МИНИСТРЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования “Донецкий национальный технический университет”

Факультет ИСП  
Кафедра ПИ им Л.П. Фельдмана

Лабораторная работа №2  
по курсу: «Качество программного обеспечения и тестирование»  
на тему: «Работа с ветками»  
Вариант №1

Выполнил:  
ст. гр. ПИ-20а  
Айдин С. А.

Проверил:

Филипишин Д. А.

ДОНЕЦК – 2023

**Ссылка на проект:** (https://github.com/IDHSRG/aidin\_pppi\_chat/tree/main)

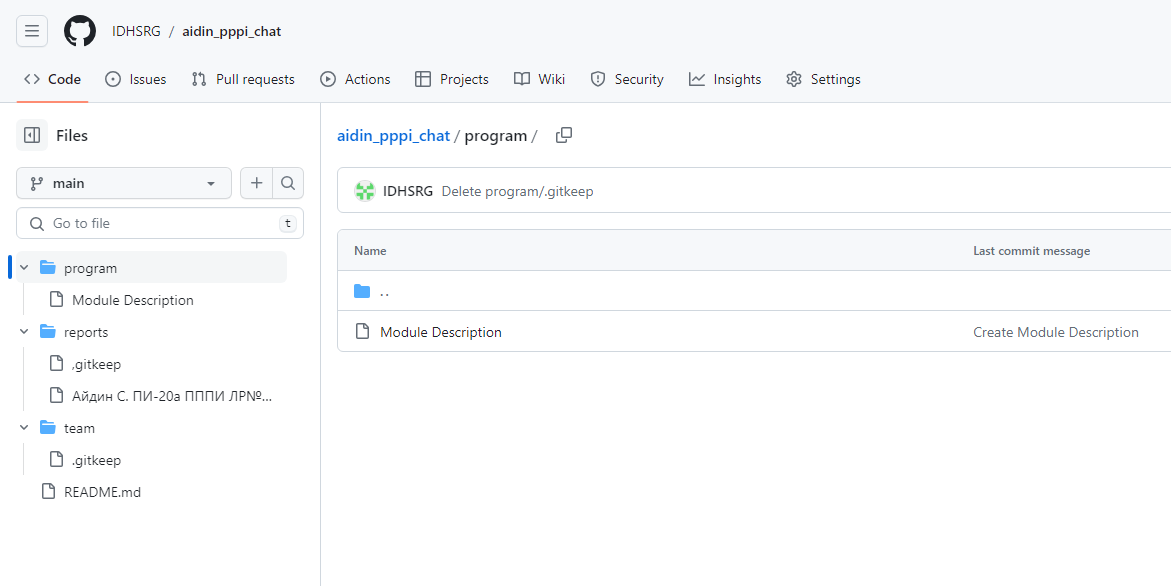


Рисунок 1 – Ветка main

**Было решено создать 3 ветки:**

**- interface** – для моделирования классов для интерфейса чата, обработки команд пользователя

**- client** – для моделирования классов модуля клиента, который будет принимать и отправлять сигналы на сервер, сохранять кэш сообщений и в целом взаимодействовать с сервером, управляется через интерфейс.

**- server** – для модуля сервера, который будет работать, принимать сигналы и сообщения, сохранять данные и передавать их соответствующим пользователям, проверять их активность и т.д.

**Далее будет представлена структура веток (какие файлы в них были созданы) и их содержимое (достаточно большое содержимое будет показано лишь частично – чтобы просмотреть полную версию можно посетить проект по ссылке выше)**

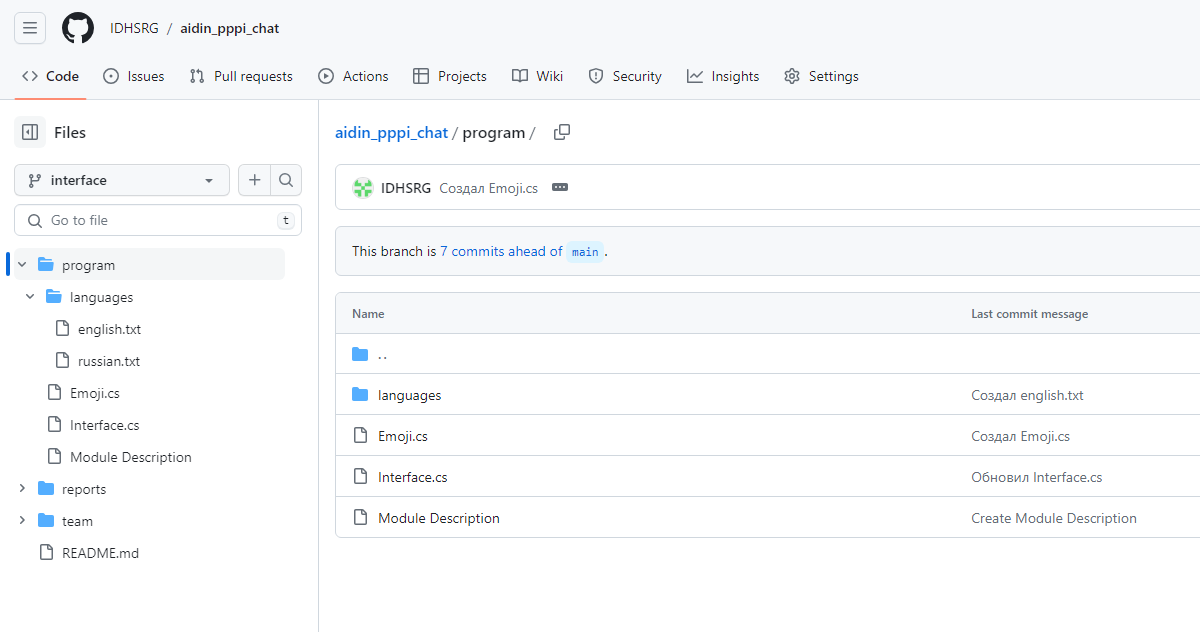


Рисунок 2 – Ветка interface (содержимое)

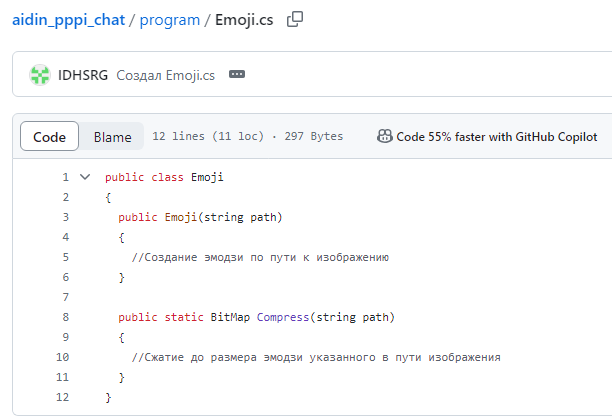


Рисунок 3 –Ветка interface (файл Emojie.cs)

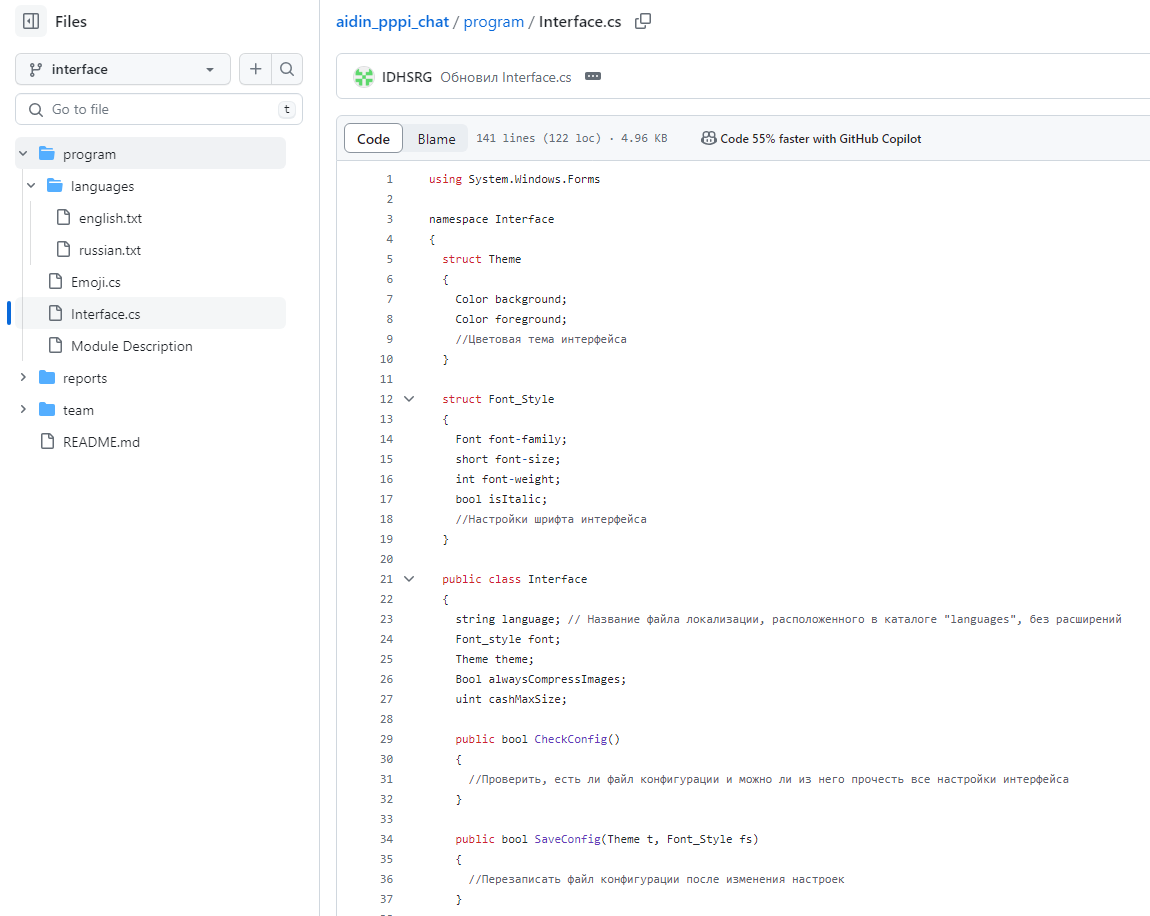


Рисунок 4 –Ветка interface (файл Interface.cs)

