# Практическое занятие

**Интерфейсы в С#. Множественное наследование**

Шаг 1. Открыть VS, создать проект WPF.NetFramework, дать наименование Interface\_Фамилия.

Шаг 2. Загрузить проект в систему контроля версий.

Шаг 3. Создать следующие папки в проекте:

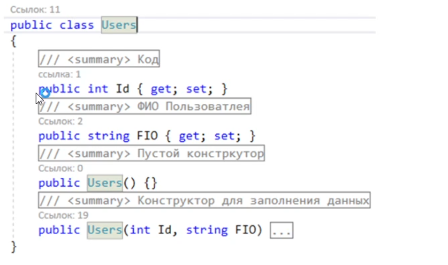
- Models – содержит модели хранимых данных;

- Interfaces – содержит интерфейсы;

- Classes - содержит классы;

- Images – содержит изображения.

Шаг 4. В папке Models создать класс Users. Класс будет содержать информацию о пользователях, находящихся в системе, прописать следующий код:



Шаг 5. В папке Models создать класс Messages. Класс будет содержать информацию о сообщениях, отправленных пользователям, прописать следующий код

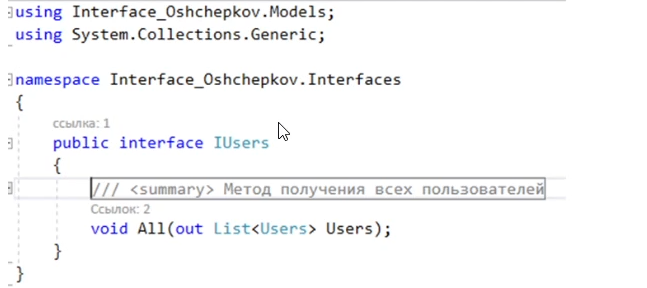


!!!! или

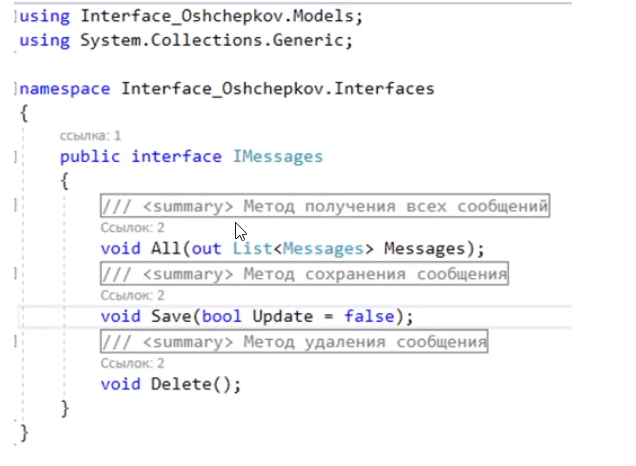


Шаг 6. Загрузить изменения в систему контроля версий, прописав комментарий Реализованные модели\_Фамилия.

Шаг 7. В папке Interfaces создать класс IUsers. Интерфейс будет говорить, какие методы необходимо реализовать классу, который будет реализовать интерфейс, прописать следующий код.

