

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

Институт цифрового развития
Кафедра инфокоммуникаций

**«Взаимодействие с базами данных SQLite3 с помощью языка
программирования Python»**

ОТЧЕТ
по лабораторной работе №21
дисциплины
«Основы программной инженерии»

Выполнила:

Кувшин Ирина Анатольевна
2 курс, группа ПИЖ-б-о-21-1,
011.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил:

(подпись)

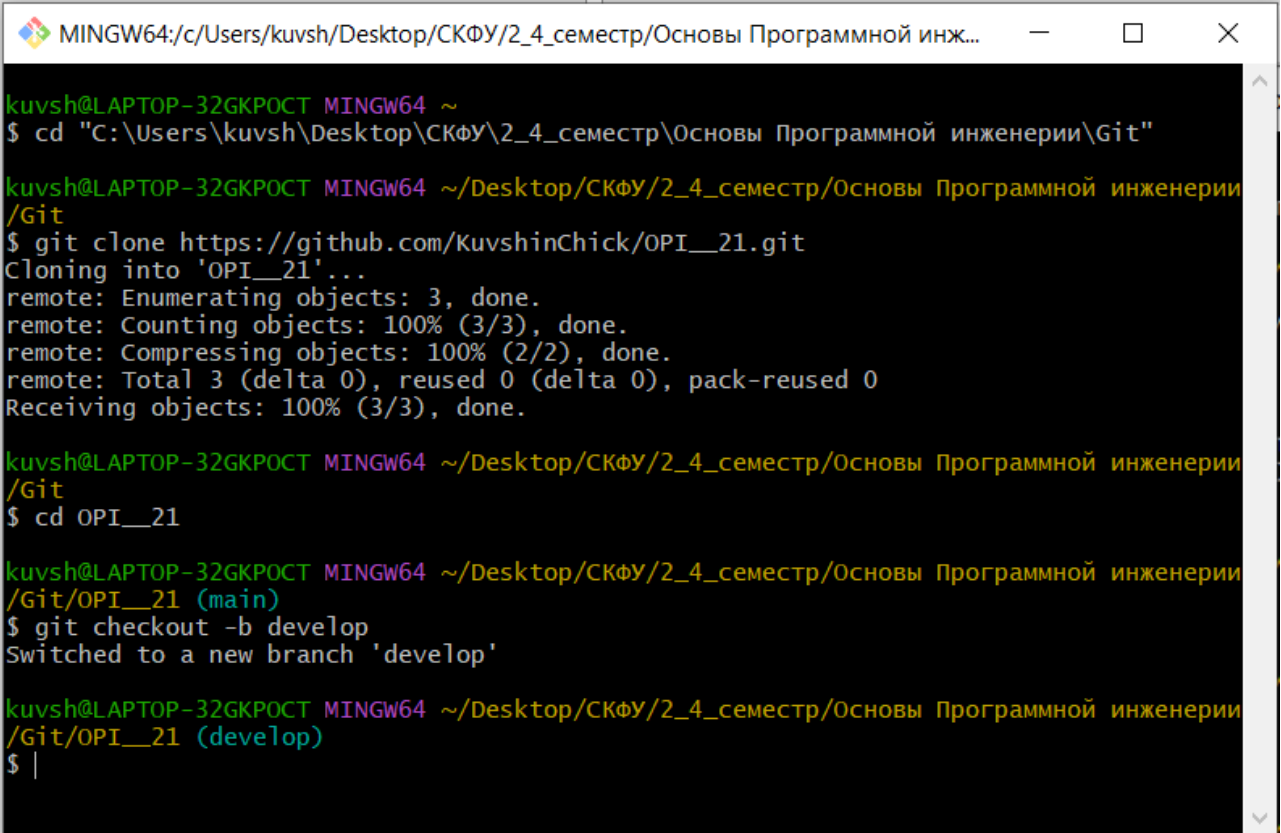
Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Цель работы: исследовать взаимодействие с базами данных SQLite3 с помощью языка программирования Python