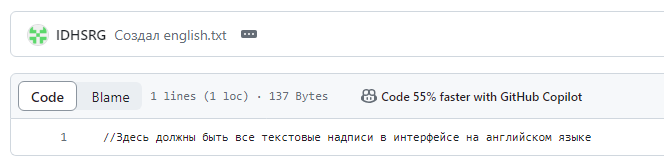


Рисунок 5 –Ветка interface (файл russian.txt)

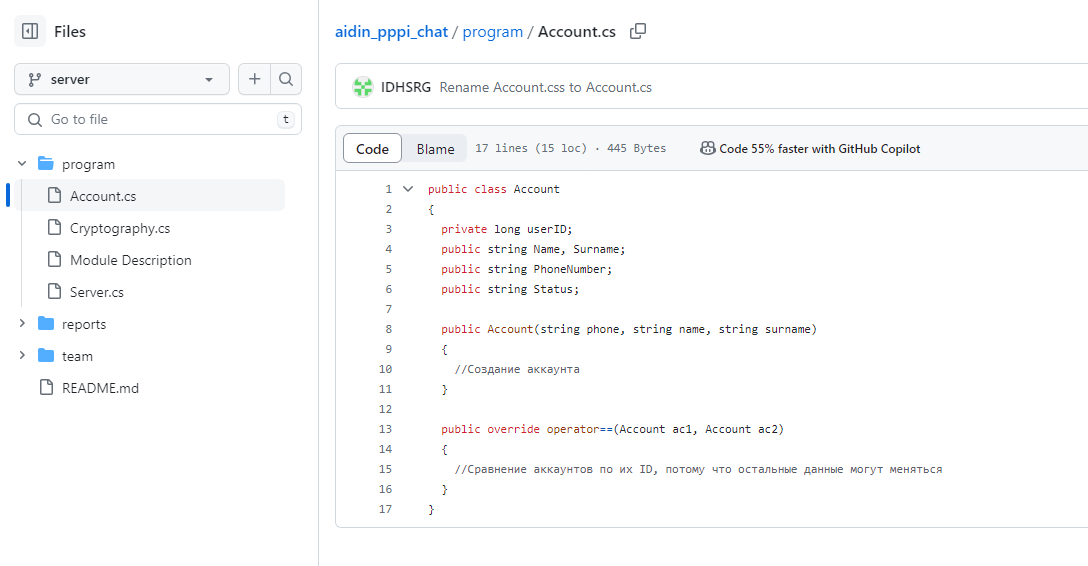


Рисунок 6 – Ветка server (содержимое, файл Accounts.cs)

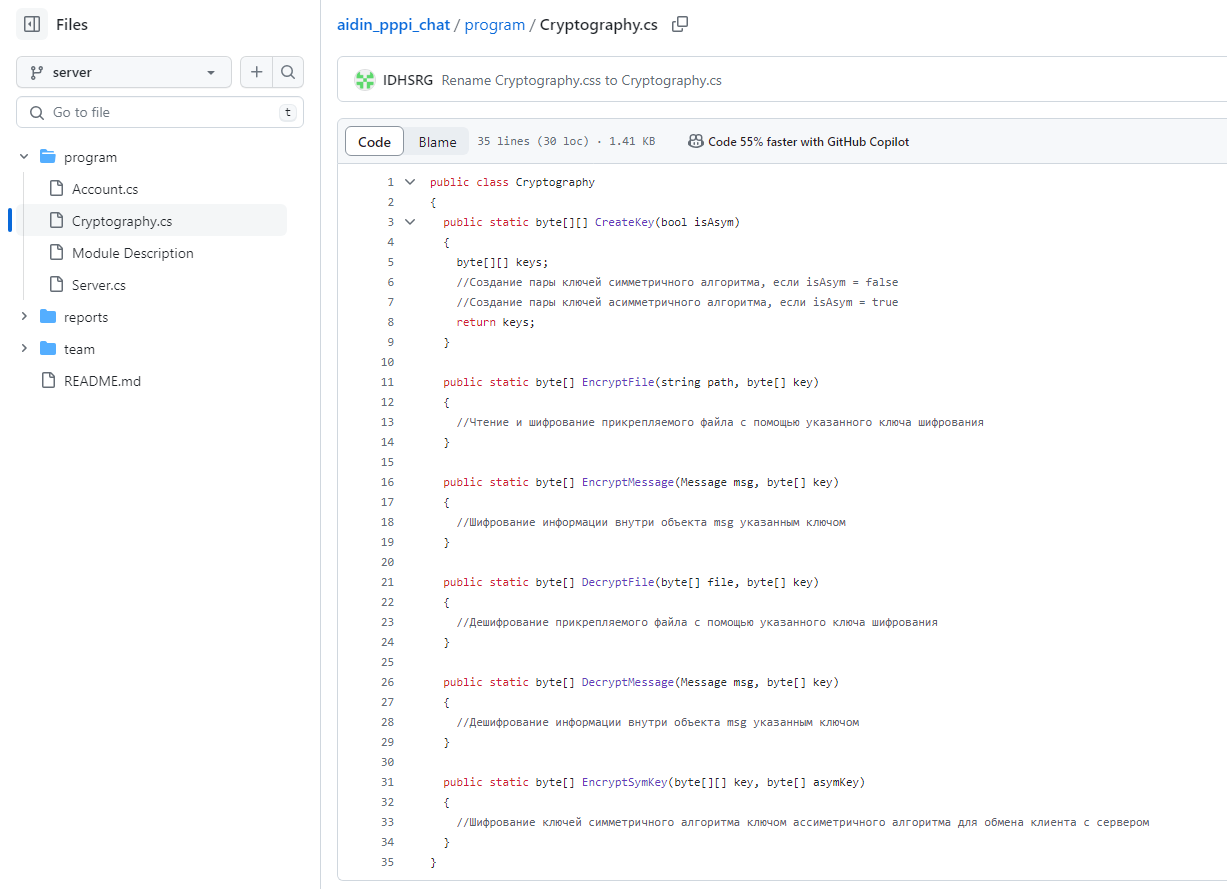


Рисунок 7 –Ветка server (файл Cryptography.cs)

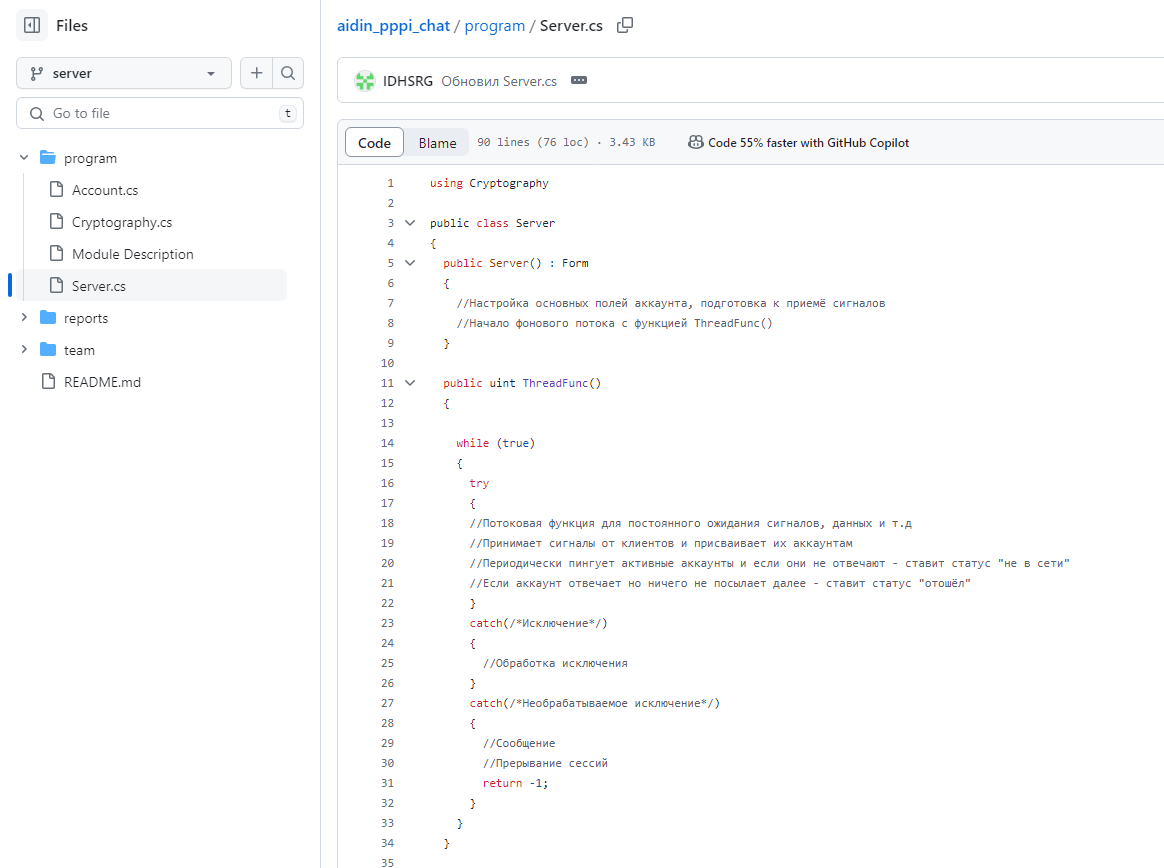


Рисунок 8 –Ветка server (файл Server.cs)

