Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра инфокоммуникаций

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

**дисциплины «Основы программной инженерии»**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Бересланов Рамазан  2 курс, группа ПИЖ-б-о-22-1,  09.03.04 «Программная инженерия», направленность (профиль) «Разработка и сопровождение программного обеспечения», очная форма обучения  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Руководитель практики:  Богданов С.С., ассистент кафедры инфокоммуникаций  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ставрополь, 2023 г.

Тема: Исследование возможностей Git для работы с локальными репозиториями.

Цель работы: исследовать базовые возможности системы контроля версий для работы с локальными репозиториями.

Ход выполнения работы:

1. Создать общедоступный репозиторий на GitHub с использованием лицензии MIT и выбранного языка программирования:

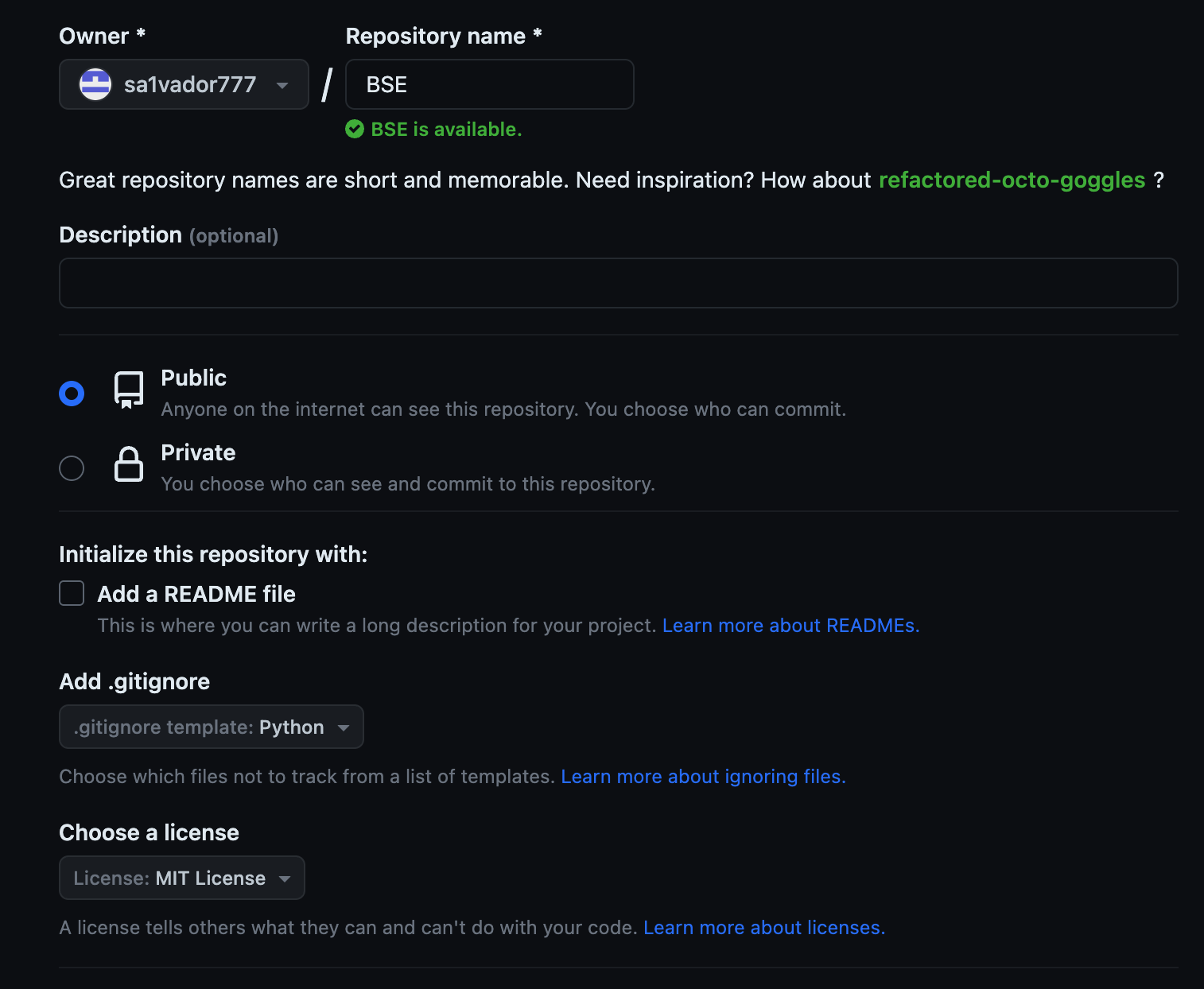


Рисунок 1 – Создание общедоступного репозитория на GitHub

1. Клонировать репозиторий на рабочий компьютер:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 2 – Клонирование созданного репозитория на локальный компьютер

1. Дополнить файл .gitignore необходимыми правилами для выбранного языка:

