# **15 Многопользовательское приложение**

Задание 1. MVVM: Создать ClientModel, OrderModel; Реализовать CRMViewModel с коллекцией клиентов и заказов; Создать команды для добавления нового клиента, редактирования и удаления. Разделение логики и представления: Вынести бизнес-логику работы с клиентами в ClientService; В View оставить только DataBinding и визуальное оформление. Асинхронное программирование: Имплементировать загрузку списка клиентов через Task.Run(); Добавить индикатор загрузки.

Листинг программы:

using System.ComponentModel;

namespace X.Models

{

public class OrderModel : INotifyPropertyChanged

{

private int orderId;

private string clientName;

public int OrderId

{

get => orderId;

set { orderId = value; OnPropertyChanged(nameof(OrderId)); }

}

public string ClientName

{

get => clientName;

set { clientName = value; OnPropertyChanged(nameof(ClientName)); }

}

public event PropertyChangedEventHandler PropertyChanged;

protected void OnPropertyChanged(string pn)

{

PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(pn));

}

}

}

using System.Collections.Generic;

using System.Threading.Tasks;

namespace X.Services

{ public class ClientService

{ public async Task<List<ClientModel>> GetClientsAsync()

{ await Task.Delay(2000);

return new List<ClientModel>

{ new ClientModel { Name = "Ангелина Авдеева", Contact = "+37529. . . . . . .", Email = "angelina@mail.com", Type = "VIP" },

new ClientModel { Name = "Имя Фамилия", Contact = "+3752912345679", Email = "name@mail.com", Type = "Обычный" } }; }

}}

Анализ результатов:

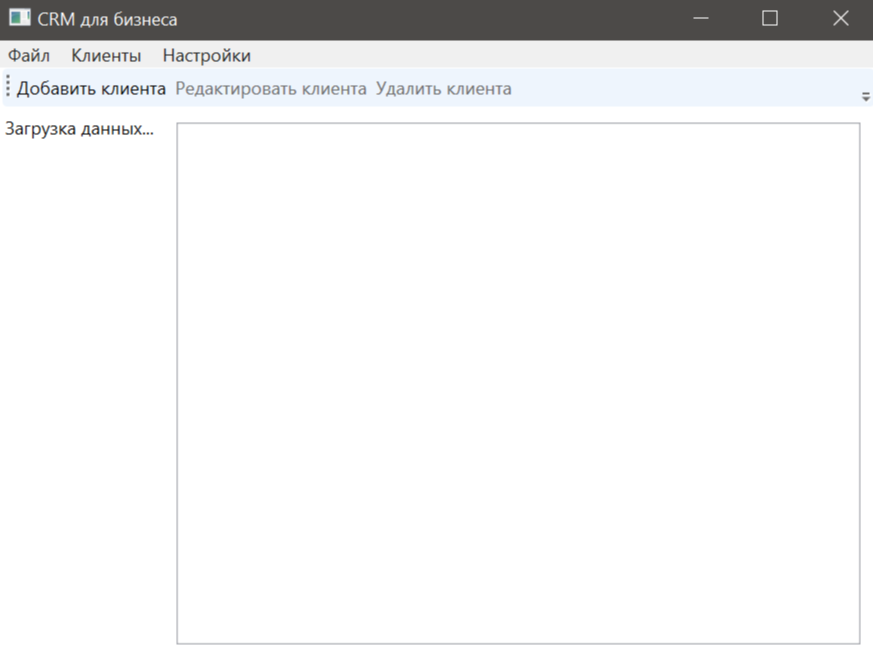


Рисунок 15.1 – Результат работы программы