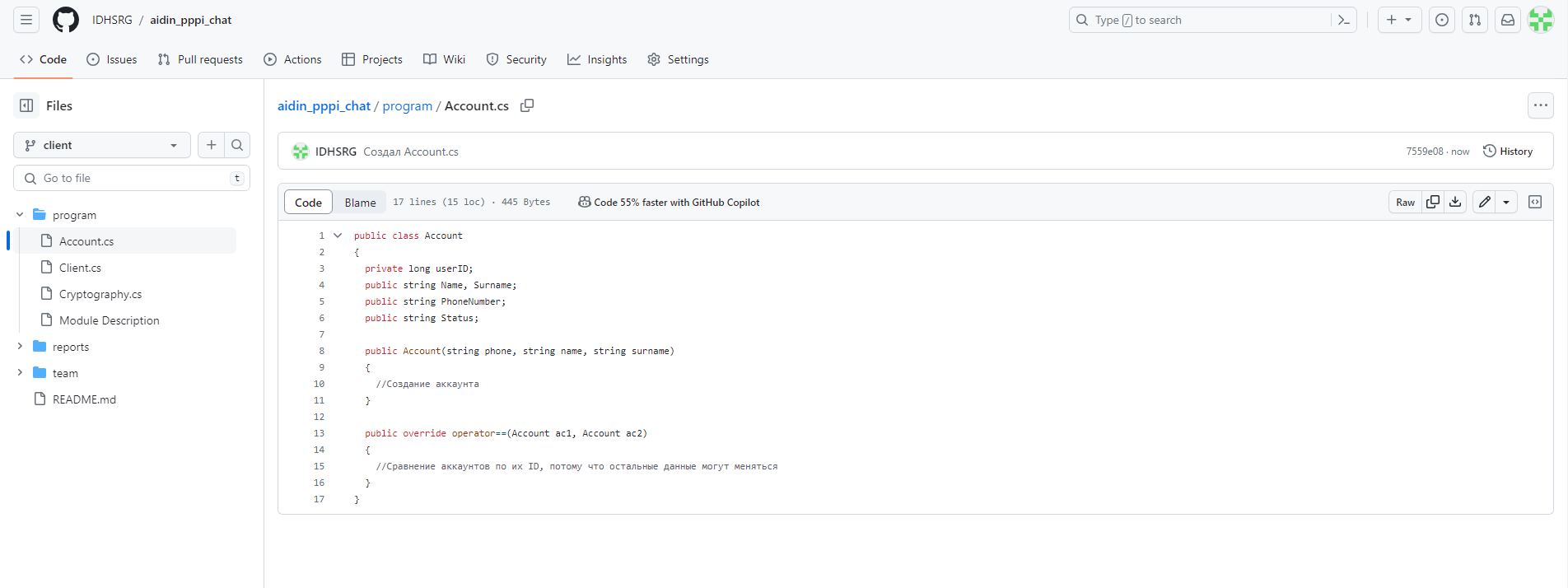


Рисунок 9 –Ветка client (Содержимое, файл Accounts.cs)

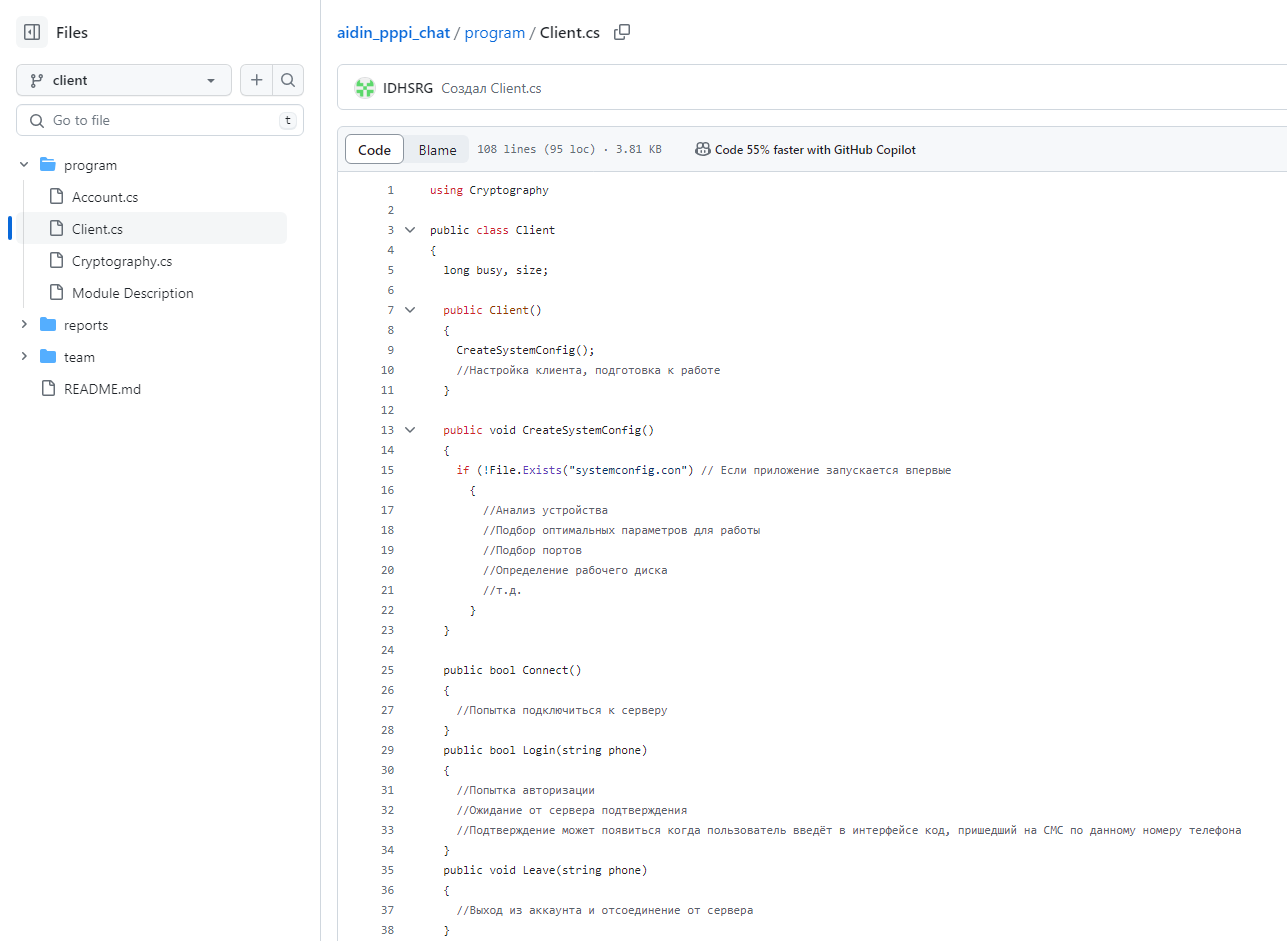


Рисунок 10 –Ветка client (файл Client.cs)

