МИНИСТЕРСТВО 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

на тему: «Работа с ветками»

по курсу: «Профессиональная практика программной инженерии»

Проверил:

асс. каф. ПИ им. Л.П.Фельдмана Филипишин Д.А.

Выполнил:

ст. гр. ПИ-20б

Кубасов Е.В.

Донецк-2024

Цель работы – познакомиться с основами использования веток в системе контроля версий Git.

Вариант 11. Генератор фракталов, с реализацией базовых алгоритмов построения фрактальных изображений (алгоритмические, геометрические и фракталы на основе метода IFS).

Аккаунт: EndoKardo, ссылка:

https://github.com/EndoKardo/Build\_Fractals

1. Распределим модули по веткам и укажем модули, принадлежащие к каждой ветке:

1. Работа с фракталами:
   * "Построение фракталов";
   * "Настройка построения фракталов";
   * "Выбор реализации фрактального построения";
   * "Алгоритмическое построение";
   * "Геометрическое построение";
   * "Построение на основе IFS";
   * "Тайм-трекинг";
   * "Отслеживание времени выполнения построения";
   * "Шаблоны фракталов";
   * "Стандартные шаблоны фракталов";
   * "Новые шаблоны фракталов";
   * "Шаблоны фракталов с нуля";
2. Анализ и работа с данными:
   * "Реальное время в облаке";
   * "Копирование данных";
   * "Резервное копирование данных";
   * "Копирование данных на облако";
   * "Тайм-трекинг";
   * "Отслеживание текущего времени".
3. Интеграция и пользовательский опыт:
   * "Чат-бот";
   * "Чат-бот для ответов";
   * "Чат-бот для быстрой работы с проектом";
   * "Связь с разработчиками";
   * "Связь с разработчиками для предложения нововведений";
   * "Связь с разработчиками по разным вопросам";
   * "Социальное взаимодействие";
   * "Социальное взаимодействие для обмена опытом";
   * "Социальное взаимодействие для обмена задачами";
   * "Социальное взаимодействие для общения с пользователями".

2. Создадим ветки на сайте GitHub (рис. 1-3).

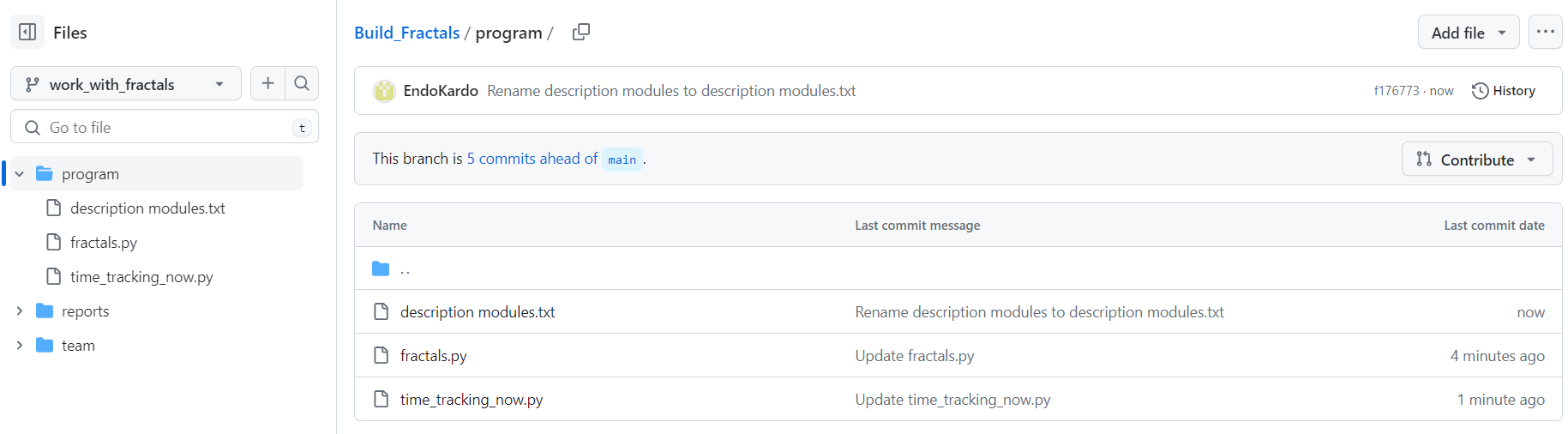


Рисунок 1 – Ветка «Работа с фракталами»

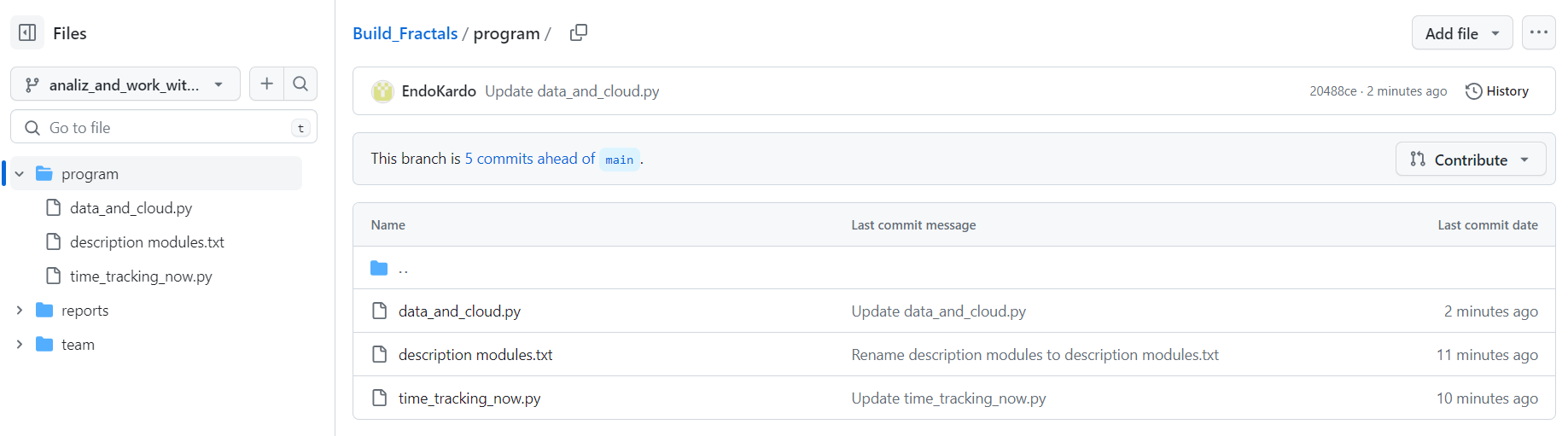


Рисунок 2 – Ветка «Анализ и работа с данными»

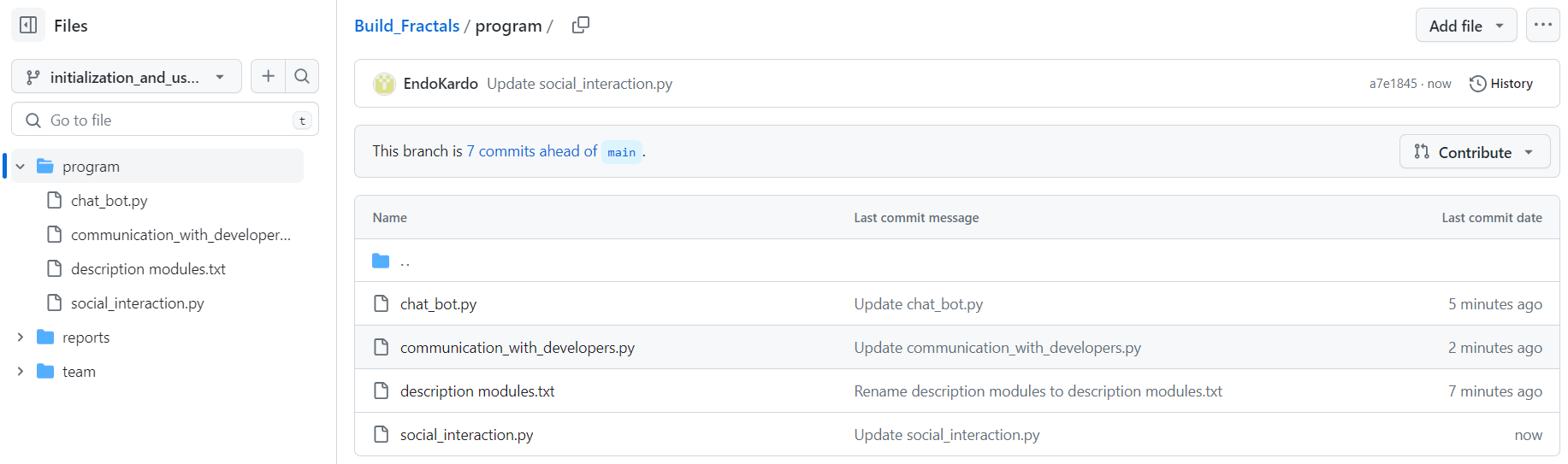


Рисунок 3 – Ветка «Интеграция и пользовательский опыт»

На рисунках 4-10 представлены файлы в ветках.

