|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **МИРЭА – Российский технологический университет**  **РТУ МИРЭА**  **Колледж приборостроения и кибербезопасности** |

**Практическая работа №1**

по учебной дисциплине МДК.01.02 Поддержка и тестирование программных модулей

**Изучение и применение методов рефакторинга**

специальность 09.02.07 Информационные системы и программирование

выполнил

студент группы ПКС-31

Лопатин Л.В.

проверил преподаватель

Мымрина М.Л.

**Ход работы:**

1. В ходе изучения возможностей инструментов рефакторинга ИСР Visual Studio были выбраны следующие методы рефакторинга: удаление неиспользованных директив using, выделение методов, инкапсуляция полей

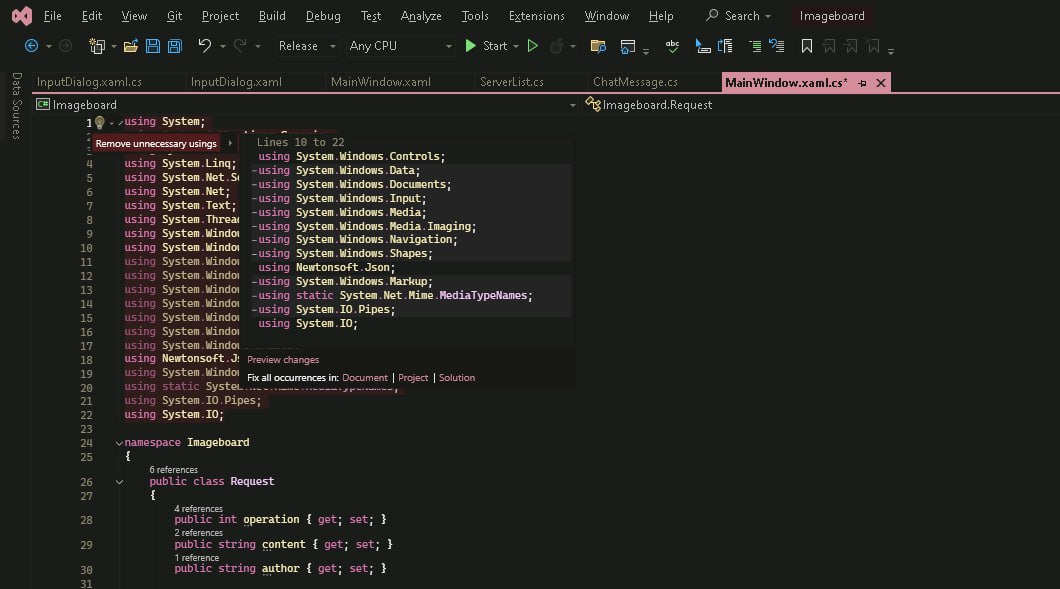
Удаление неиспользованных директив using — избавление от всех неиспользованных в конце разработке директив, для улучшения читаемости кода и большего понимания какие директивы используют разные классы.

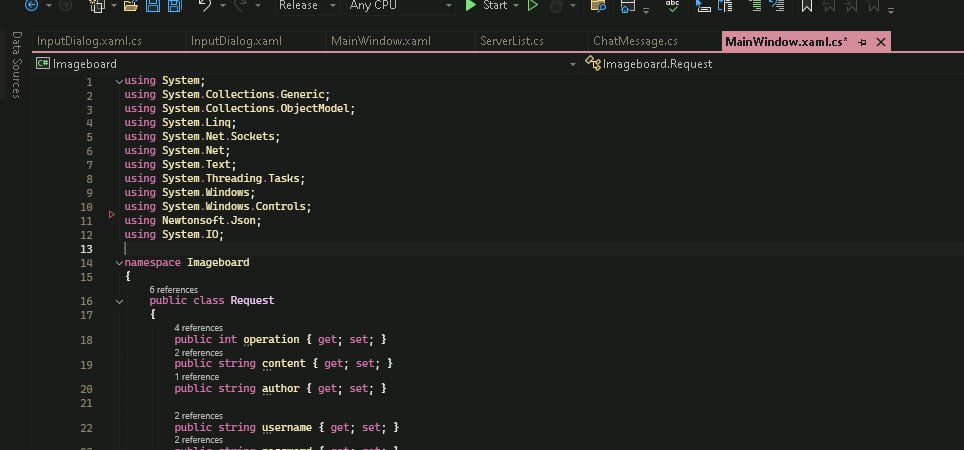
Выделение метода — выделение самостоятельных или повторяющихся несколько раз частей кода.

