

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 TorgovajaTochka
            (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 TorgovajaTochka 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 TorgovajaTochka 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 TorgovajaTochka 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 TorgovajaTochka 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 TorgovajaTochka 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 TorgovajaTochka 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 TorgovajaTochka 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 TorgovajaTochka 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 TorgovajaTochka 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 TorgovajaTочка 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 составления 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.