Ход работы:

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.
3. Выполните клонирование созданного репозитория.
4. Дополните файл .gitignore необходимыми и для выбранного языка программирования и интегрированной среды разработки.
5. Организуйте свой репозиторий в соответствие с моделью ветвления git - flow.
6. Создайте проект PyCharm в папке репозитория.



```
MINGW64:/c:/Users/kuvsh/Desktop/СКФУ/2_4_семестр/Основы Программной инж...
kuvsh@LAPTOP-32GKPOCT MINGW64 ~
$ cd "C:\Users\kuvsh\Desktop\СКФУ\2_4_семестр\Основы Программной инженерии\Git"

kuvsh@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_4_семестр/Основы Программной инженерии/Git
$ git clone https://github.com/KuvshinChick/OPI__21.git
Cloning into 'OPI__21'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

kuvsh@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_4_семестр/Основы Программной инженерии/Git
$ cd OPI__21

kuvsh@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_4_семестр/Основы Программной инженерии/Git/OPI__21 (main)
$ git checkout -b develop
Switched to a new branch 'develop'

kuvsh@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_4_семестр/Основы Программной инженерии/Git/OPI__21 (develop)
$ |
```

Рисунок 21.1 – Клонирование репозитория и создание ветки develop

```
MINGW64:/c:/Users/kuvsh/Desktop/CKФУ/2_4_семестр/Основы Программной инж...
kuvsh@LAPTOP-32GKPOCT MINGW64 ~/Desktop/CKФУ/2_4_семестр/Основы Программной инженерии
/Git/OPI__21 (develop)
$ git add .

kuvsh@LAPTOP-32GKPOCT MINGW64 ~/Desktop/CKФУ/2_4_семестр/Основы Программной инженерии
/Git/OPI__21 (develop)
$ git commit -m "modified .gitignore & readme"
[develop cd3ef56] modified .gitignore & readme
2 files changed, 133 insertions(+)
create mode 100644 .gitignore
create mode 100644 README.md

kuvsh@LAPTOP-32GKPOCT MINGW64 ~/Desktop/CKФУ/2_4_семестр/Основы Программной инженерии
/Git/OPI__21 (develop)
$ git push origin develop
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 1.47 KiB | 1.47 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/KuvshinChick/OPI__21/pull/new/develop
remote:
To https://github.com/KuvshinChick/OPI__21.git
 * [new branch]      develop -> develop

kuvsh@LAPTOP-32GKPOCT MINGW64 ~/Desktop/CKФУ/2_4_семестр/Основы Программной инженерии
```

Рисунок 21.2 – Обновление .gitignore и readme

7. Проработайте примеры лабораторной работы. Создайте для них отдельные модули языка Python. Зафиксируйте изменения в репозитории.
8. Приведите в отчете скриншоты результатов выполнения примера при различных исходных данных вводимых с клавиатуры.
9. Зафиксируйте сделанные изменения в репозитории.