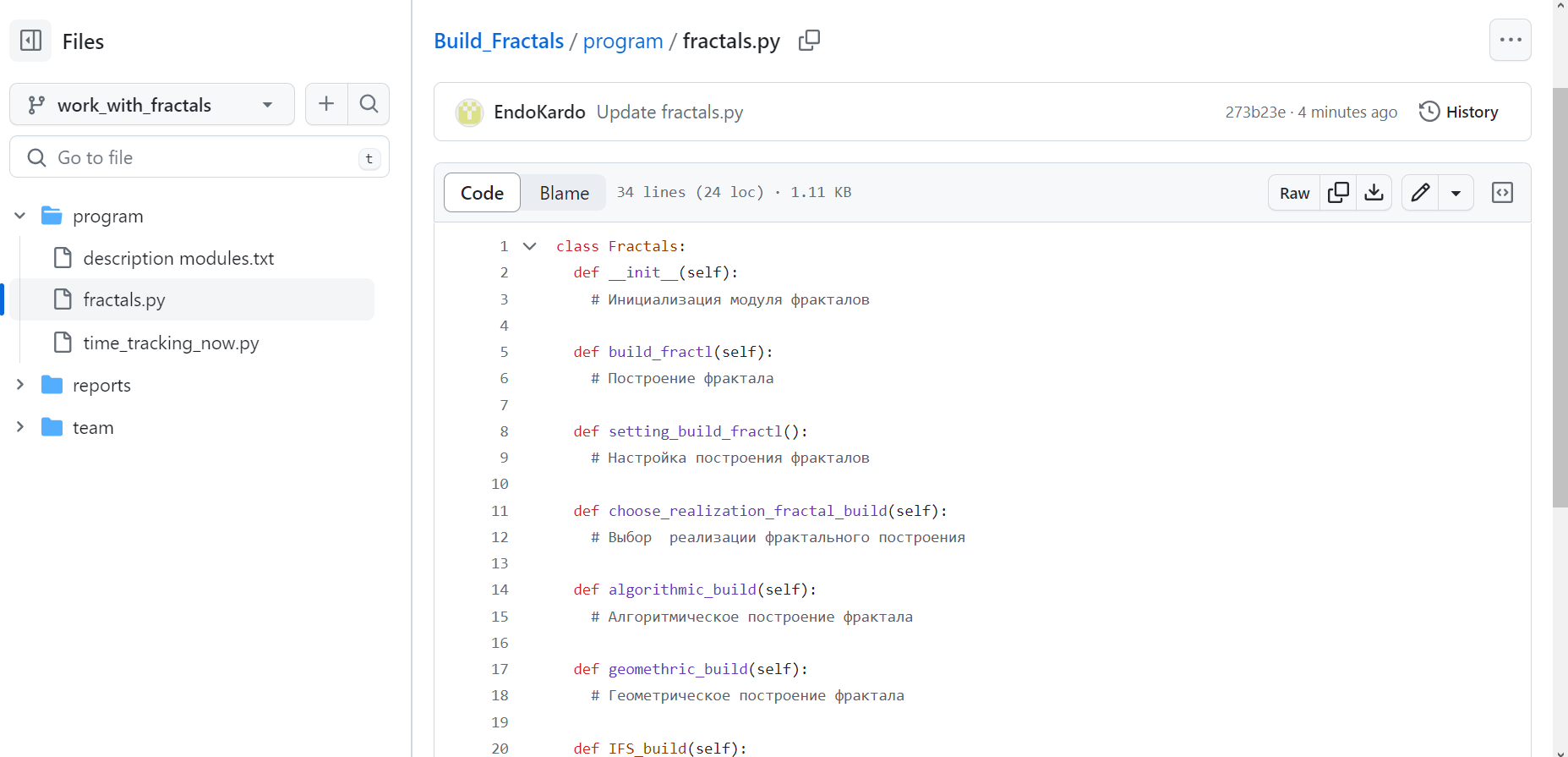


Рисунок 4 – Ветка «Работа с фракаталами», файл fractals.py

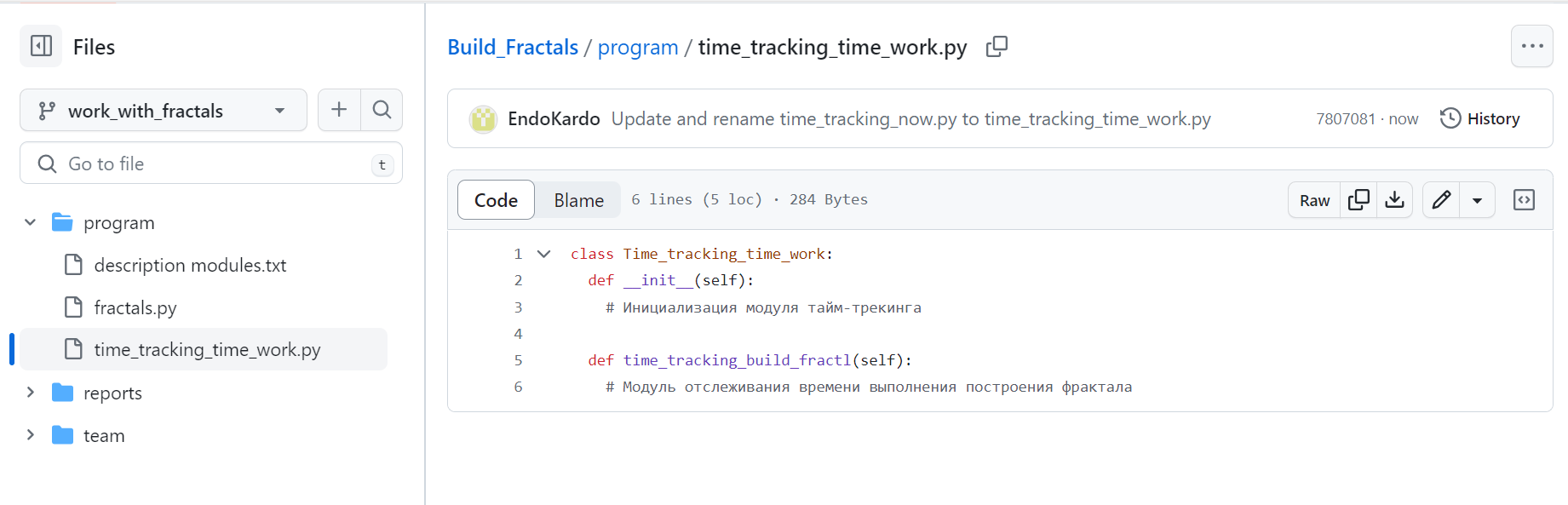


Рисунок 5 – Ветка «Работа с фракталами», файл time\_tracking\_time\_work.py

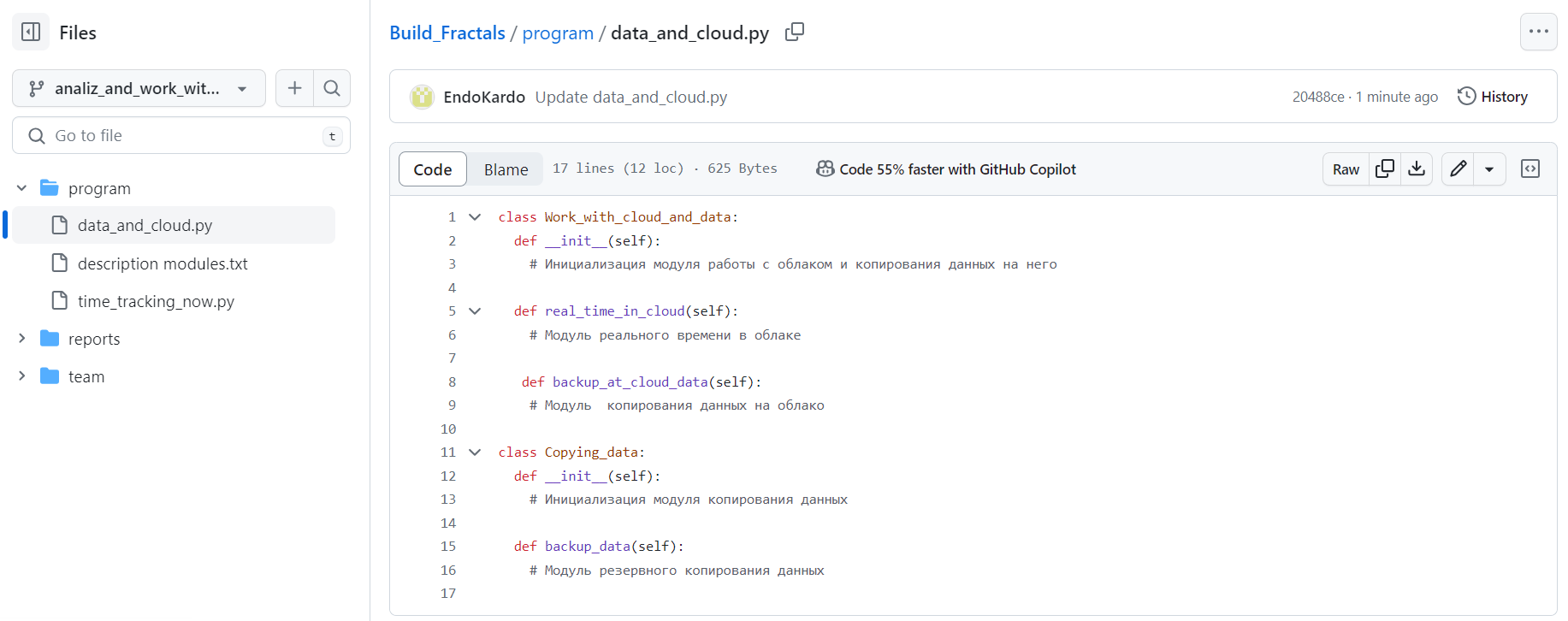


Рисунок 6 – Ветка «Анализ и работа с данными», файл data\_and\_cloud.py

