**13** **Работа с графикой. Анимация**

Задание 1. Автоматизация учета сотрудников

Листинг задачи:

public class ChatViewModel : INotifyPropertyChanged

{

private readonly ChatClient \_client;

private string \_message;

private string \_chatHistory;

public ChatViewModel()

{

SendCommand = new RelayCommand(\_ => SendMessage());

var server = new ChatServer("ChatPipe");

server.MessageReceived += async (s, e) => await server.BroadcastMessageAsync(e.Message);

\_ = server.StartAsync();

\_client = new ChatClient("ChatPipe");

\_client.MessageReceived += (s, msg) =>

ChatHistory += $"{DateTime.Now:T} - {msg}{Environment.NewLine}";

\_ = \_client.ConnectAsync();

}

public string Message

{

get => \_message;

set

{

\_message = value;

OnPropertyChanged(nameof(Message));

}

}

public string ChatHistory

{

get => \_chatHistory;

set

{

\_chatHistory = value;

OnPropertyChanged(nameof(ChatHistory));

}

}

public ICommand SendCommand { get; }

private async void SendMessage()

{

if (!string.IsNullOrWhiteSpace(Message))

{

await \_client.SendAsync(Message);

Message = string.Empty;

}

}

public event PropertyChangedEventHandler PropertyChanged;

protected virtual void OnPropertyChanged(string propertyName)

{

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

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Сообщение | Вывод сообщения |

Анализ результатов:

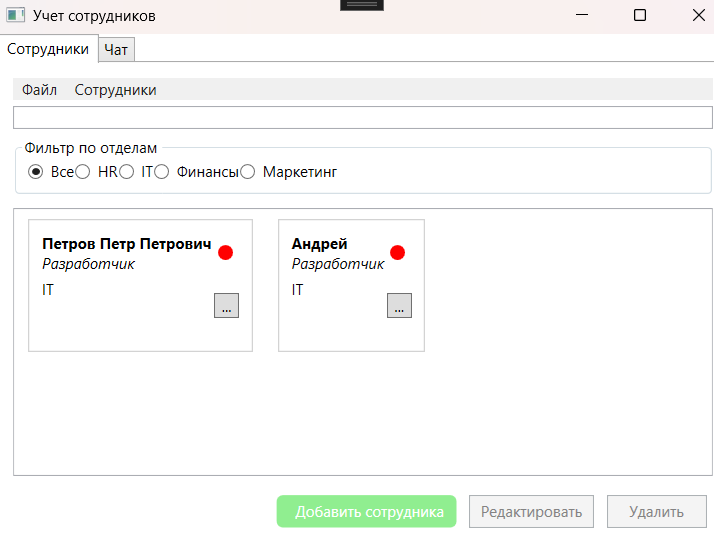


Рисунок 1.1 – Результат работы программы