Инкапсуляция полей — включение поле в свойство и обновление всех случаев использования этого поля, чтобы применить созданное свойство.

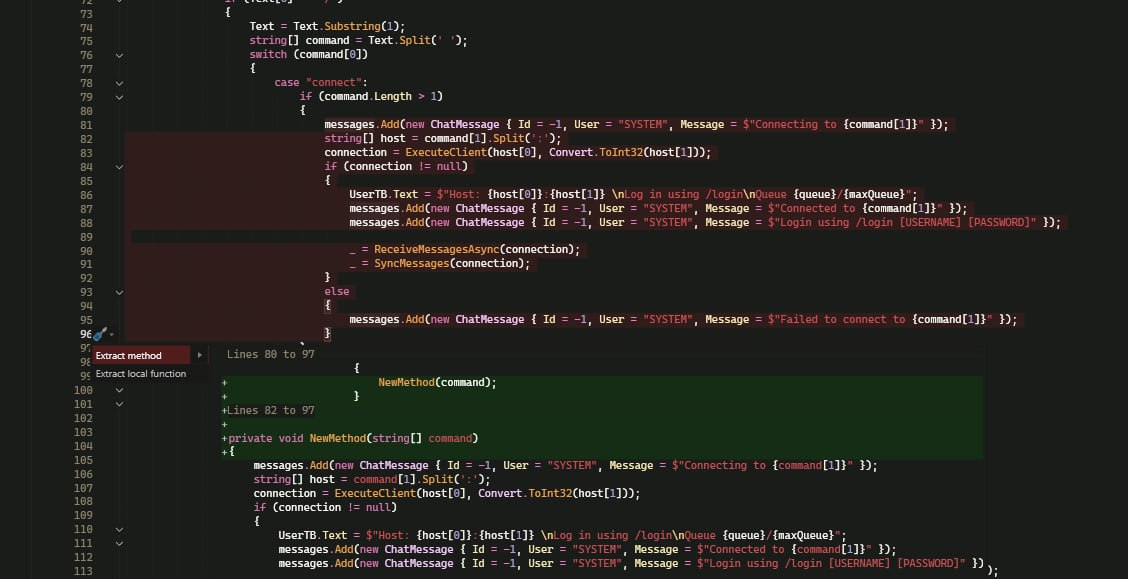
2.

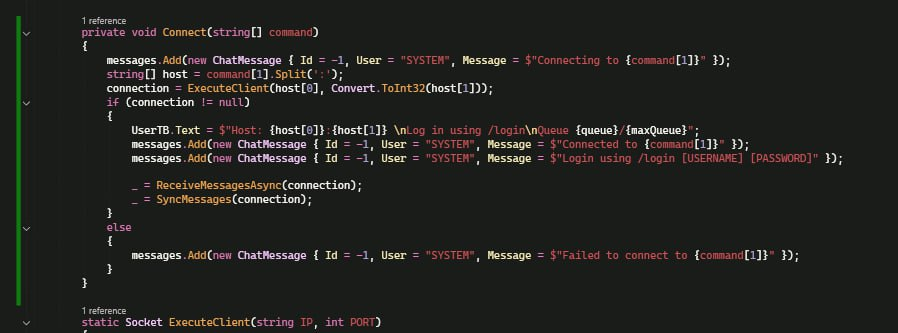
В файле класса MainWindows обнаружил неиспользуемые директивы using. С помощью средств ИСР удалил все ненужные директивы (см. рисунки 1-2)

Рис. 1 Применение инструмента рефакторинга

Рис. 2 Результат применения

В методе обработчика команд были обнаружены большие фрагменты кода. Было принято решение выделить их в отдельные методы для лучшей читаемости и редактируемости (см. рисунки 3-4).