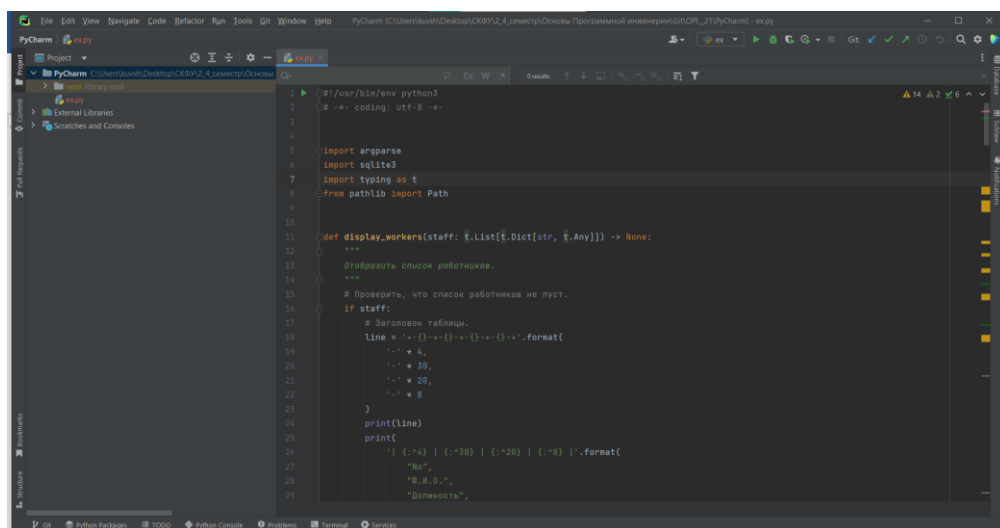


Рисунок 21.3 – Проработка программы

```
Terminal: C:\WINDOWS\system32\cmd.exe
(base) (venv) C:\Users\kuvsh\Desktop\СКФУ\2_4_семестр\Основы Программной инженерии\Git\OPI_21\PyCharm\Example>python ex.py add --name="Иванов И." --post="Программист" --year="1999"
(base) (venv) C:\Users\kuvsh\Desktop\СКФУ\2_4_семестр\Основы Программной инженерии\Git\OPI_21\PyCharm\Example>
(base) (venv) C:\Users\kuvsh\Desktop\СКФУ\2_4_семестр\Основы Программной инженерии\Git\OPI_21\PyCharm\Example>python ex.py add --name="Петров В." --post="Тестировщик" --year="2005"
(base) (venv) C:\Users\kuvsh\Desktop\СКФУ\2_4_семестр\Основы Программной инженерии\Git\OPI_21\PyCharm\Example>python ex.py display
+-----+
| No |      Ф.И.О.      |      Должность      |      Год      |
+-----+
| 1 | Иванов И.        | Программист         | 1999 |
+-----+
| 2 | Петров В.        | Тестировщик         | 2005 |
+-----+
(base) (venv) C:\Users\kuvsh\Desktop\СКФУ\2_4_семестр\Основы Программной инженерии\Git\OPI_21\PyCharm\Example>python ex.py select --period=23
+-----+
| No |      Ф.И.О.      |      Должность      |      Год      |
+-----+
| 1 | Иванов И.        | Программист         | 1999 |
+-----+
(base) (venv) C:\Users\kuvsh\Desktop\СКФУ\2_4_семестр\Основы Программной инженерии\Git\OPI_21\PyCharm\Example>
```

Рисунок 21.4 – Результат работы программы

10. Приведите в отчете скриншоты работы программ решения индивидуальных заданий.
11. Зафиксируйте сделанные изменения в репозитории.
12. Добавьте отчет по лабораторной работе в формате PDF в папку doc репозитория. Зафиксируйте изменения.
13. Выполните слияние ветки для разработки с веткой master/main.
14. Отправьте сделанные изменения на сервер GitHub.
15. Отправьте адрес репозитория GitHub на электронный адрес преподавателя.

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.