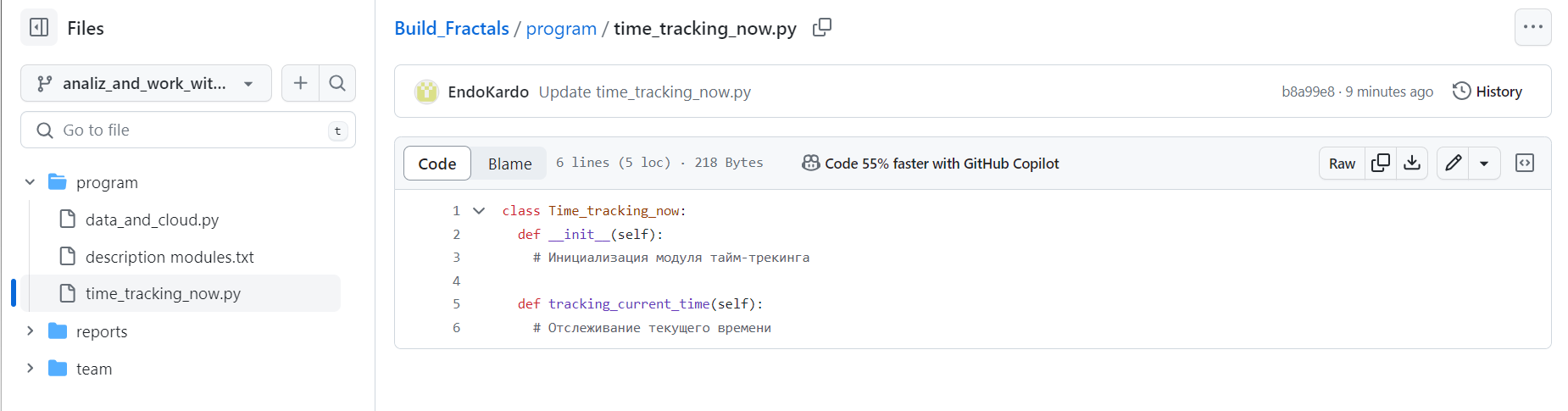


Рисунок 7 – Ветка «Анализ и работа с данными», файл time\_tracking\_now.py

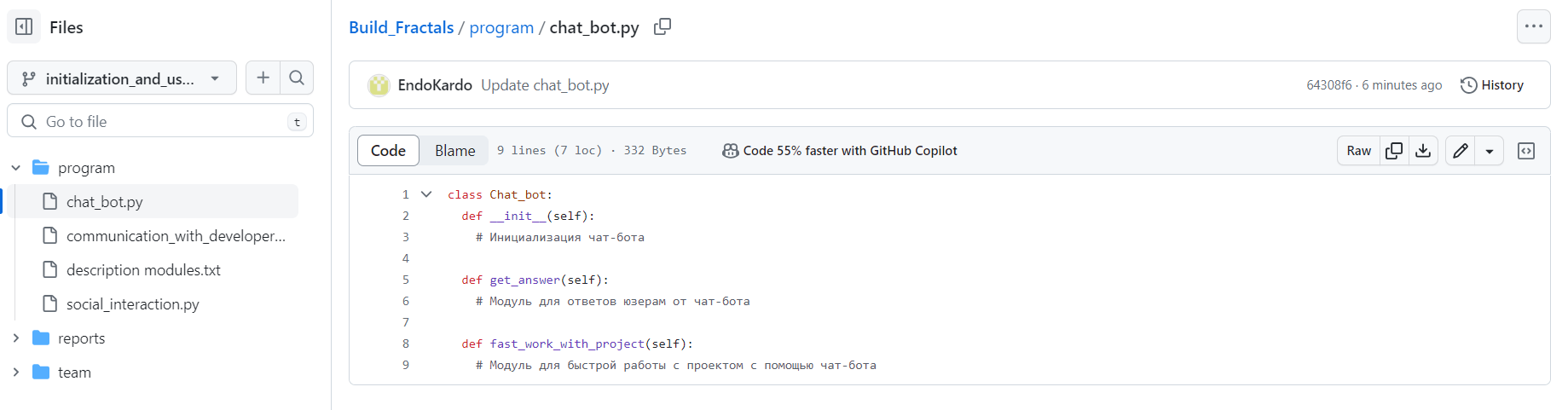


Рисунок 8 – Ветка «Интеграция и пользовательский опыт», файл chat\_bot.py

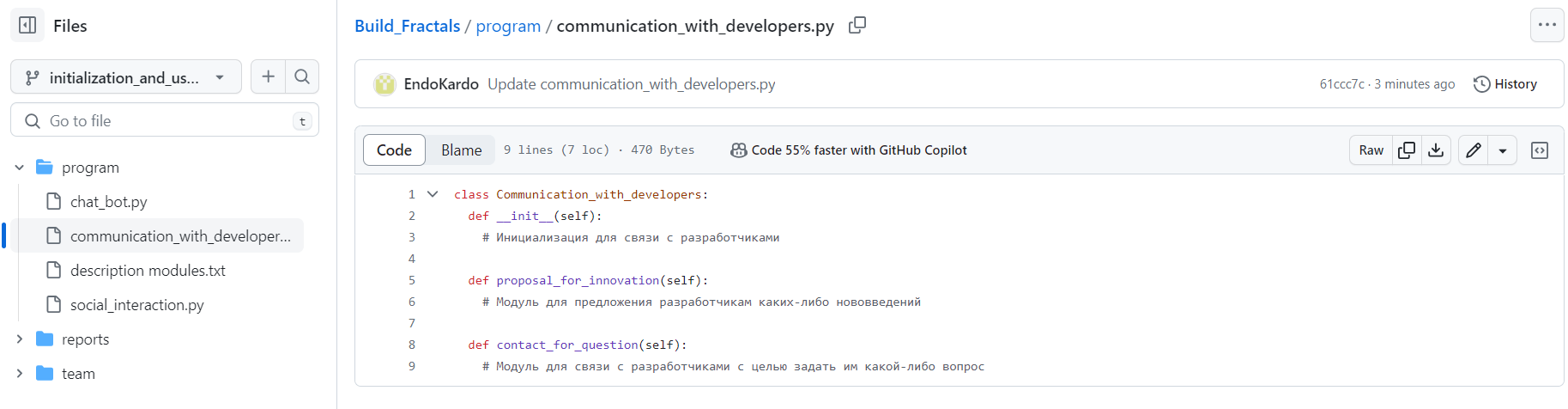


Рисунок 9 – Ветка «Интеграция и пользовательский опыт», файл communication\_with\_developers.py