Рис. 3 Применения инструмента рефакторинга

Рис. 4 Выделенный метод

При анализе кода было принято решение инкапсулировать несколько полей (см. рисунок 5-6).

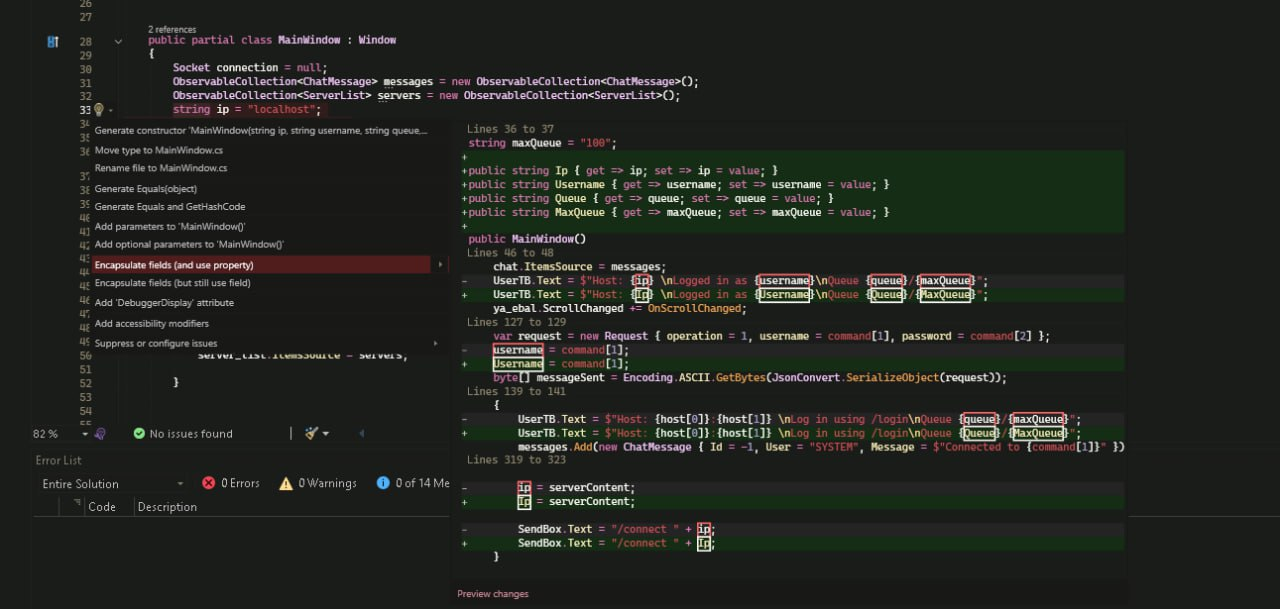
Рис. 5 Инкапсуляция полей

Рис. 6 Применение полей

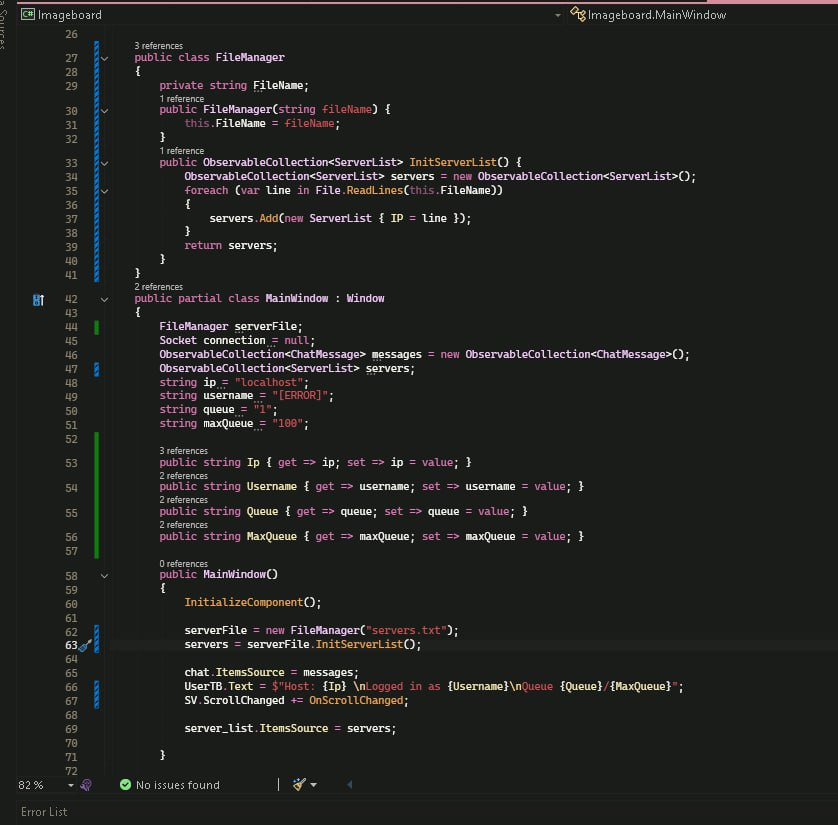
При анализе кода было принято пребегнуть к ручном методам рефакторинга:

Выделение класса - создание нового класса для логически связанного функционала, чтобы улучшить читаемость кода и упростить поддержку.

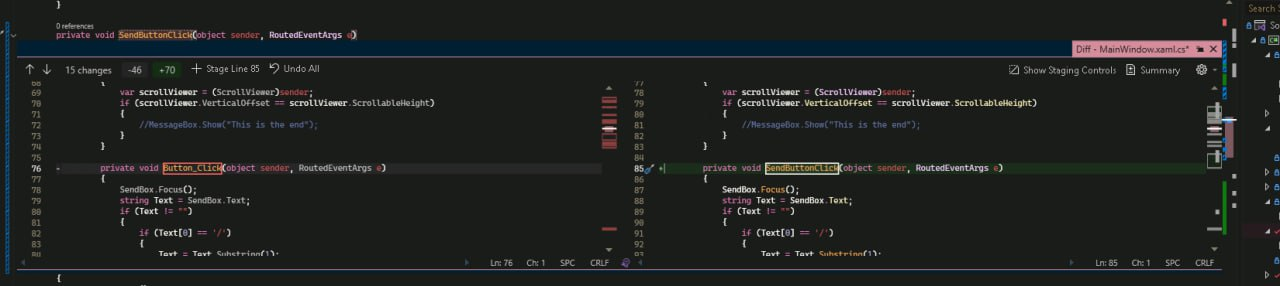
Переименование метода - выбор более осмысленного и говорящего названия метода, отражающего его назначение.

Выделение констант - замена «магических чисел» и строковых литералов на именованные константы для повышения понятности кода.

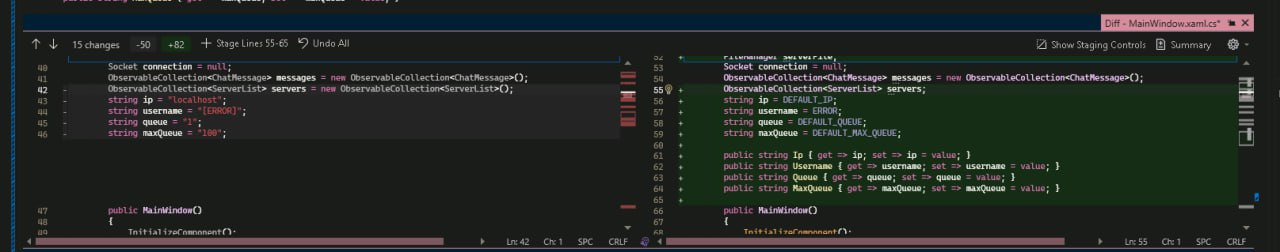
Был выделен класс FileManager, предназначенный для чтения и записи необходимых адрессов серверов(см. Рисунок 7).

Рис. 7 Выделение класса

В ходе анализа был выявлен метод с неявным названием. В результате переименовал его(см. Рисунок 8)

Рис. 8 Переименование функции

В ходе анализа было принято решение заменить «магические» значения на константы(см. рисунок 9)

Рис. 9 Выделение констант

Вывод: в ходе работы изучил методы автоматизированного и ручного рефакторинга и научился их применять.