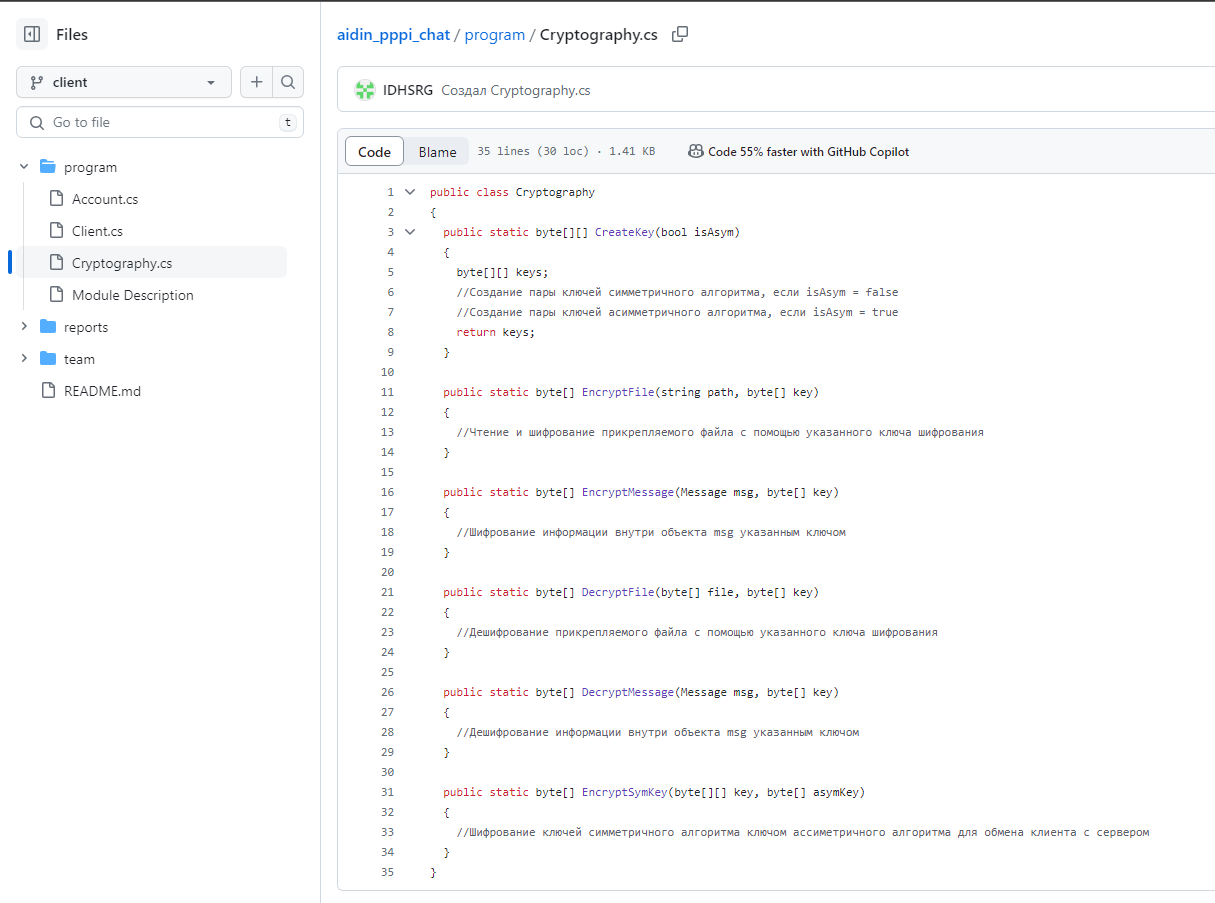
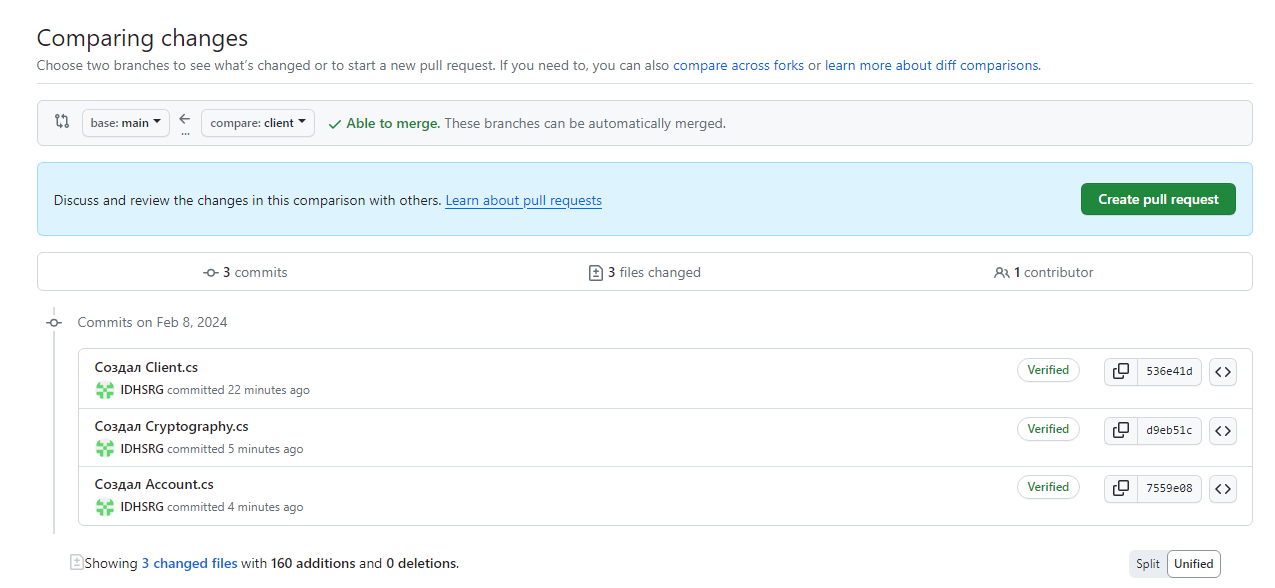
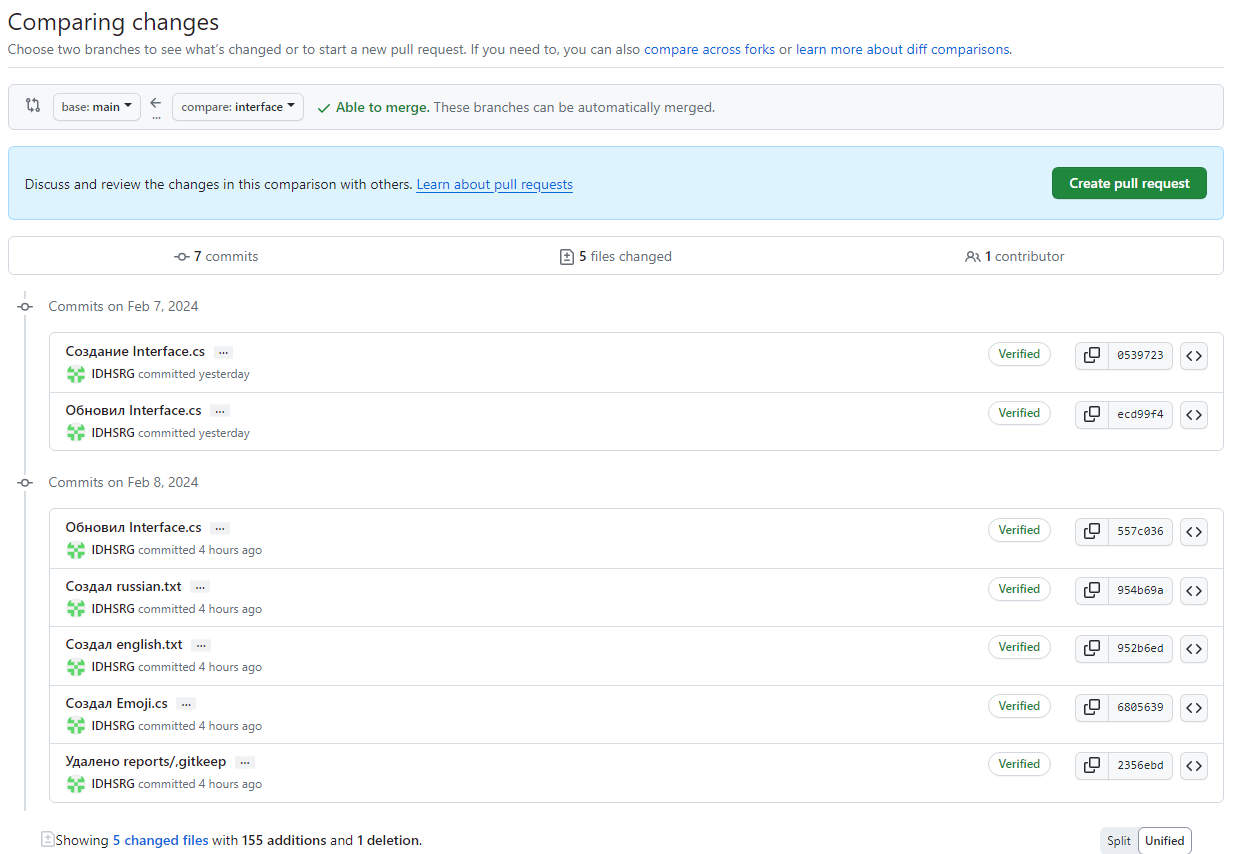


Рисунок 11 – Ветка client (Файл Cryptography.cs)