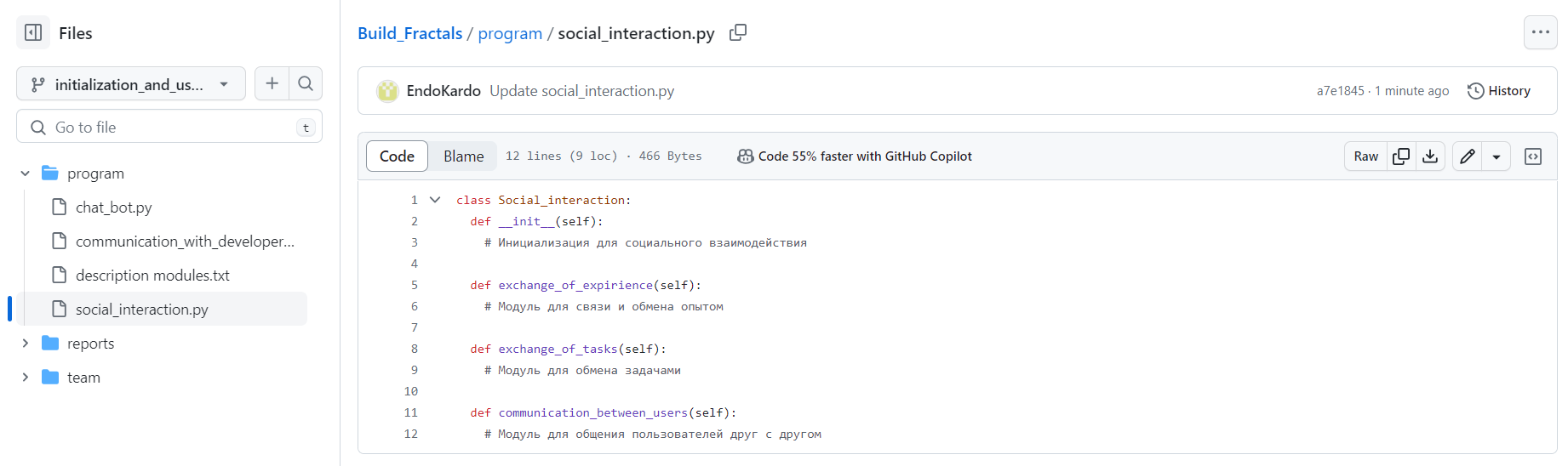


Рисунок 10 – Ветка «Интеграция и пользовательский опыт», файл social\_interaction.py

На рисунках 11-13 представлены истории коммитов для каждой ветки.

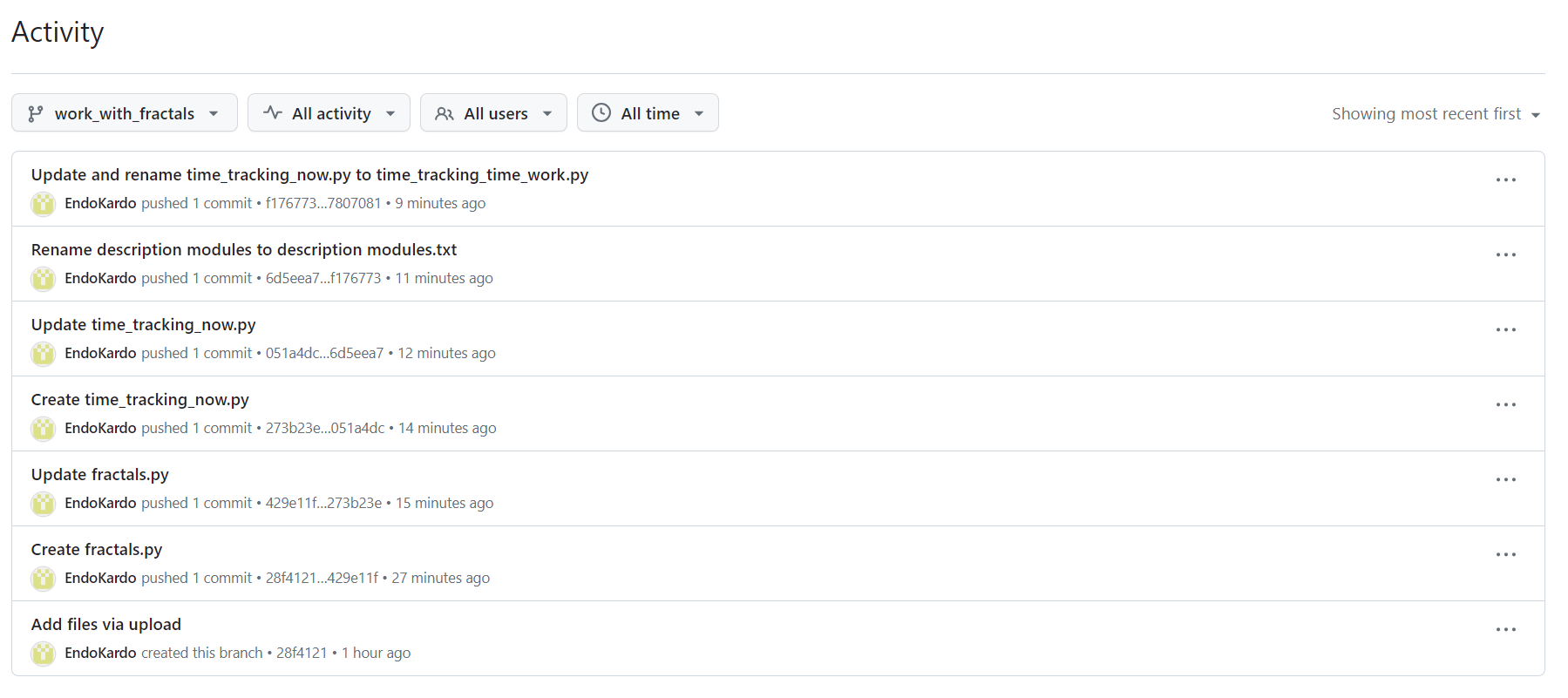


Рисунок 11 – История коммитов ветки «Работа с фракталами»

