средства языка SQL SQLite3.



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 # Для примера 1 лабораторной работы 2.17 реализуйте возможность хранения
5 # данных в базе данных SQLite3.
6 # При решении задачи будем хранить данные о должностях в таблице posts, а данные о
7 # работниках в таблице workers. При каждом запуске программы будет проверяться наличие
8 # файла базы данных, и если такого файла нет, то он будет создан вместе с соответствующими
9 # таблицами. Для работы с датой и временем используются средства языка SQL SQLite3.
10
11 import argparse
12 import sqlite3
13 import typing as t
14 from pathlib import Path
15
16 def display_workers(staff: t.List[t.Dict[str, t.Any]]) -> None:
17     """
18     Отобразить список работников.
19     """
20     # Проверить, что список работников не пуст.
21     if staff:
22         # Заголовки таблиц.
23         line = '{:4} | {:30} | {:20} | {:8}'.format(
24             'id',
25             'name',
26             'position',
27             'date'
28         )
29         print(line)
30         print(
31             '{:4} | {:30} | {:20} | {:8}'.format(
32                 'id',
33                 'name',
34                 'position',
35                 'date'
36             )
37         )
38         print(line)
39
40     # Вывести данные о всех сотрудниках.
41     select_by_period()
```

Рисунок 3. Выполнение примера

Задание 4.

Индивидуальное задание

Вариант 10

Создал новый файл под названием idz.py.

Условие задания: 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.

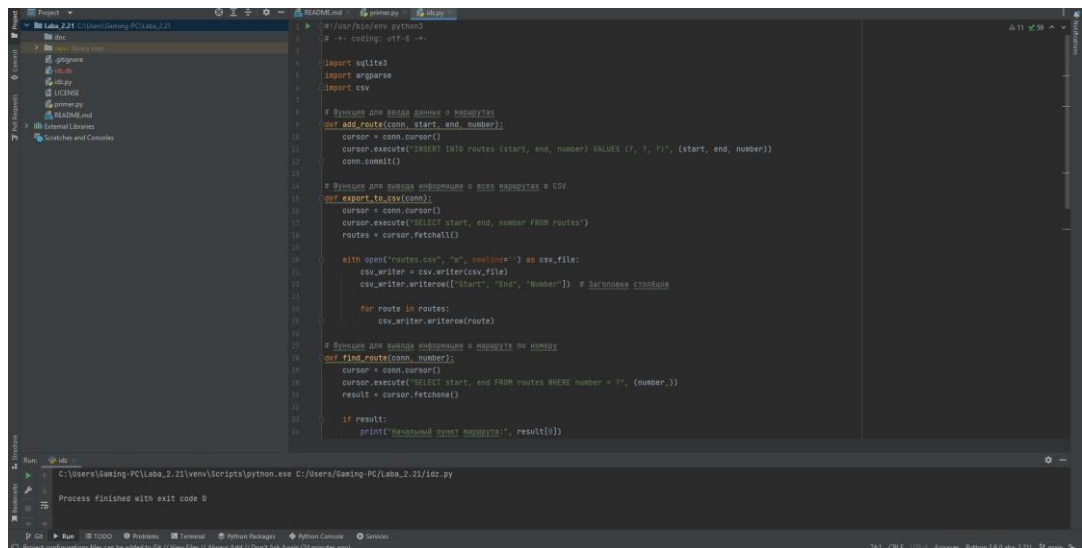


Рисунок 4. Код индивидуального задания

```

PS C:\Users\Gaming-PC> cd C:\Users\Gaming-PC\Laba_2.21
PS C:\Users\Gaming-PC\Laba_2.21> python idz.py --add
Введите начальный пункт маршрута: Белгород
Введите конечный пункт маршрута: Казань
Введите номер маршрута: 23
PS C:\Users\Gaming-PC\Laba_2.21> python idz.py --add
Введите начальный пункт маршрута: Анапа
Введите конечный пункт маршрута: Сочи
Введите номер маршрута: 67
PS C:\Users\Gaming-PC\Laba_2.21> python idz.py --add
Введите начальный пункт маршрута: Рим
Введите конечный пункт маршрута: Берлин
Введите номер маршрута: 122
PS C:\Users\Gaming-PC\Laba_2.21> python idz.py --number 67
Начальный пункт маршрута: Анапа
Конечный пункт маршрута: Сочи
PS C:\Users\Gaming-PC\Laba_2.21> python idz.py --export
PS C:\Users\Gaming-PC\Laba_2.21>

```

Рисунок 5. Результат программы

	A	B	C	D	E
1	Start,End,Number				
2	Белгород,Казань,23				
3	Анапа,Сочи,67				
4	Рим,Берлин,122				
5					
6					

Рисунок 6. Файл CSV

Задание 5.

После выполнения работы на ветке develop, слил ее с веткой main и отправил изменения на удаленный сервер. Создал файл environment.yml и деактивировал виртуальное окружение.

```
(2.21) PS C:\Users\Gaming-PC\Laba_2.21> conda env export > enviroment.yml  
(2.21) PS C:\Users\Gaming-PC\Laba_2.21> conda deactivate
```

Рисунок 7. Деактивация ВО

Ответы на контрольные вопросы:

1. Каково назначение модуля sqlite3?

Модуль sqlite3 предоставляет интерфейс для взаимодействия с базами данных SQLite из программ, написанных на языке Python. Он позволяет создавать, управлять и выполнять запросы к базам данных SQLite.

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

Соединение с базой данных SQLite3 выполняется с использованием функции sqlite3.connect(). Эта функция возвращает объект соединения. Курсор базы данных (cursor) используется для выполнения SQL-запросов и получения результатов.

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

Чтобы подключиться к базе данных SQLite3, находящейся в оперативной памяти, нужно использовать специальное имя файла базы данных ":memory:" при вызове connect.

4. Как корректно завершить работу с базой данных SQLite3?

Чтобы корректно завершить работу с базой данных SQLite3, вызовите метод close объекта соединения.

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

Вставка данных в таблицу выполняется с использованием SQL-запроса INSERT INTO.

6. Как осуществляется обновление данных таблицы базы данных SQLite3?

Обновление данных в таблице выполняется с использованием SQL-запроса UPDATE.

7. Как осуществляется выборка данных из базы данных SQLite3?

Выборка данных выполняется с использованием SQL-запроса SELECT.

8. Каково назначение метода rowcount?

Метод rowcount возвращает количество строк, затронутых последним выполненным SQL-запросом.

9. Как получить список всех таблиц базы данных SQLite3?

Используйте запрос к таблице sqlite_master.

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

Вы можете использовать условие IF NOT EXISTS при создании таблицы и запрос к sqlite_master при проверке существования.

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

Для массовой вставки данных обычно используется метод executemany.

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

SQLite3 поддерживает тип данных DATE и TIMESTAMP для хранения даты и времени. При вставке и выборке данных с использованием этих типов следует использовать соответствующий формат.

Вывод: исследовал базовые возможности системы управления базами данных SQLite3 в языке Python.