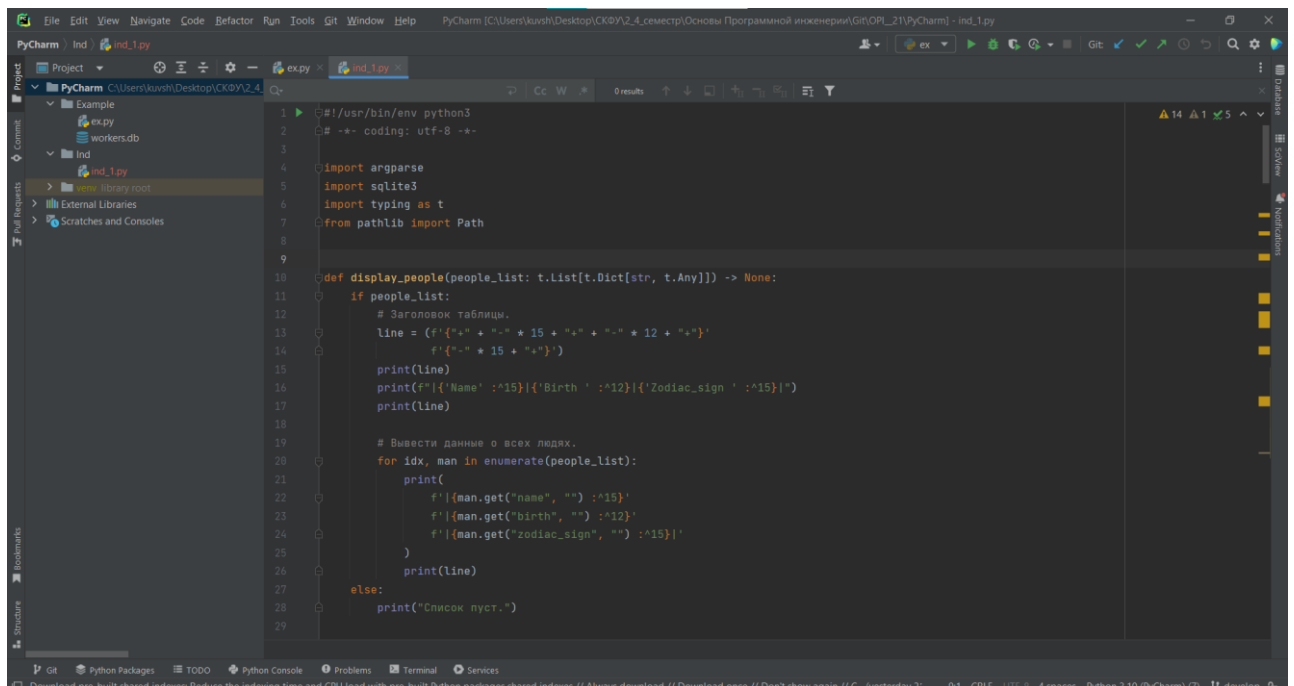


Рисунок 21.5 – Проработка программы

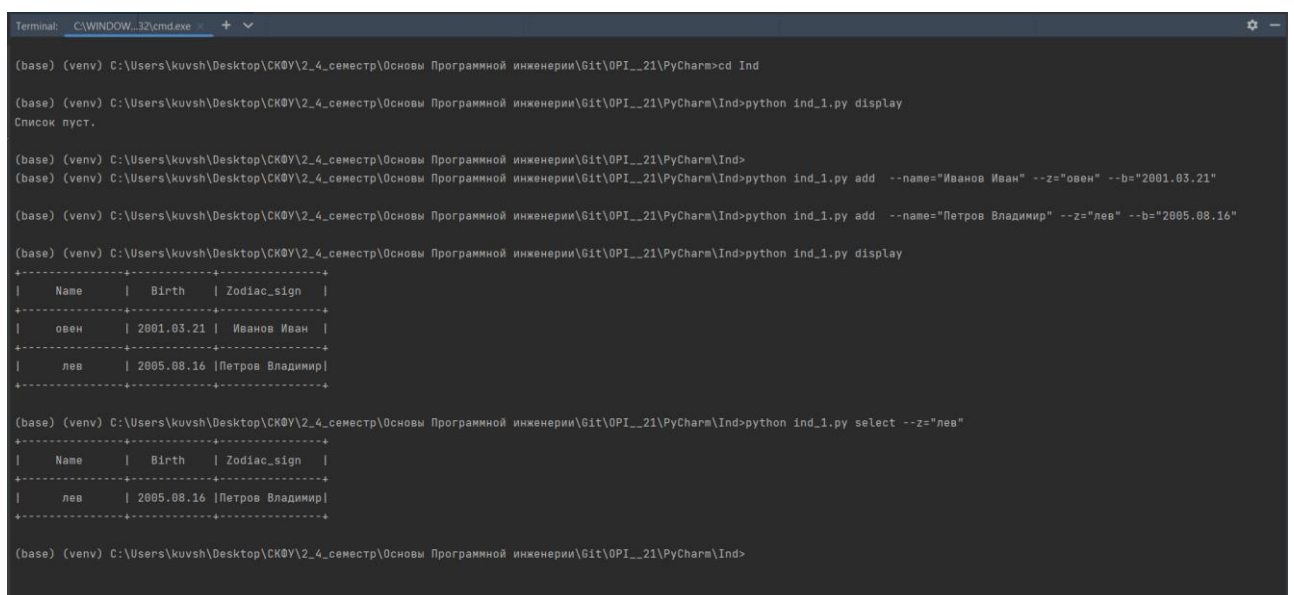


Рисунок 21.6 – Результат работы программы

Контрольные вопросы

1. Каково назначение модуля sqlite3 ?

Модуль sqlite3 предназначен для работы с базой данных SQLite3.

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

Соединение с базой данных SQLite3 выполняется через функцию `connect()`. Курсор базы данных - это объект, который используется для выполнения операций с базой данных.

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

Чтобы подключиться к базе данных SQLite3, находящейся в оперативной памяти компьютера, нужно передать имя файла `":memory:"` при вызове функции `connect()`.

```
def sql_connection():  
    try:  
        con = sqlite3.connect(':memory:')  
        print("Connection is established: Database is created in memory")
```

4. Как корректно завершить работу с базой данных SQLite3?

Для корректного завершения работы с базой данных SQLite3 нужно закрыть курсор и соединение с базой данных.

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

Данные в таблицу базы данных SQLite3 можно вставить с помощью метода `execute()`, используя SQL-запрос с командой `INSERT`.

```
import sqlite3

con = sqlite3.connect('mydatabase.db')

def sql_insert(con, entities):
    cursor_obj = con.cursor()
    cursor_obj.execute(
        """
        INSERT INTO employees(id, name, salary, department, position, hireDate)
        VALUES(?, ?, ?, ?, ?, ?)
        """,
        entities
    )
    con.commit()
```

```
entities = (2, 'Andrew', 800, 'IT', 'Tech', '2018-02-06')
sql_insert(con, entities)
```

6. Как осуществляется обновление данных таблицы базы данных SQLite3?

Данные в таблице базы данных SQLite3 можно обновить с помощью метода `execute()`, используя SQL-запрос с командой `UPDATE`.