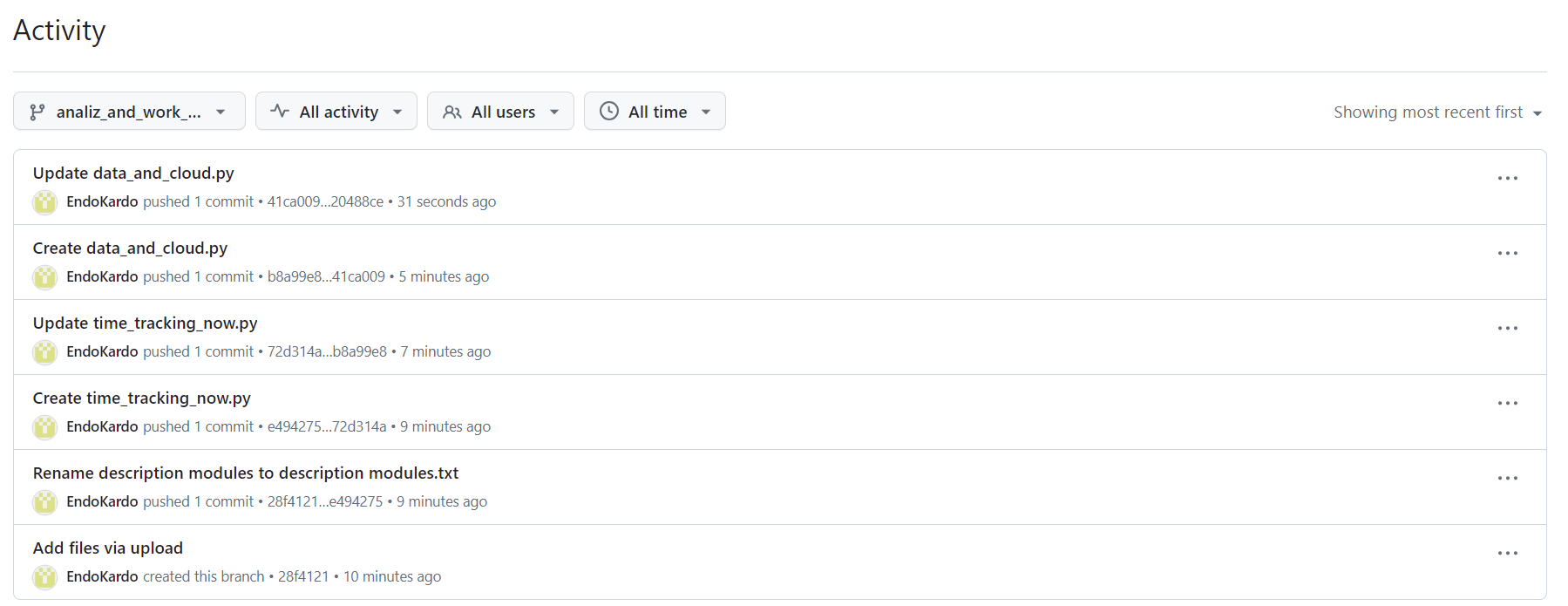


Рисунок 12 – История коммитов ветки «Анализа и работа с данными»

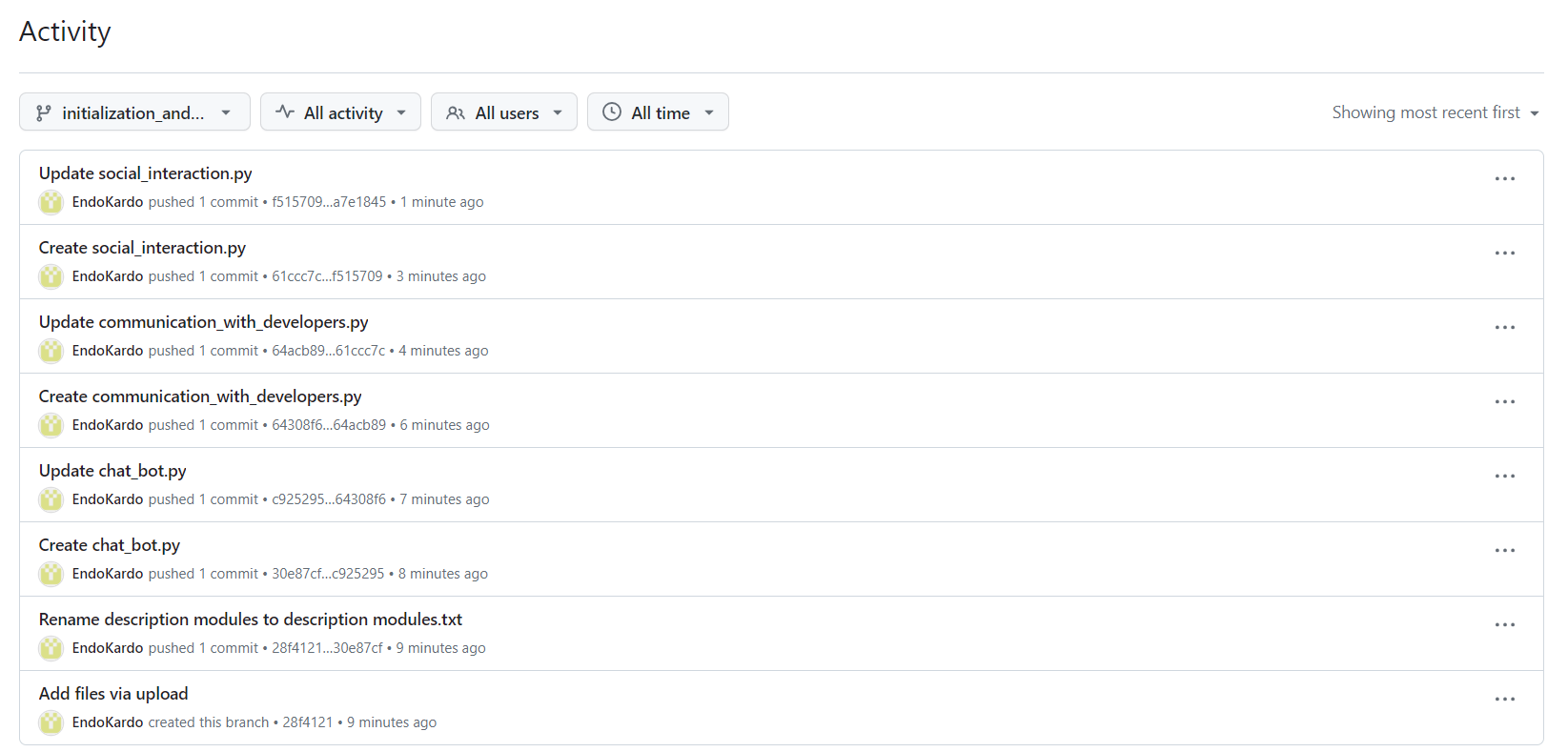


Рисунок 13 – История коммитов ветки «Интеграция и пользовательский опыт»

Далее произведём merge ветки main с веткой work\_with\_fractals. Создадим pull request (рис. 14).

