Практическое занятие № 15 1. Наименование практического занятия: составление программ для работы с базами данных в IDE PyCharm Community.

2. Цели практического занятия: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, работы с БД в IDE PyCharm Communit

Постановка задачи: Приложение ПЛАТНАЯ ПОЛИКЛИНИКА для некоторой организации. БД должна содержать таблицу Пациент со следующей структурой записи: ФИО пациента, ФИО врача, диагноз, стоимость лечение

Тип алгоритма: линейный

Текст программ: import sqlite3

conn = sqlite3.connect('paid\_clinic.db')

cursor = conn.cursor()

cursor.execute('''

CREATE TABLE IF NOT EXISTS Пациент (

    id INTEGER PRIMARY KEY AUTOINCREMENT,

    patient\_name TEXT NOT NULL,

    doctor\_name TEXT NOT NULL,

    diagnosis TEXT NOT NULL,

    treatment\_cost REAL NOT NULL

)

''')

def add\_patient(patient\_name, doctor\_name, diagnosis, treatment\_cost):

    cursor.execute('''

    INSERT INTO Пациент (patient\_name, doctor\_name, diagnosis, treatment\_cost)

    VALUES (?, ?, ?, ?)

    ''', (patient\_name, doctor\_name, diagnosis, treatment\_cost))

    conn.commit()

def get\_all\_patients():

    cursor.execute('SELECT \* FROM Пациент')

    return cursor.fetchall()

def search\_patients(patient\_name=None, doctor\_name=None, diagnosis=None):

    query = "SELECT \* FROM Пациент WHERE 1=1"

    params = []

    if patient\_name:

        query += " AND patient\_name=?"

        params.append(patient\_name)

    if doctor\_name:

        query += " AND doctor\_name=?"

        params.append(doctor\_name)

    if diagnosis:

        query += " AND diagnosis=?"

        params.append(diagnosis)

    cursor.execute(query, params)

    return cursor.fetchall()

def delete\_patient(patient\_id):

    cursor.execute("DELETE FROM Пациент WHERE id=?", (patient\_id,))

    conn.commit()

def update\_patient(patient\_id, patient\_name=None, doctor\_name=None, diagnosis=None, treatment\_cost=None):

    query = "UPDATE Пациент SET "

    params = []

    if patient\_name:

        query += "patient\_name=?, "

        params.append(patient\_name)

    if doctor\_name:

        query += "doctor\_name=?, "

        params.append(doctor\_name)

    if diagnosis:

        query += "diagnosis=?, "

        patients.append(diagnosis)

    if treatment\_cost:

        query += "treatment\_cost=? "

        params.append(treatment\_cost)

    query = query.rstrip(", ") + " WHERE id=?"

    params.append(patient\_id)

    cursor.execute(query, params)

    conn.commit()

def main\_menu():

    print("Главное меню:")

    print("1. Добавить пациента")

    print("2. Поиск пациентов")

    print("3. Удалить пациента")

    print("4. Редактировать данные пациента")

    print("5. Вывести всех пациентов")

    print("6. Выход")

def add\_patient\_menu():

    patient\_name = input("Введите имя пациента: ")

    doctor\_name = input("Введите имя врача: ")

    diagnosis = input("Введите диагноз: ")

    treatment\_cost = float(input("Введите стоимость лечения: "))

    add\_patient(patient\_name, doctor\_name, diagnosis, treatment\_cost)

    print("Пациент успешно добавлен!")

def search\_patients\_menu():

    patient\_name = input("Введите имя пациента (или Enter для пропуска): ")

    doctor\_name = input("Введите имя врача (или Enter для пропуска): ")

    diagnosis = input("Введите диагноз (или Enter для пропуска): ")

    patients = search\_patients(patient\_name, doctor\_name, diagnosis)

    for patient in patients:

        print(patient)

def delete\_patient\_menu():

    patient\_id = int(input("Введите id пациента, которого нужно удалить: "))

    delete\_patient(patient\_id)

    print("Пациент успешно удален!")

def update\_patient\_menu():

    patient\_id = int(input("Введите id пациента, данные которого нужно обновить: "))

    patient\_name = input("Введите новое имя пациента (или Enter для пропуска): ")

    doctor\_name = input("Введите новое имя врача (или Enter для пропуска): ")

    diagnosis = input("Введите новый диагноз (или Enter для пропуска): ")

    treatment\_cost = input("Введите новую стоимость лечения (или Enter для пропуска): ")

    if treatment\_cost:

        treatment\_cost = float(treatment\_cost)

    update\_patient(patient\_id, patient\_name, doctor\_name, diagnosis, treatment\_cost)

    print("Данные пациента успешно обновлены!")

if \_\_name\_\_ == "\_\_main\_\_":

    while True:

        main\_menu()

        choice = int(input("Выберите пункт меню: "))

        if choice == 1:

            add\_patient\_menu()

        elif choice == 2:

            search\_patients\_menu()

        elif choice == 3:

            delete\_patient\_menu()

        elif choice == 4:

            update\_patient\_menu()

        elif choice == 5:

            patients = get\_all\_patients()

            for patient in patients:

                print(patient)

        elif choice == 6:

            conn.close()

            break

        else:

            print("Неверный выбор, попробуйте снова.")

изначальное меню выглядит так :

Главное меню:

1. Добавить пациента

2. Поиск пациентов

3. Удалить пациента

4. Редактировать данные пациента

5. Вывести всех пациентов

6. Выход

Выберите пункт меню: 1

Введите имя пациента: михаил

Введите имя врача: иван

Введите диагноз: рука

Введите стоимость лечения: 10000

Пациент успешно добавлен!

Выберите пункт меню: 2

Введите имя пациента (или Enter для пропуска): иван

Введите имя врача (или Enter для пропуска): мария

Введите диагноз (или Enter для пропуска): туберкулез

(1, 'иван', 'мария', 'туберкулез ', 10000.0)

Вывод: В процессе выполнения практического занятие выработал навыки составления программ с базами данных и работы с ними в IDE PyCharm Community. Выполнены разработка кода, отладка ,тестирование, оптимизация, программного кода. Готовые программные коды выложены на GitHub