```
import sqlite3

con = sqlite3.connect('mydatabase.db')

def sql_update(con):
    cursor_obj = con.cursor()
    cursor_obj.execute(
        "UPDATE employees SET name = 'Rogers' where id = 2"
    )

    con.commit()

sql_update(con)
```

7. Как осуществляется выборка данных из базы данных SQLite3?

Данные из базы данных SQLite3 можно получить с помощью метода `execute()`, используя SQL-запрос с командой `SELECT`.

```
import sqlite3

con = sqlite3.connect('mydatabase.db')

def sql_fetch(con):
    cursor_obj = con.cursor()
    cursor_obj.execute("SELECT * FROM employees")

    rows = cursorObj.fetchall()
    for row in rows:
        print(row)

sql_fetch(con)
```

8. Каково назначение метода `rowcount`?

Метод `rowcount` возвращает количество строк, затронутых последней операцией.

9. Как получить список всех таблиц базы данных SQLite3?

Список всех таблиц базы данных SQLite3 можно получить с помощью метода `execute()`, используя SQL-запрос с командой `SELECT`.


```
import sqlite3

con = sqlite3.connect('mydatabase.db')

def sql_fetch(con):
    cursor_obj = con.cursor()
    cursor_obj.execute(
        "SELECT name from sqlite_master where type='table'"
    )

    print(cursor_obj.fetchall())

sql_fetch(con)
```

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

Существование таблицы можно проверить с помощью метода `execute()`, используя SQL-запрос с командой `CREATE` или `DROP`.

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

Массовую вставку данных в базу данных SQLite3 можно выполнить с помощью метода `executemany()`.

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

Для работы с датой и временем в базе данных SQLite3 есть функции и определенные форматы данных. Например, можно использовать функцию `strftime()` для форматирования даты и времени.