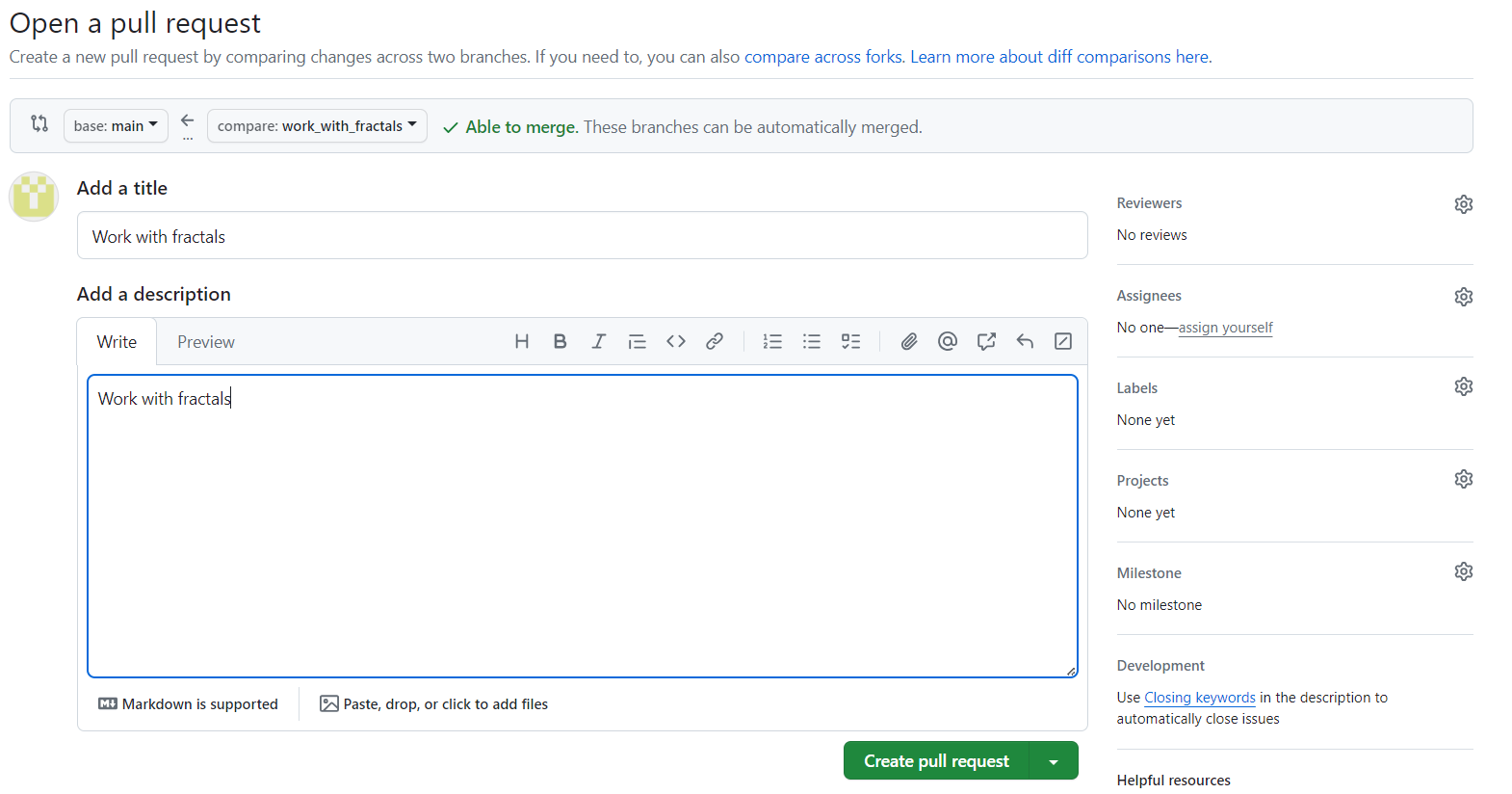


Рисунок 14 – Создание pull request

После нажатия на кнопку «Create pull request» на экране появляется информация об отсутствии конфликтах между ветками – это означает, что merge возможен (рис. 15).

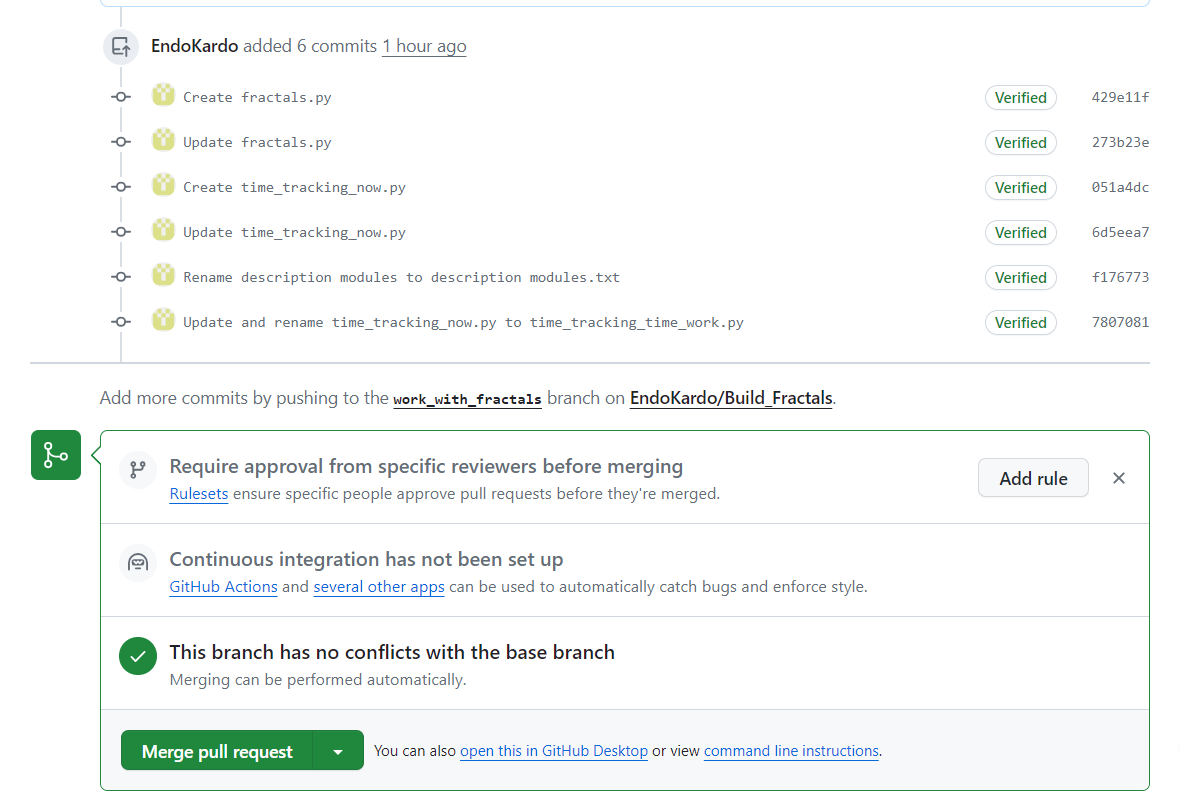


Рисунок 15 – Уведомление о возможности merge

После нажатия на кнопку «Merge pull request» и подтверждения действия, merge будет успешно выполнен (рис. 16).

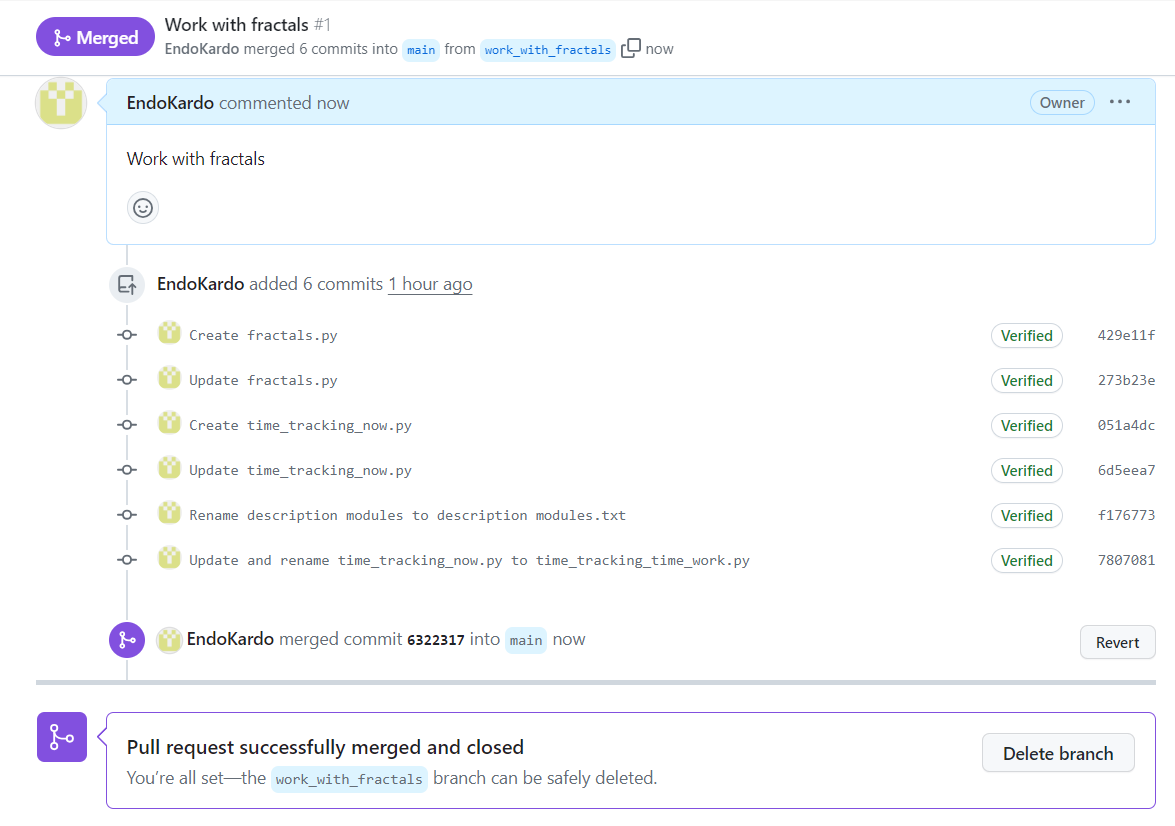


Рисунок 16 – Успешное выполнение merge

Создадим конфликтную ситуацию – в ветку analiz\_and\_work\_with\_data добавим файл fractals.py (этот файл ранее был в ветке work\_with\_fractals, которую включили в ветку main).