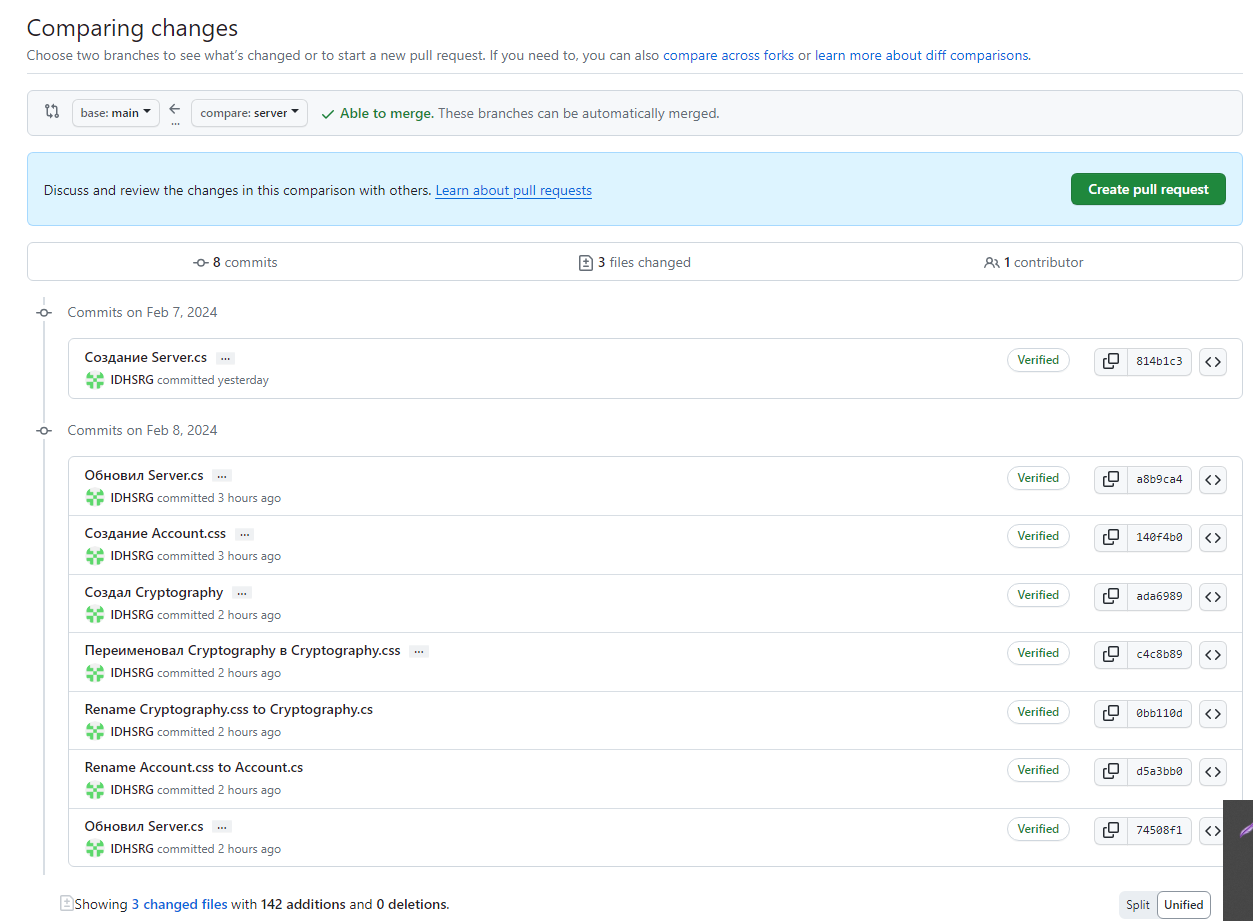


Рисунок 12 – Истории коммитов для каждой ветки

Далее произведём мерж ветки main с веткой server

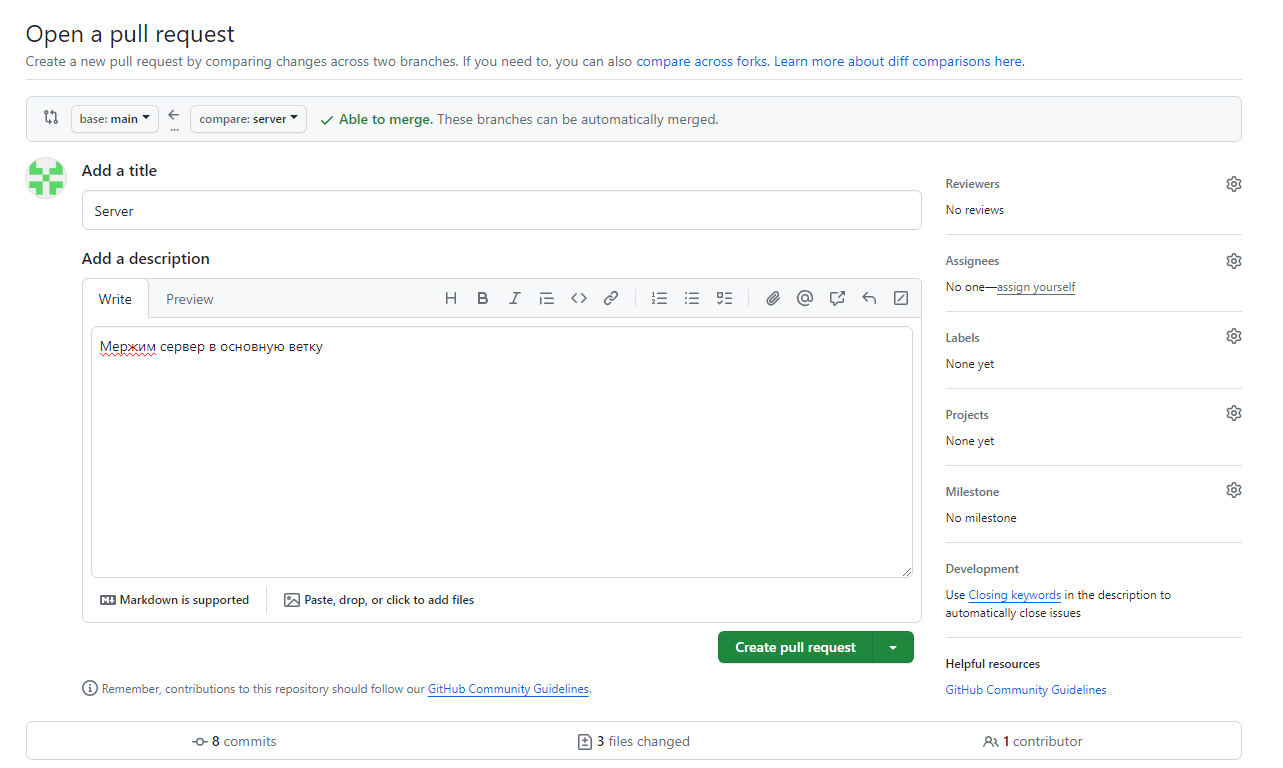


Рисунок 13 –Создание пулл-реквеста

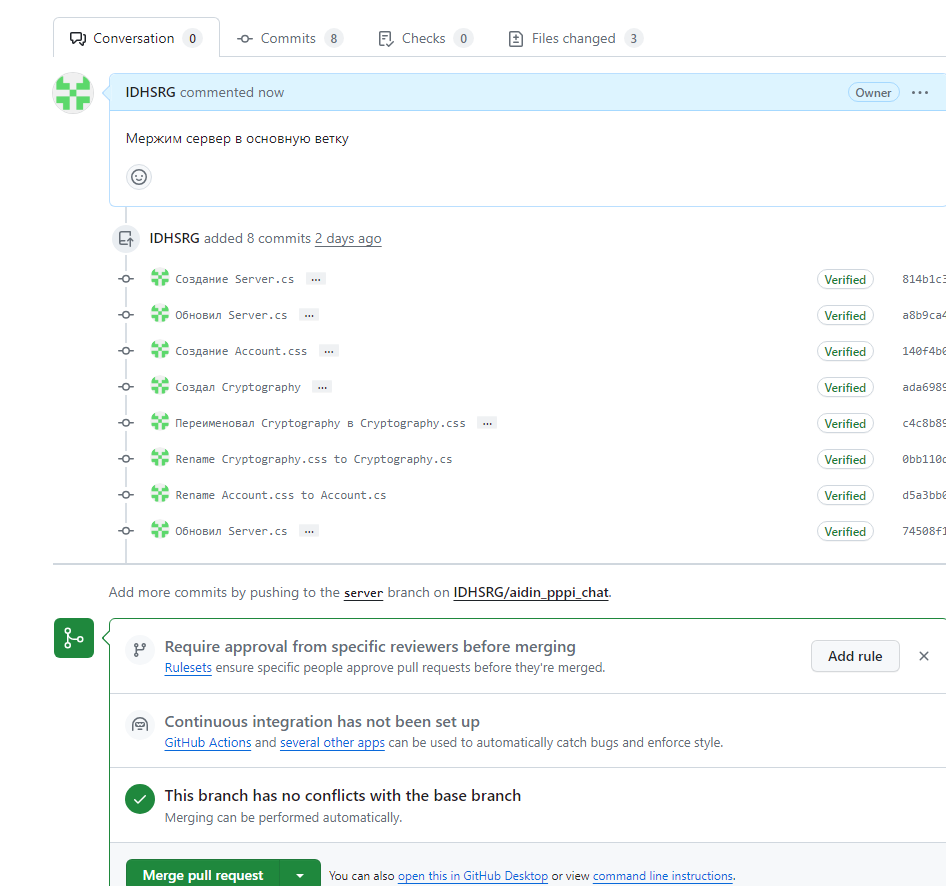


Рисунок 14 – Информация о том что авто-мерж возможен

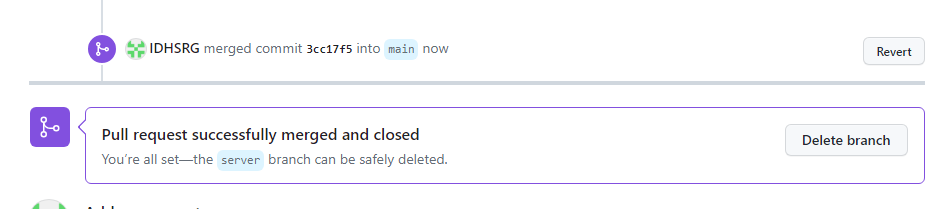


Рисунок 15 – Удачное объединение веток

Перед тем как продолжить объединение веток, допустим, привнесём изменения в файл Accounts.cs в ветке client (этот же файл был в ветке server, при объединении основной ветки с веткой client возникнет конфликт, который необходимо разрешить вручную)

