Студент группы ИС-23 Федосеев Тихон

**Практическая работа №16**

**Тема:** составление программ для работы с базами данных в IDE PyCharm Community.

**Цель**: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, работы с БД в IDE PyCharm Community.

**Тип алгоритма:** Последовательный

**Постановка задачи:**

**Задача 1.**

Приложение СДАЧА В АРЕНДУ ТОРГОВЫХ ПЛОЩАДЕЙ для некоторой организации. БД должна содержать таблицу Торговая точка со следующей структурой записи: этаж, площадь, наличие кондиционера и стоимость аренды в день.

import sqlite3  
  
# Создание таблицы "Торговая точка"  
conn = sqlite3.connect('rental.db')  
c = conn.cursor()  
c.execute('''CREATE TABLE IF NOT EXISTS TorgovajaТochka  
 (etazh integer, ploshchad real, konditsioner integer, stoimost\_arendy real)''')  
conn.commit()  
  
# Функция для ввода данных в БД  
def vvod\_dannykh():  
 # Вставка данных в таблицу  
 floor = int(input("Введите этаж: "))  
 area = float(input("Введите площадь: "))  
 ac = int(input("Введите статус кондиционера (0 или 1): "))  
 rent = float(input("Введите аренду: "))  
  
 c.execute("INSERT INTO TorgovajaТochka VALUES (?, ?, ?, ?)", (floor, area, ac, rent))  
 conn.commit()  
 print("Данные успешно добавлены в БД.")  
  
  
# Функция для поиска данных в БД  
def poisk\_dannykh():  
 # Ввод этажа для поиска  
 etazh = int(input("Введите этаж для поиска: "))  
  
 # Поиск данных в таблице по этажу  
 c.execute("SELECT \* FROM TorgovajaТochka WHERE etazh = ?", (etazh,))  
 results = c.fetchall()  
 if results:  
 # Вывод найденных данных  
 for row in results:  
 print(f"Этаж: {row[0]}, Площадь: {row[1]}, Кондиционер: {row[2]}, Стоимость аренды: {row[3]}")  
 else:  
 print("Торговые точки на указанном этаже не найдены.")  
  
 # Ввод площади для поиска  
 ploshchad = float(input("Введите площадь для поиска: "))  
  
 # Поиск данных в таблице по площади  
 c.execute("SELECT \* FROM TorgovajaТochka WHERE ploshchad = ?", (ploshchad,))  
 results = c.fetchall()  
 if results:  
 for row in results:  
 print(f"Этаж: {row[0]}, Площадь: {row[1]}, Кондиционер: {row[2]}, Стоимость аренды: {row[3]}")  
 else:  
 print("Торговые точки с указанной площадью не найдены.")  
  
 # Ввод наличия кондиционера для поиска  
 konditsioner = int(input("Введите наличие кондиционера для поиска (0 - нет, 1 - да): "))  
  
 # Поиск данных в таблице по наличию кондиционера  
 c.execute("SELECT \* FROM TorgovajaТochka WHERE konditsioner = ?", (konditsioner,))  
 results = c.fetchall()  
 if results:  
 for row in results:  
 print(f"Этаж: {row[0]}, Площадь: {row[1]}, Кондиционер: {row[2]}, Стоимость аренды: {row[3]}")  
 else:  
 print("Торговые точки с указанным наличием кондиционера не найдены.")  
  
  
# Функция для удаления данных из БД  
def udalenie\_dannykh():  
 # Ввод этажа для удаления  
 etazh = int(input("Введите этаж для удаления: "))  
  
 # Удаление данных из таблицы по этажу  
 c.execute("DELETE FROM TorgovajaТochka WHERE etazh = ?", (etazh,))  
 conn.commit()  
 print(f"{c.rowcount} строк удалено.")  
  
 # Ввод площади для удаления  
 ploshchad = float(input("Введите площадь для удаления: "))  
  
 # Удаление данных из таблицы по площади  
 c.execute("DELETE FROM TorgovajaТochka WHERE ploshchad = ?", (ploshchad,))  
 conn.commit()  
 print(f"{c.rowcount} строк удалено.")  
  
 # Ввод наличия кондиционера для удаления  
 konditsioner = int(input("Введите наличие кондиционера для удаления (0 - нет, 1 - да): "))  
  
 # Удаление данных из таблицы по наличию кондиционера  
 c.execute("DELETE FROM TorgovajaТочка WHERE konditsioner = ?", (konditsioner,))  
 conn.commit()  
 print(f"{c.rowcount} строк удалено.")  
  
  
# Функция для редактирования данных в БД  
def redaktirovanie\_dannykh():  
 # Ввод этажа для редактирования  
 etazh = int(input("Введите этаж для редактирования: "))  
  
 # Ввод новых данных  
 ploshchad = float(input("Введите новую площадь: "))  
 konditsioner = int(input("Наличие кондиционера (0 - нет, 1 - да): "))  
 stoimost\_arendy = float(input("Введите новую стоимость аренды в день: "))  
  
 # Обновление данных в таблице по этажу  
 c.execute("UPDATE TorgovajaТочка SET ploshchad = ?, konditsioner = ?, stoimost\_arendy = ? WHERE etazh = ?",  
 (ploshchad, konditsioner, stoimost\_arendy, etazh))  
 conn.commit()  
 print(f"{c.rowcount} строк обновлено.")  
  
 # Обновление данных в таблице по площади  
 c.execute("UPDATE TorgovajaТочка SET etazh = ?, konditsioner = ?, stoimost\_arendy = ? WHERE ploshchad = ?",  
 (etazh, konditsioner, stoimost\_arendy, ploshchad))  
 conn.commit()  
 print(f"{c.rowcount} строк обновлено.")  
  
 # Обновление данных в таблице по наличию кондиционера  
 c.execute("UPDATE TorgovajaТочка SET etazh = ?, ploshchad = ?, stoimost\_arendy = ? WHERE konditsioner = ?",  
 (etazh, ploshchad, stoimost\_arendy, konditsioner))  
 conn.commit()  
 print(f"{c.rowcount} строк обновлено.")  
  
  
# Главное меню программы  
while True:  
 print("\nГлавное меню:")  
 print("1. Ввод данных")  
 print("2. Поиск данных")  
 print("3. Удаление данных")  
 print("4. Редактирование данных")  
 print("5. Выход")  
  
 choice = input("Выберите действие (1-5): ")  
  
 if choice == '1':  
 vvod\_dannykh()  
 elif choice == '2':  
 poisk\_dannykh()  
 elif choice == '3':  
 udalenie\_dannykh()  
 elif choice == '4':  
 redaktirovanie\_dannykh()  
 elif choice == '5':  
 print("Выход из программы.")  
 break  
 else:  
 print("Неверный выбор. Попробуйте еще раз.")  
  
conn.close()

**Протокол работы программы:**

Главное меню:

1. Ввод данных

2. Поиск данных

3. Удаление данных

4. Редактирование данных

5. Выход

Выберите действие (1-5): 5

Выход из программы.

Process finished with exit code 0

**Вывод:** в процессе выполнения практического занятия закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составление программ с использованием БД в IDE PyCharm Community.Были использованы языковые конструкции while, if и другие. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.