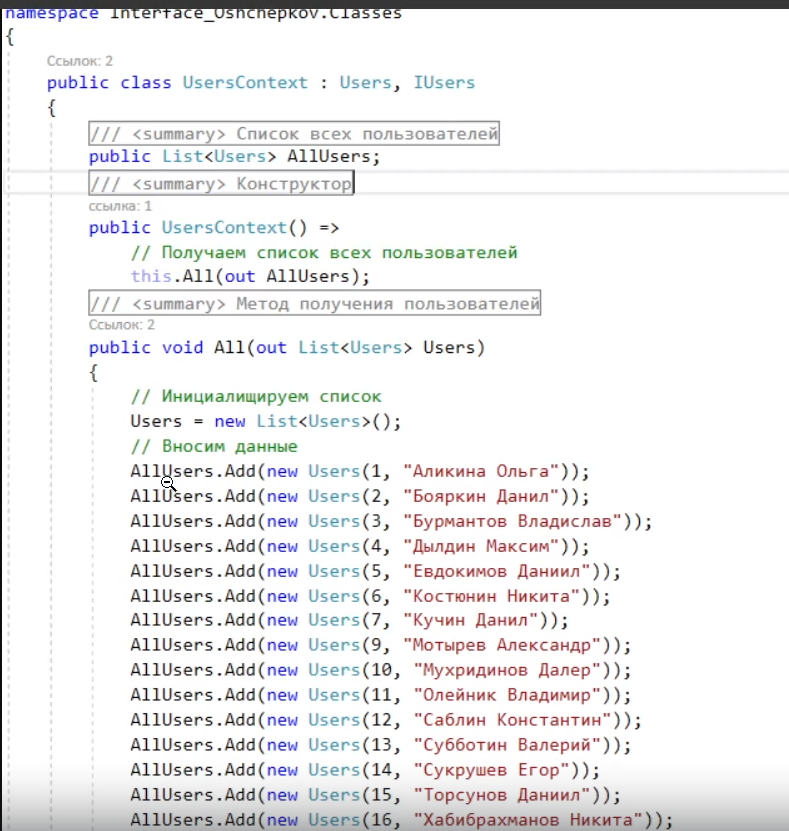


Шаг 8. В папке Interfaces создать класс IMessages. Интерфейс будет говорить, какие методы необходимо реализовать классу, который будет реализовывать интерфейс, прописать следующий код:

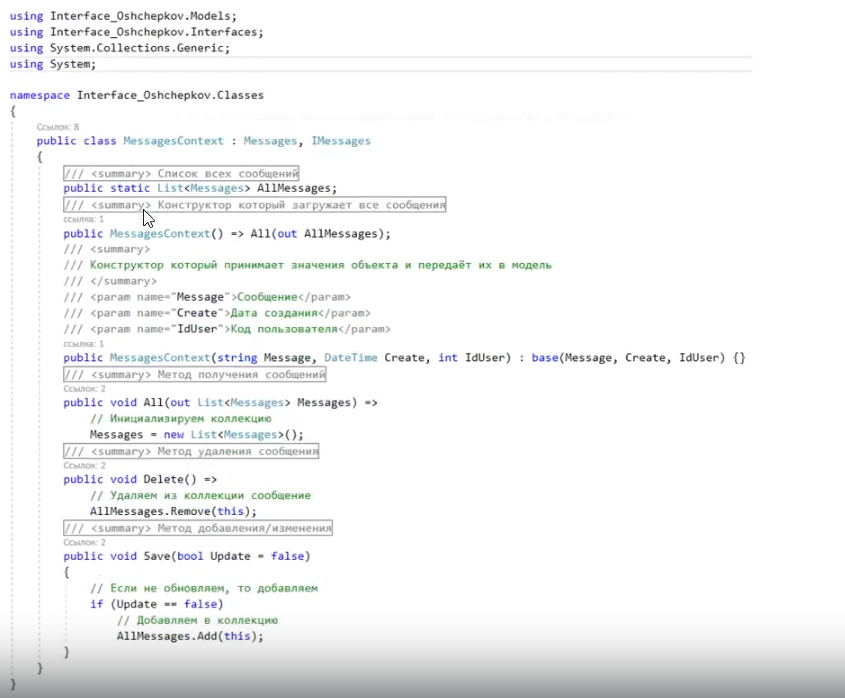


Шаг 9. Загрузить изменения в систему контроля версий, прописав комментарий Реализованные интерфейсы\_Фамилия.

Шаг 10. В папке Classes создать класс UsersContext. Класс будет реализовывать интерфейс IUsers и выполнять роль базы данных, прописать следующий код:



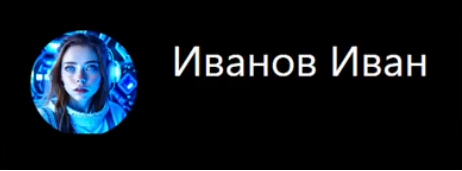
Шаг 11. В папке Classes создать класс MessagesContext. Класс бедет реализовывать интерфейс IMessages и выполнять роль базы данных, прописать следующий код:



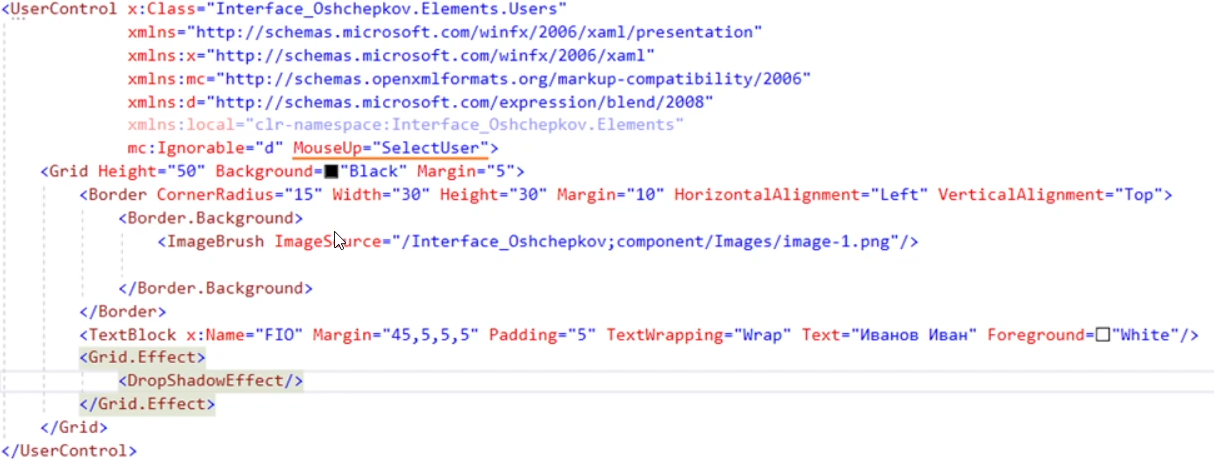
Шаг 12. Загрузить изменения в систему контроля версий, прописав комментарий Реализованные классы реализующие интерфейсы\_Фамилия.

Шаг 13. В папку Images загрузите фото пользователей (сгенерированные нейросетями).

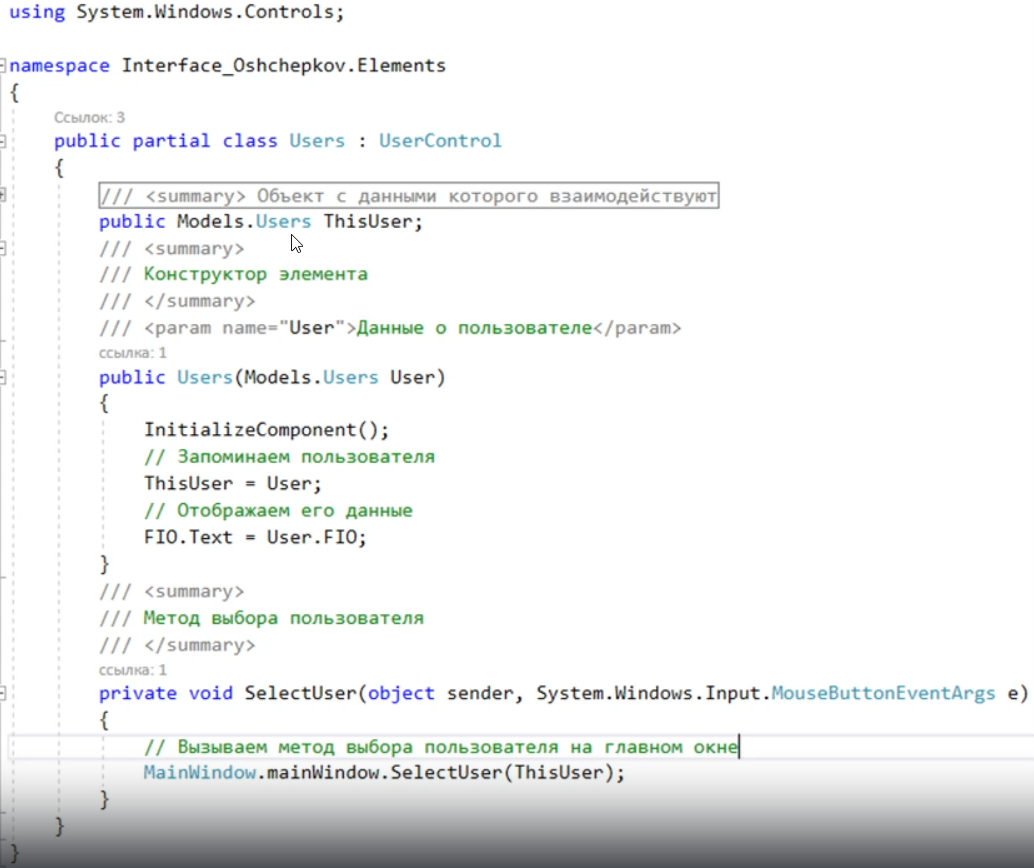
Шаг 14. В папке Elements создать пользовательский элемент управления Users, данный элемент будет отображать пользователей системы, реализовать следующий интерфейс:



XAML-код пользовательского элемента:



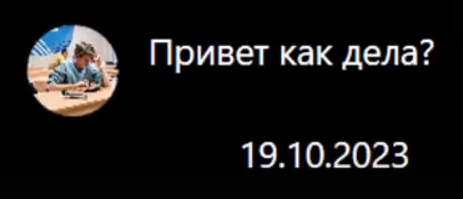
Шаг 15. Для пользовательского элемента прописать следующий код:



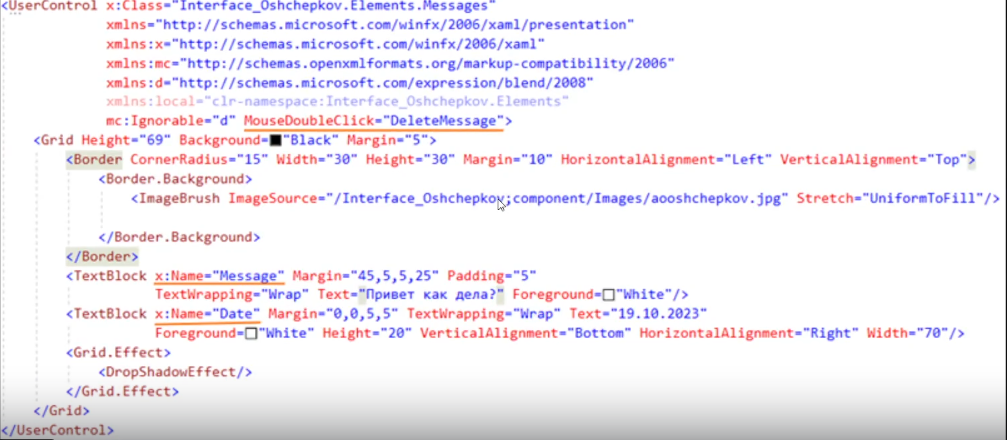
MainWindow.xaml.cs



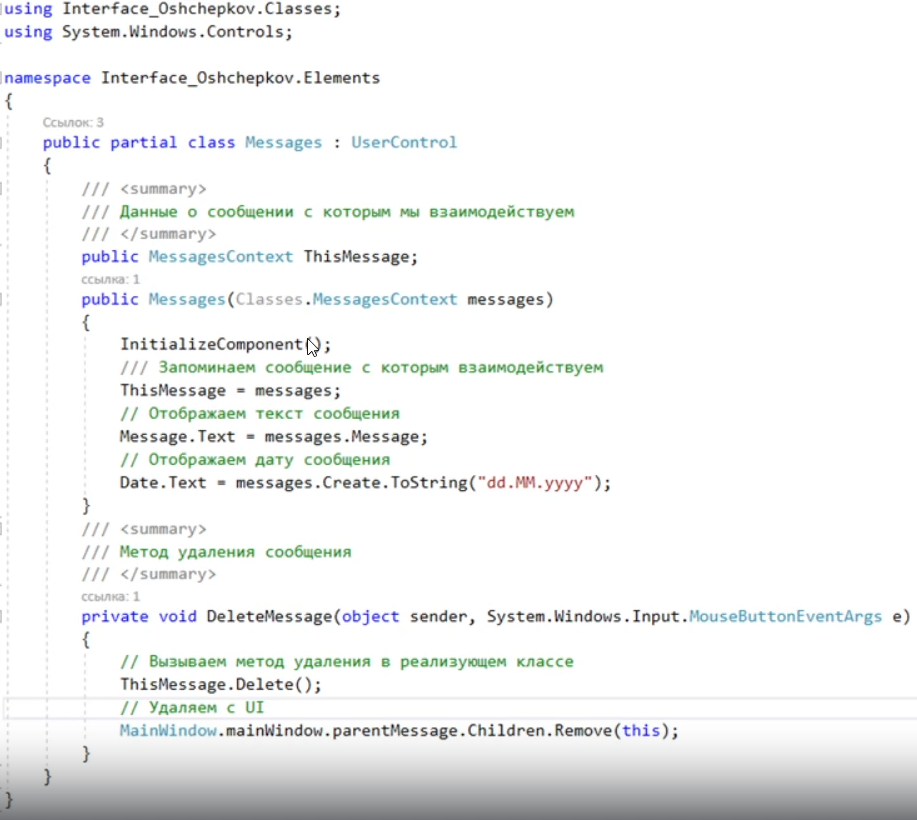
Шаг 16. В папке Elements создать пользовательский элемент управления Messages, данный элемент будет отображать сообщения пользователю системы, реализовать следующий интерфейс:



Для пользовательского элемента Messages прописать следующий код:

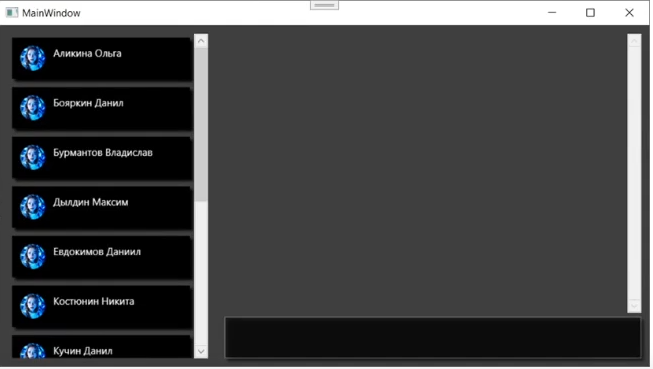


Шаг 17. Для созданного пользовательского элемента управления Messages прописать следующий код:

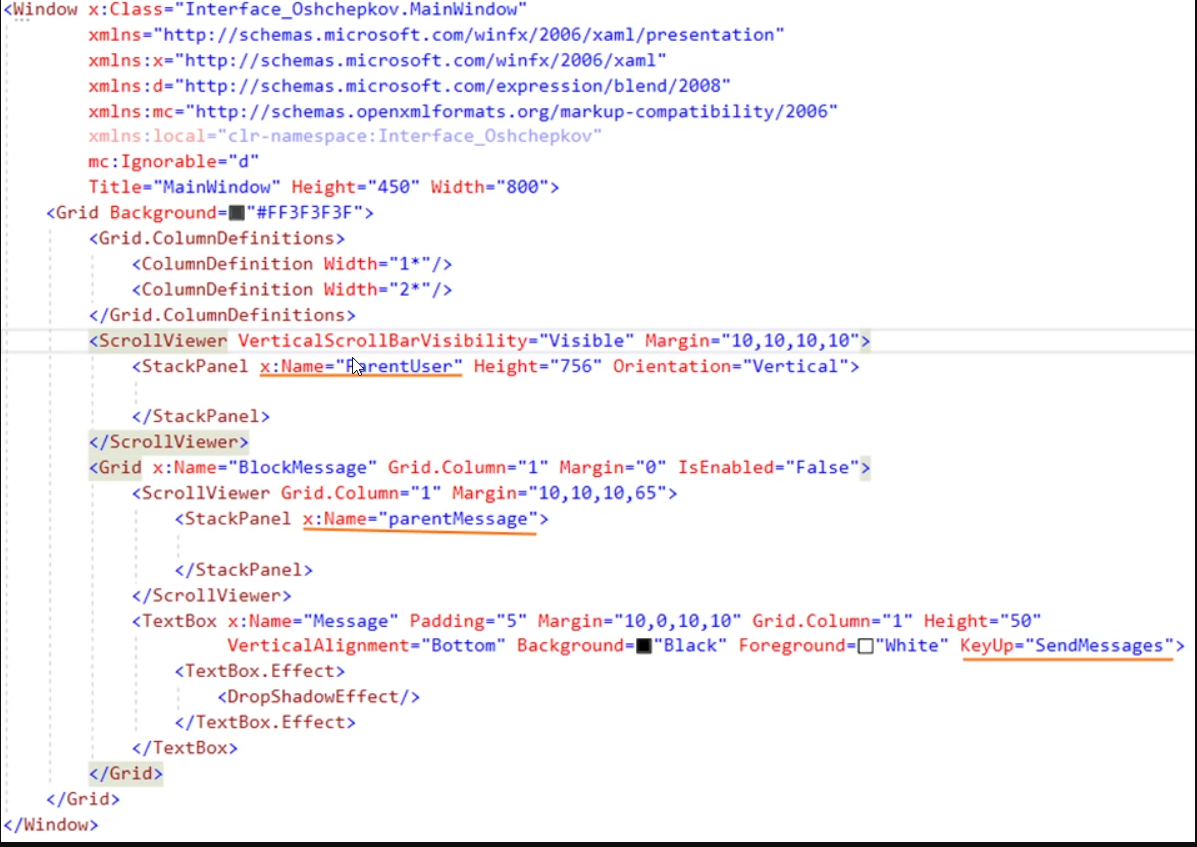


Шаг 18. Загрузить изменения в систему контроля версий, прописав комментарий Реализованные пользовательские элементы управления\_Фамилия.

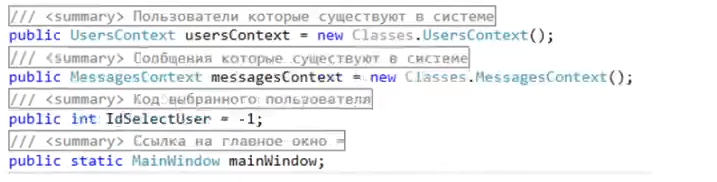
Шаг 19. Для главного окна MainWindow реализовать следующий интерфейс:



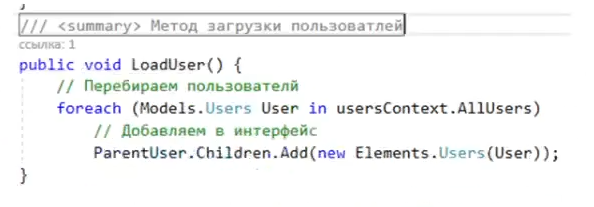
XAML-код интерфейса:



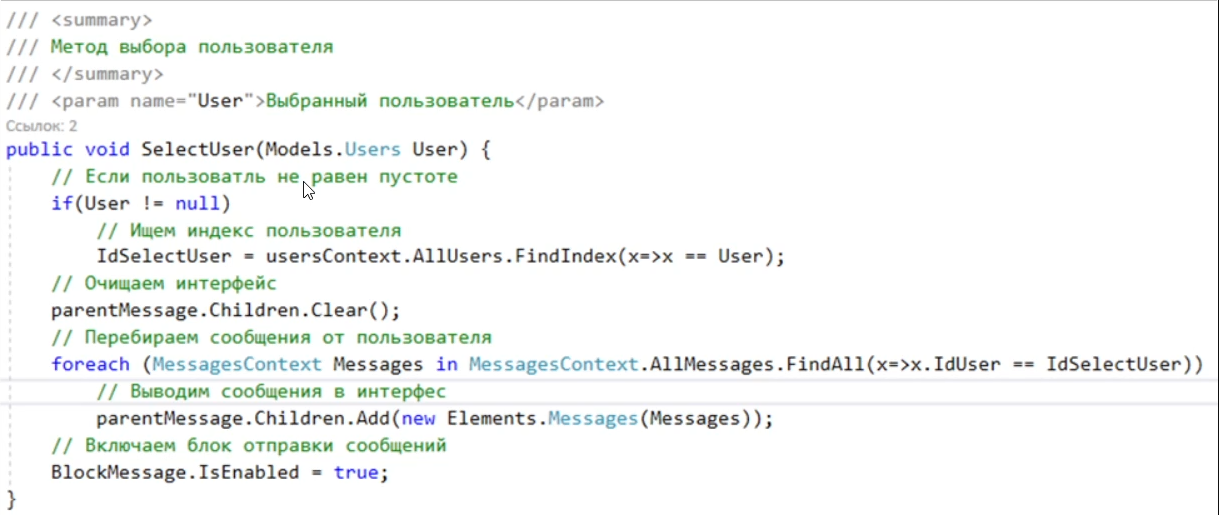
Шаг 20. Для главного окна MainWindow прописать переменные, которые будут использоваться:



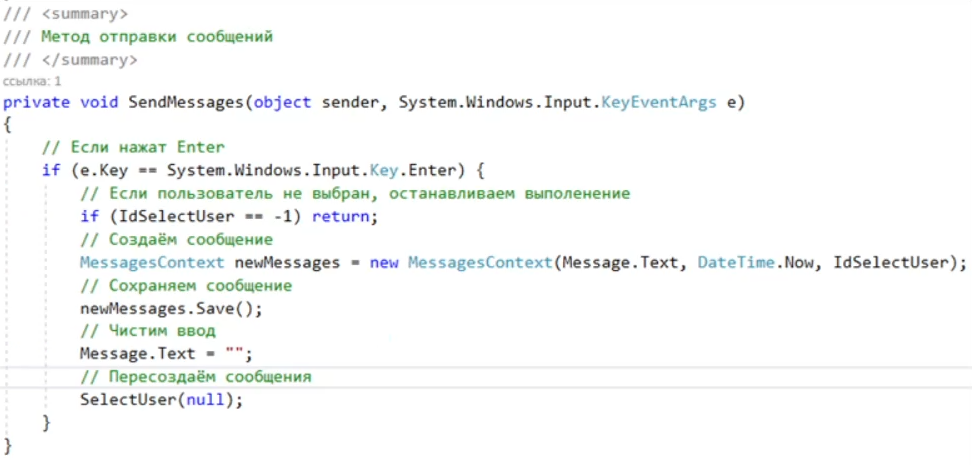
Шаг 21. Для главного окна MainWindow прописать метод загрузки пользователей в интерфейс программы:



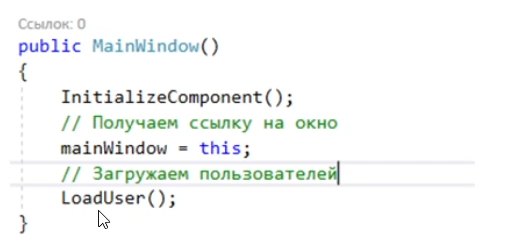
Шаг 22. Для главного окна MainWindow прописать метод выбора пользователей.



Шаг 23. Для главного окна MainWindow реализовать метод отправки сообщений пользователям:



Шаг 24. Для главного окна MainWindow прописать конструктор



Шаг 25. Загрузить изменения в систему контроля версий, прописав комментарий «Реализованная практическая работа на оценку удовлетворительно. Фамилия».

Шаг 26. Ответить на контрольные вопросы письменно!

Контрольные вопросы:

1. Что такое интерфейс?
2. Как отличить интерфейс от класса?
3. Что представляет собой множественное наследование?
4. В каком языке реализовано множественное наследование?
5. Как реализовано множественное наследование в C#?

Дополнительно

Для получения оценки «Хорошо» необходимо реализовать возможность изменения отправленного сообщения.

Для получения оценки «Отлично» необходимо реализовать функционал, предложенный на оценку «Хорошо», реализовать возможность отправки изображения в чат.