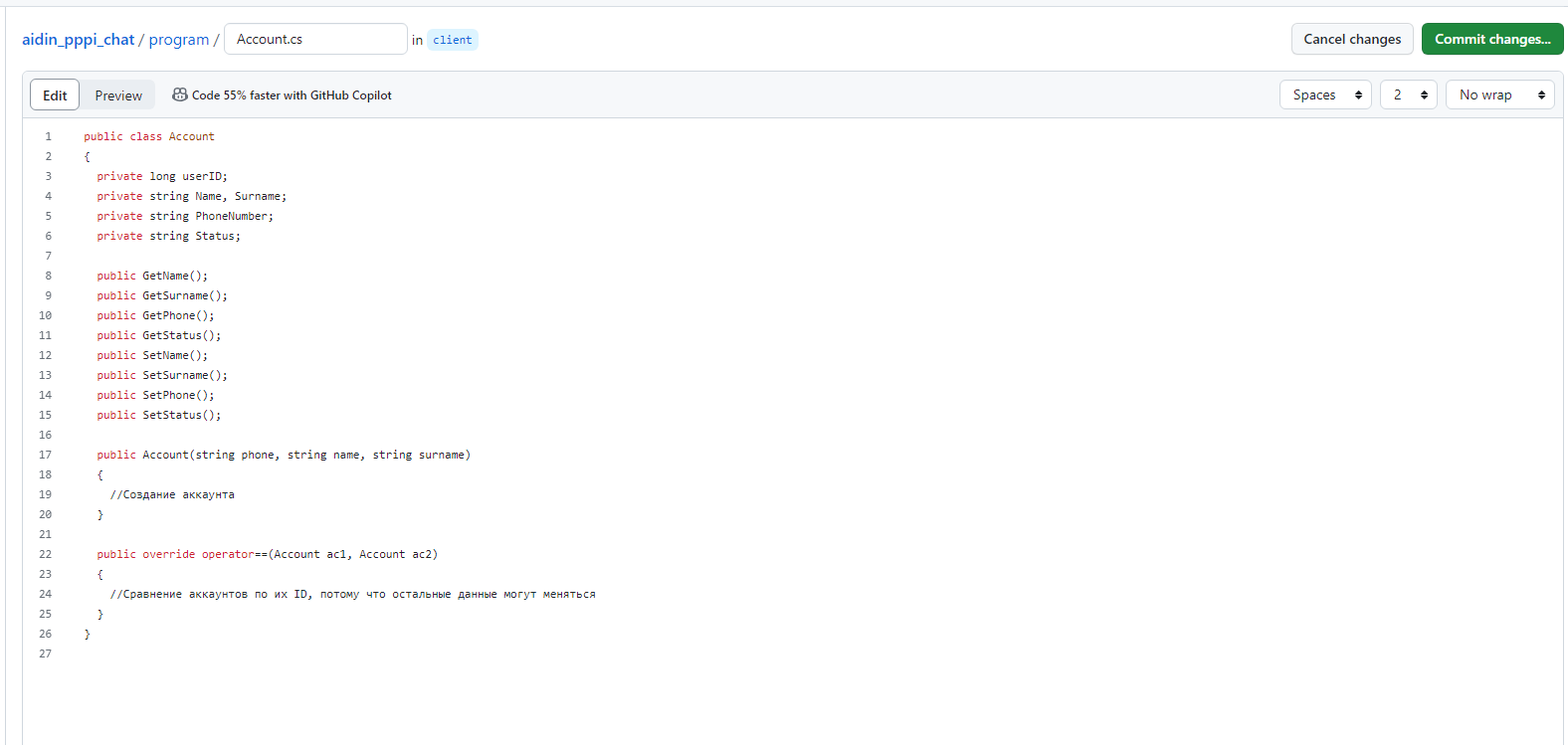


Рисунок 16 – Изменённый файл Accounts.cs

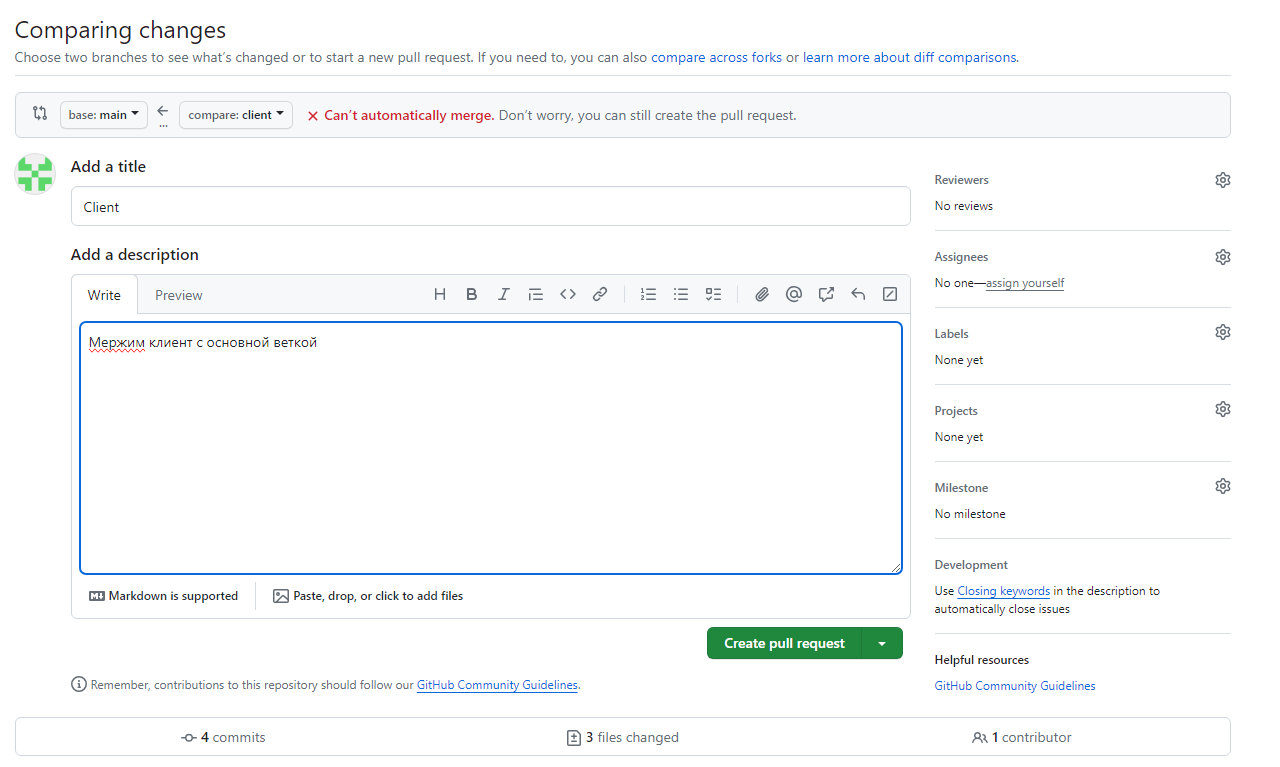


Рисунок 17 – Пулл-реквест (уже видно конфликт)