Попробуем произвести merge ветки main с веткой analiz\_and\_work\_with\_data. Уже при создании pull request можно заметить сообщение **«Can’t automatically merge»** (рис. 17).

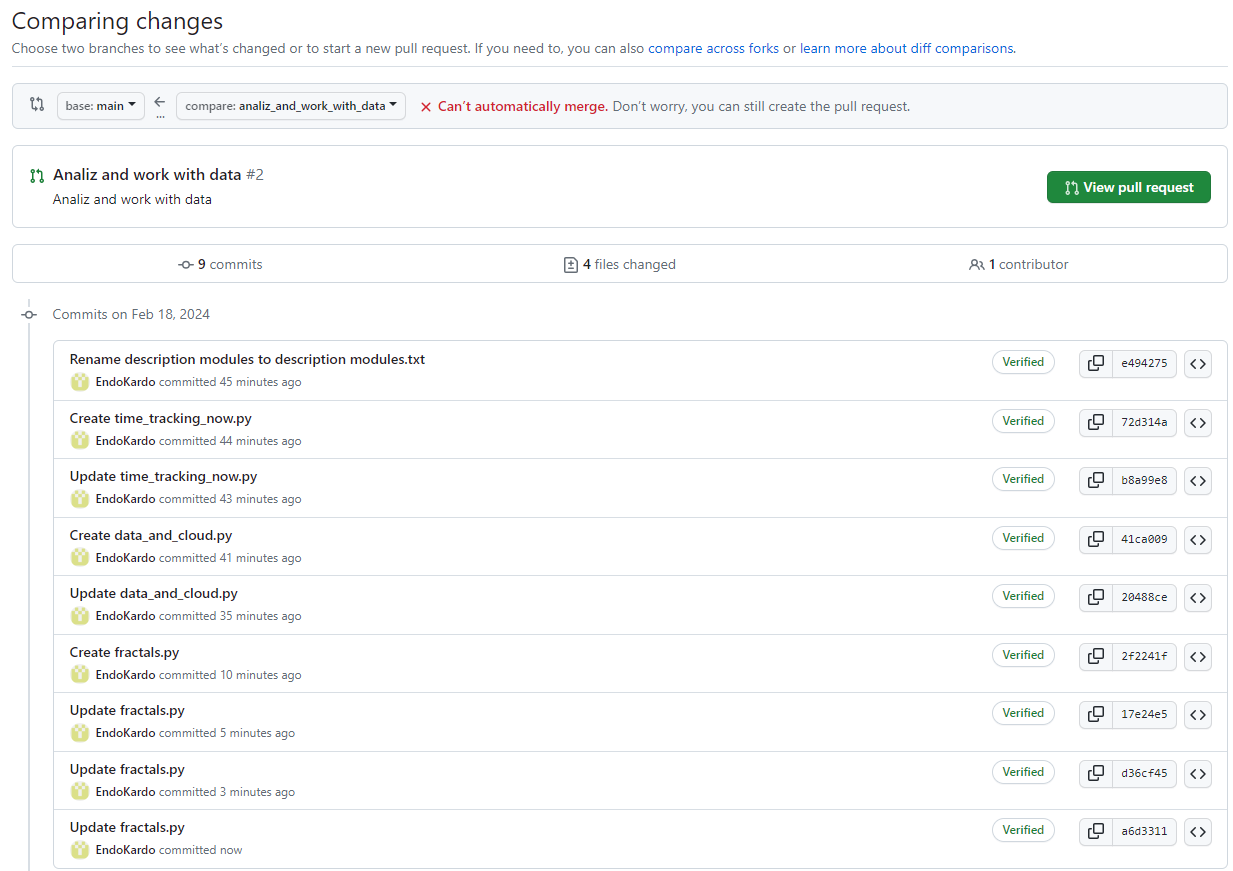


Рисунок 17 – Создание pull request с несовпадающими данными

После нажатия на кнопку «Create pull request» на экране появляется сообщение о наличие конфликта в ветке, для успешного merge этот конфликт должен быть решён (рис. 18).

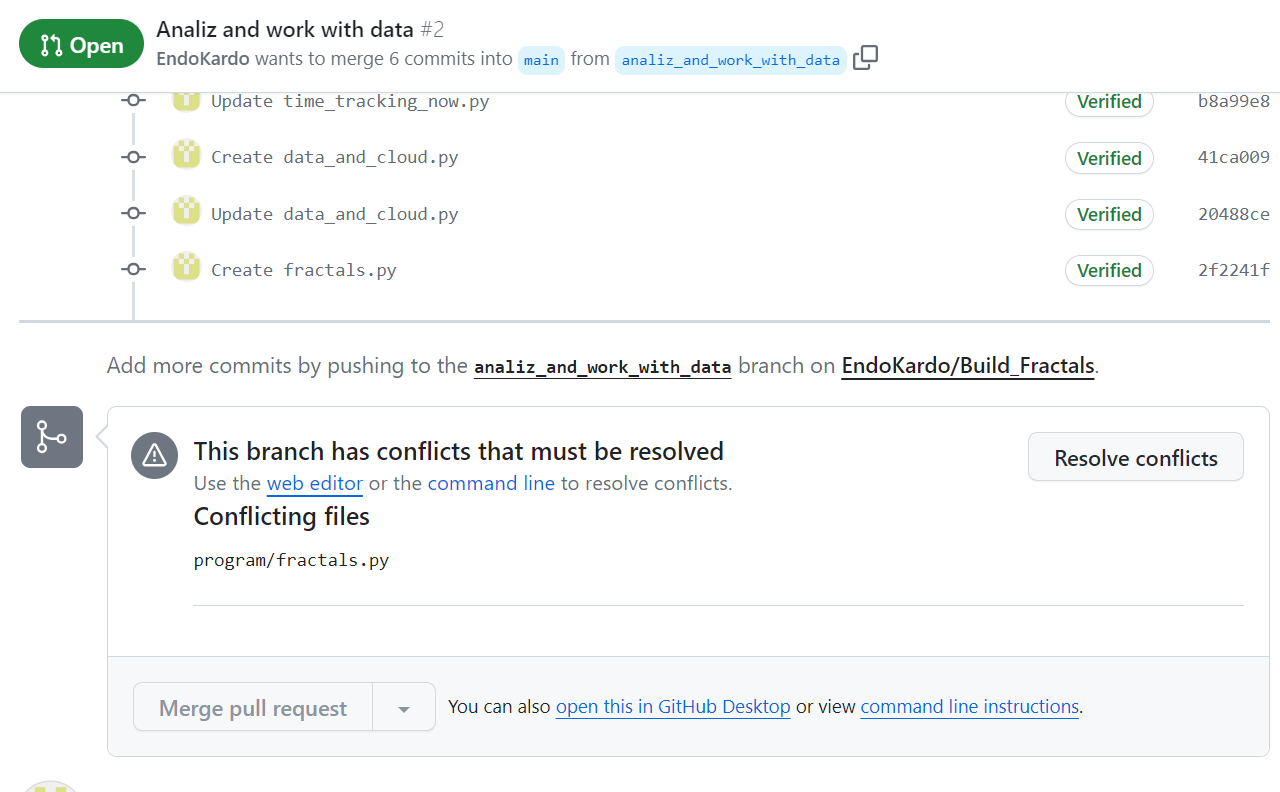


Рисунок 18 – Сообщение о наличии конфликта

Решение конфликта ручными способом предполагает выбор нужной части кода (рис. 19).

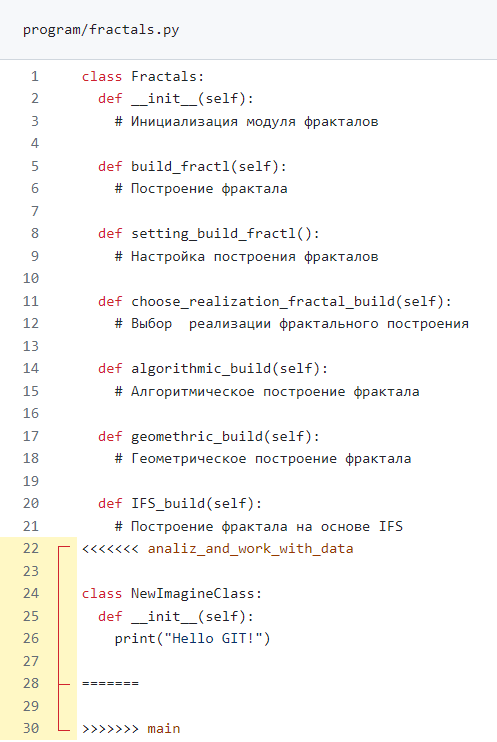


Рисунок 19 – Конфликт в коде

В данном случае, выберем вторую часть кода и нажмём на кнопку «Mark as resolved» (рис. 20).