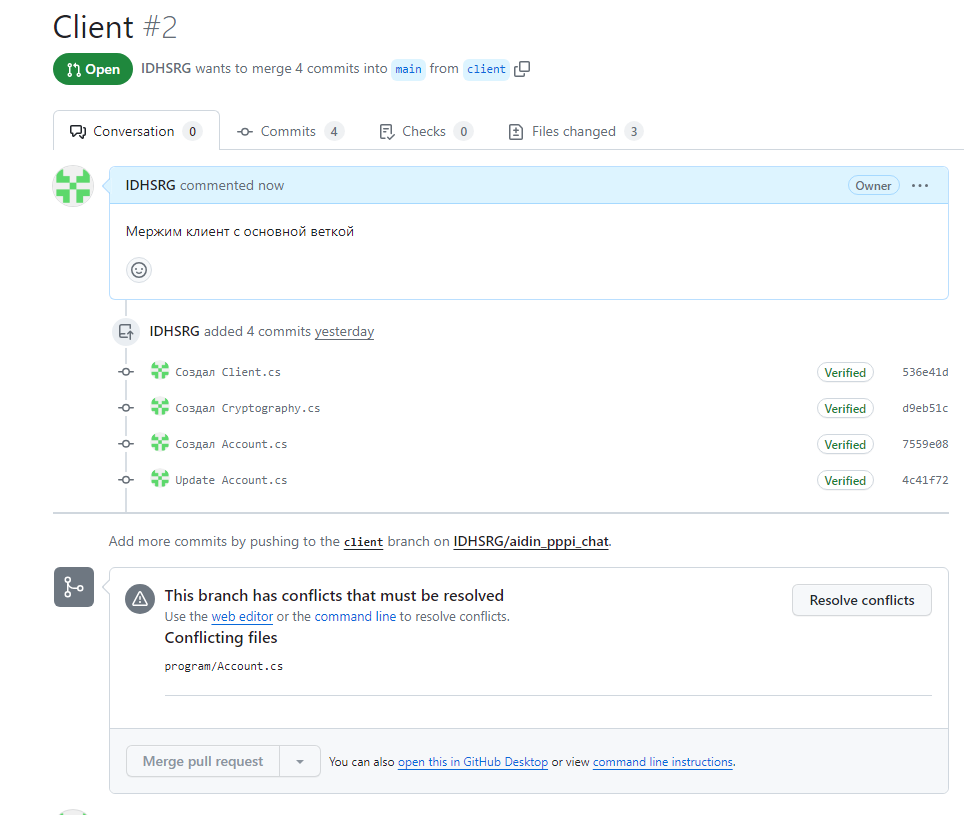


Рисунок 18 – Конфликт требует ручного решения перед объединением

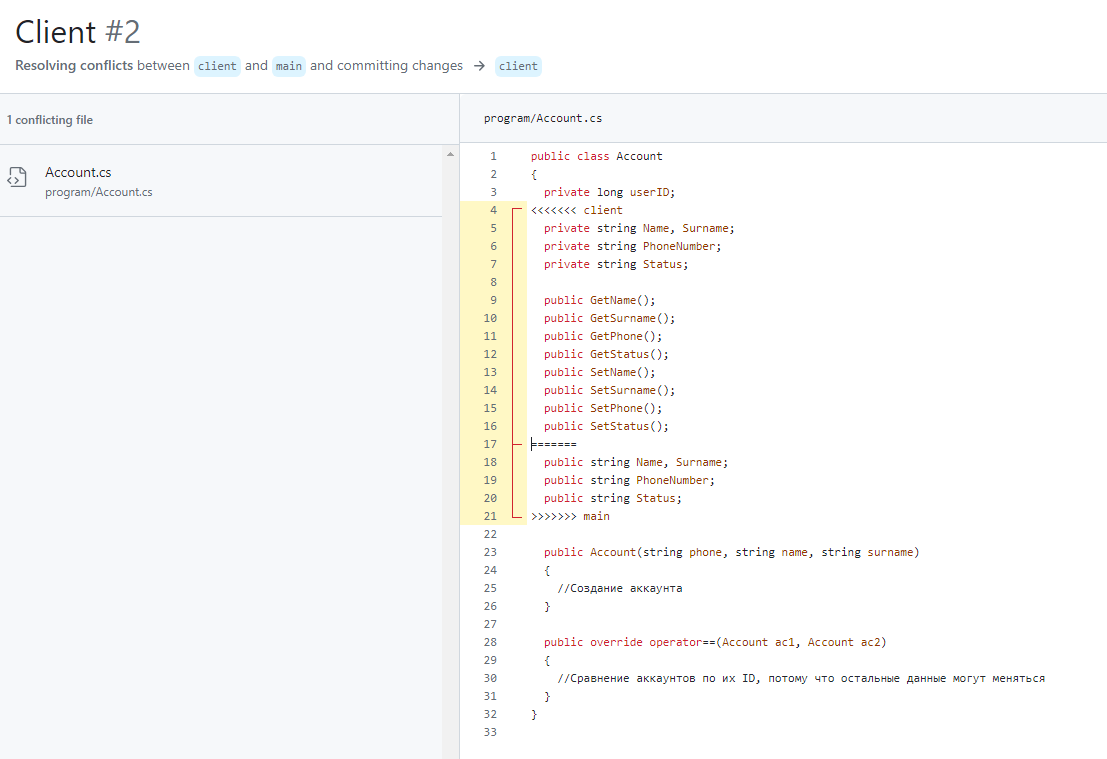


Рисунок 19 – Разница в строках при объединении веток

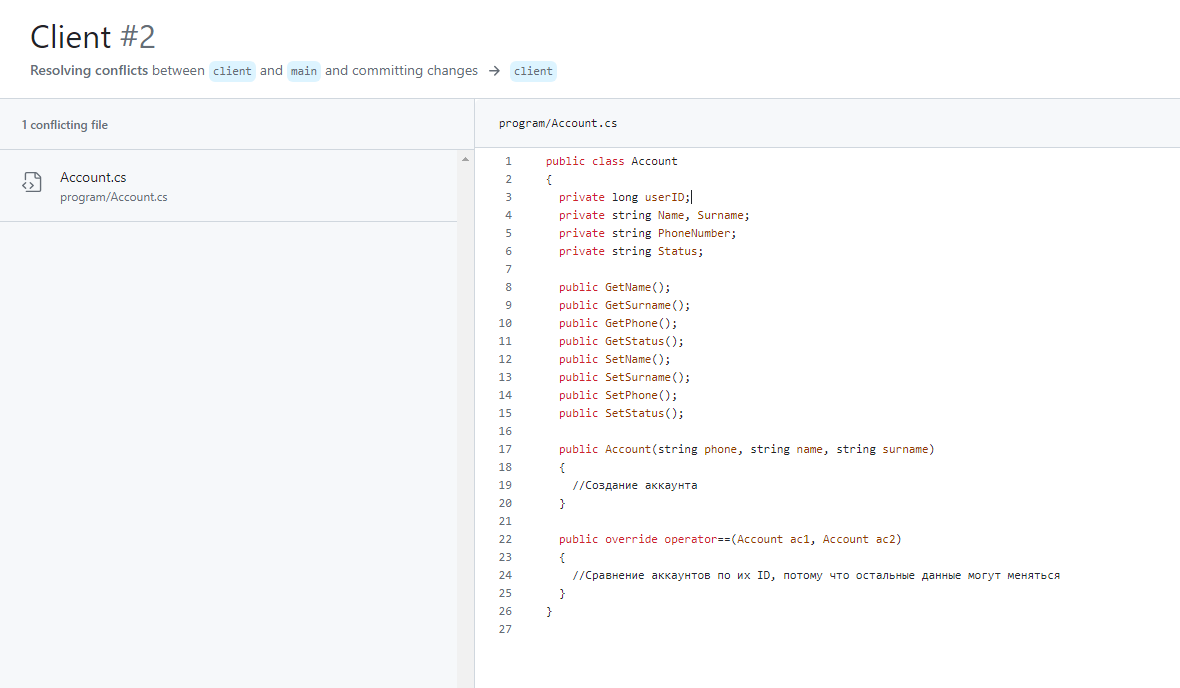


Рисунок 20 – Было решено взять верхний вариант (на деле можно написать что угодно)

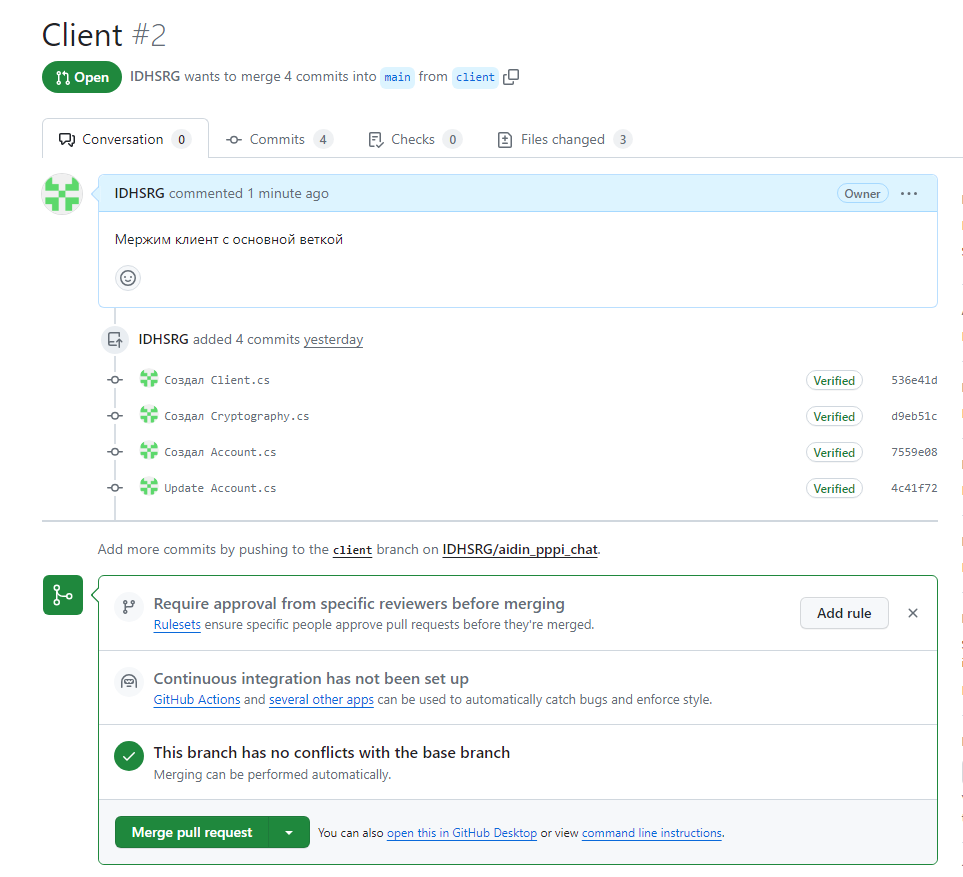


Рисунок 21 – Теперь мерж доступен

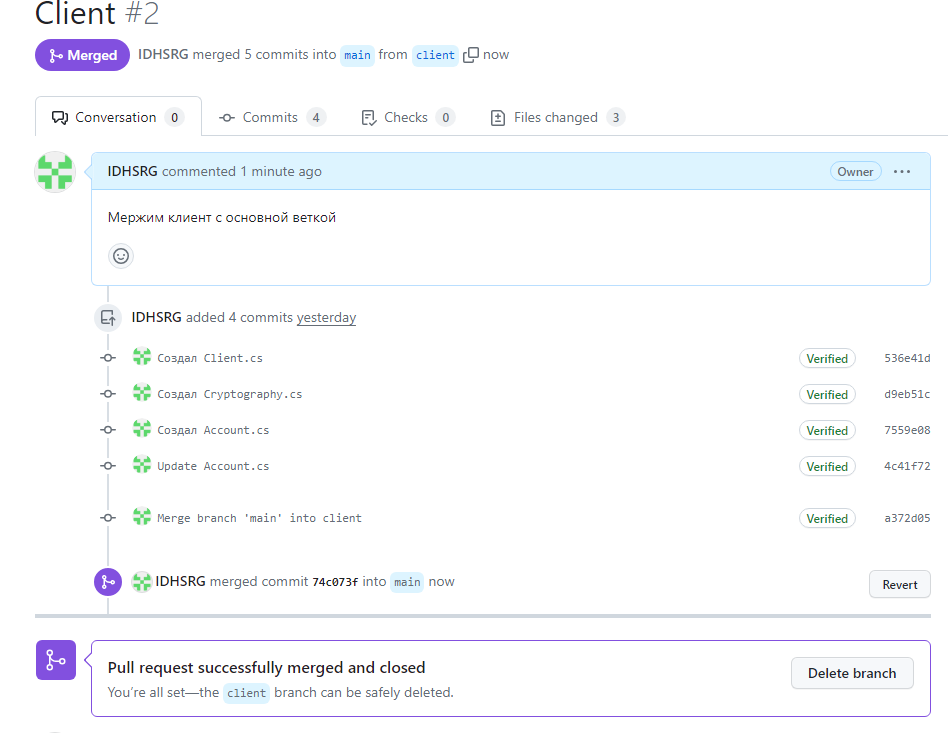


Рисунок 22 – Ветки объединены

Выполним объединение ветки main и interface таким же образом – конфликтов не возникнет. Теперь при просмотре веток мы видим, что они все были объединены с main и их можно удалить (пока что не удалены, мало ли править что-то надо будет по отчёту).

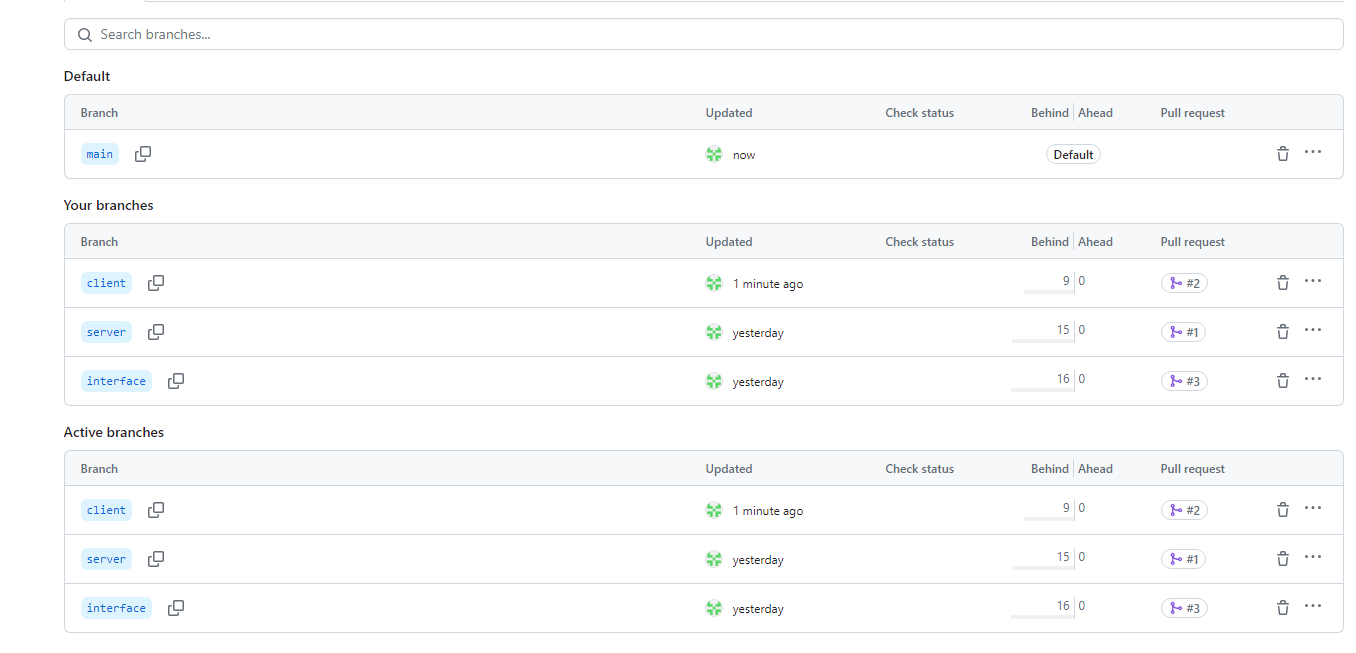


Рисунок 23 – Ветки

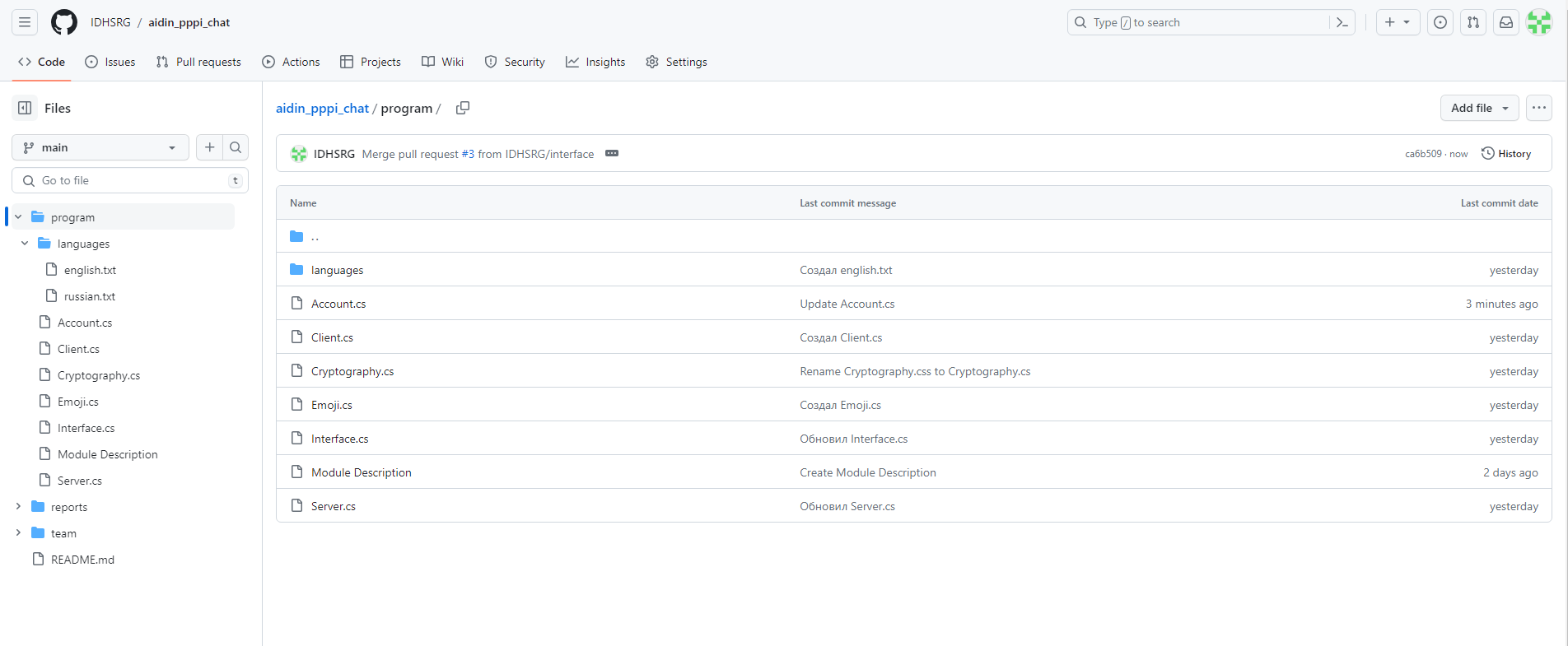


Рисунок 24 – Структура ветки main после объединений