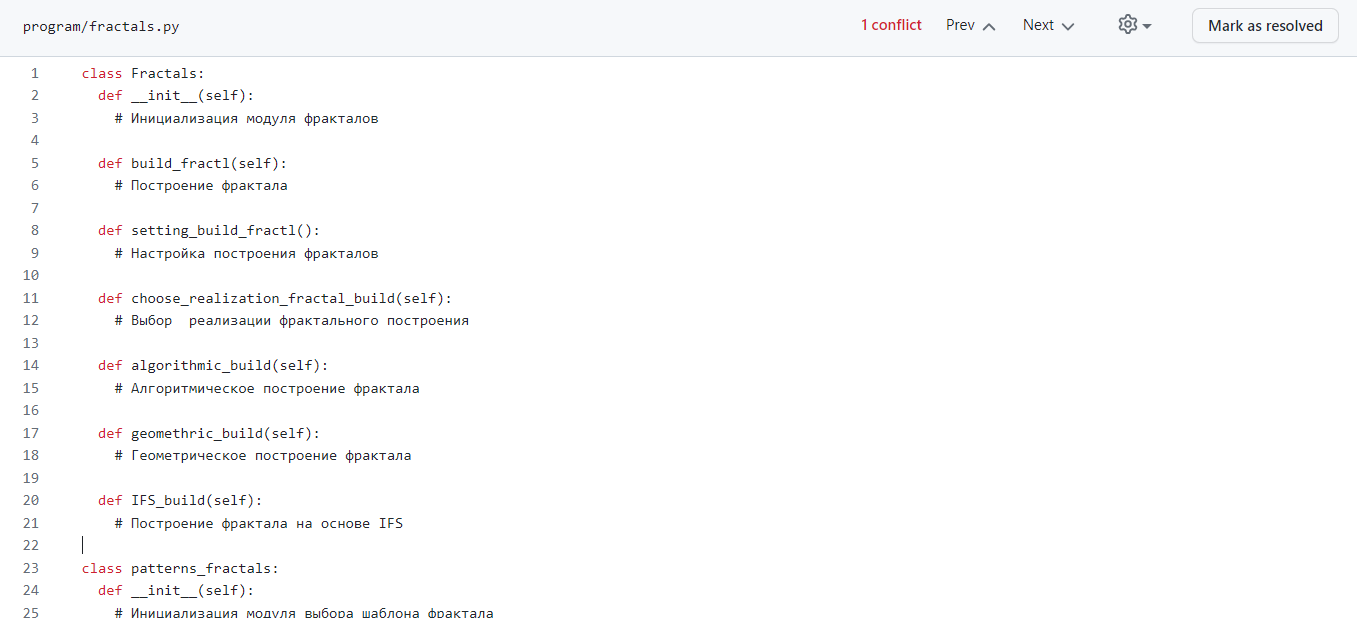


Рисунок 20 – Решение конфликта

Попробуем выполнить новый commit и получим успешный результат (рис. 21).

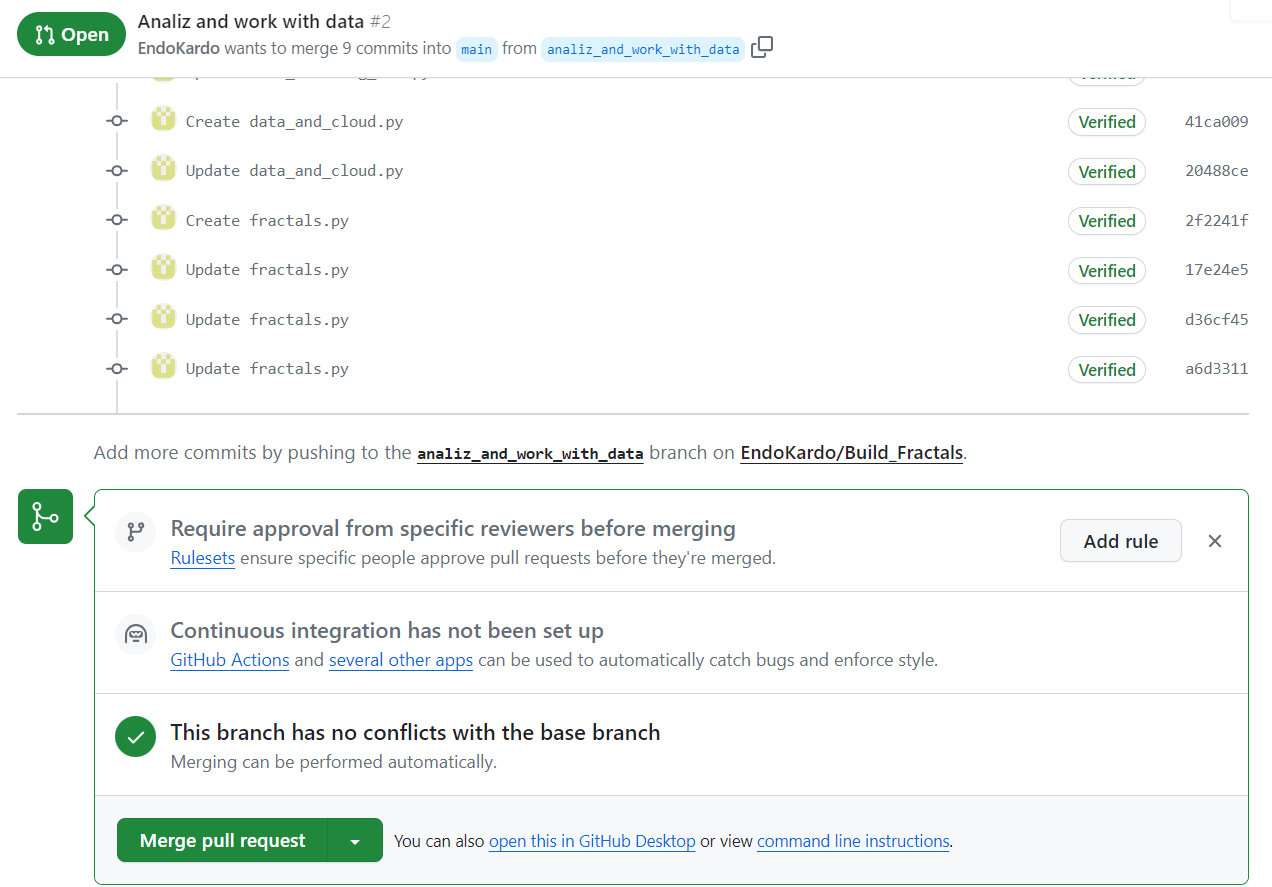


Рисунок 21 – Успешное решение конфликта

По тому же принципу объединим ветку initialization\_and\_user\_expirience с main и получим результат – все ветки были объединены в main (рис. 22).

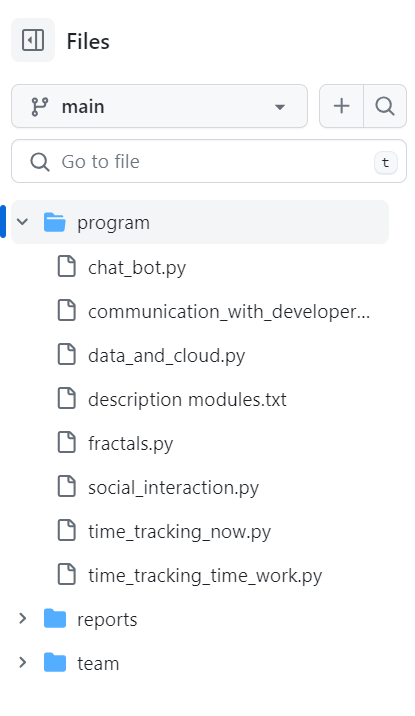


Рисунок 22 – Результат объединения всех веток