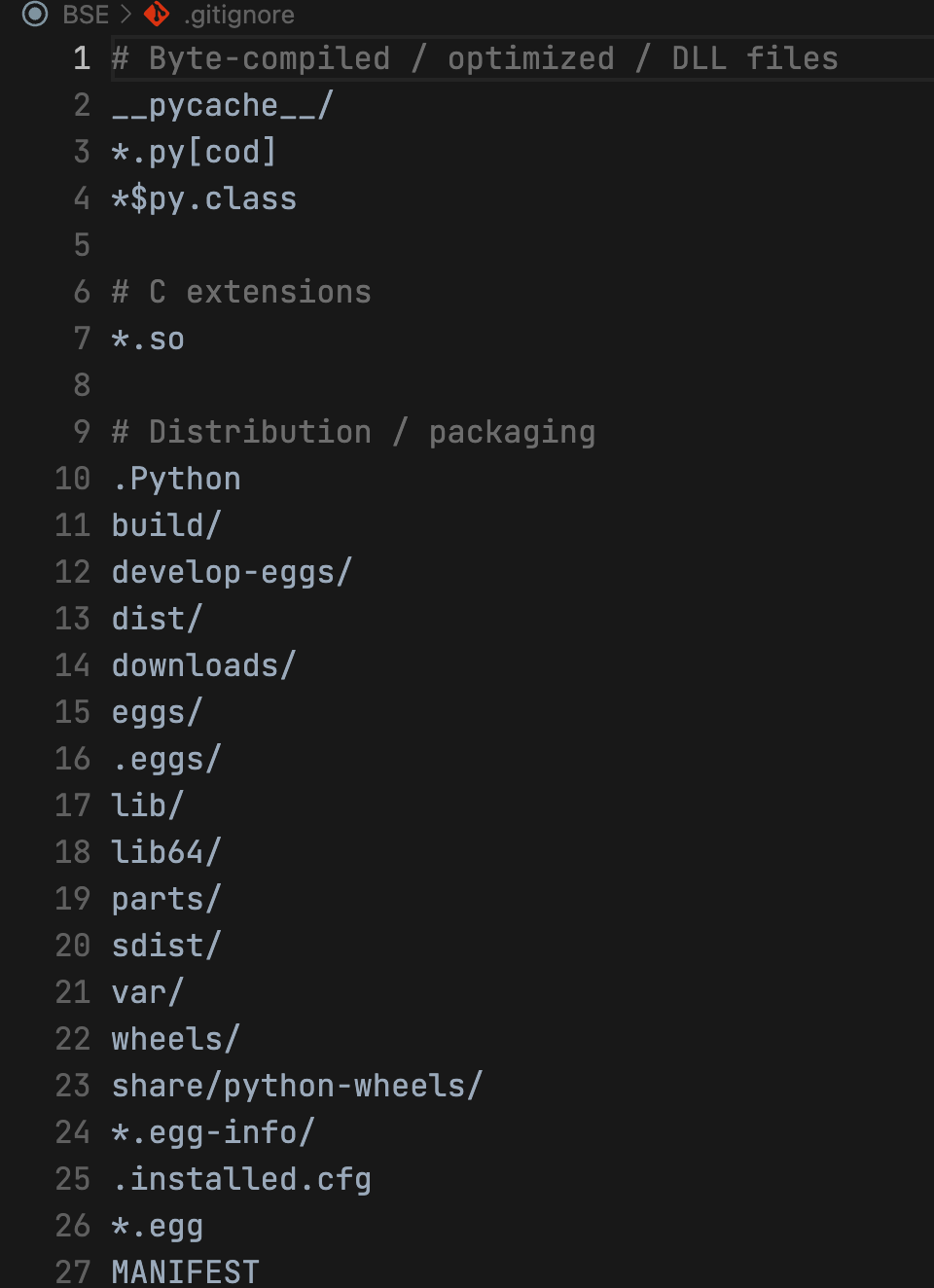


Рисунок 3 – Часть созданных, игнорируемых файлов и расширений для Python и IDE

1. Добавить в файл README.md информацию о ФИО автора

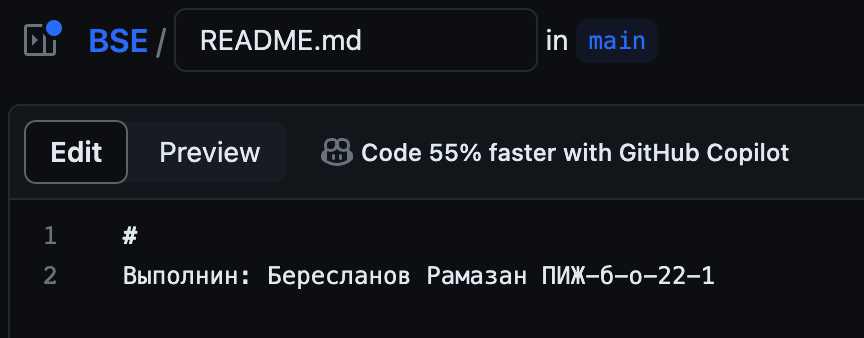


Рисунок 4 – Добавление информации в README.md

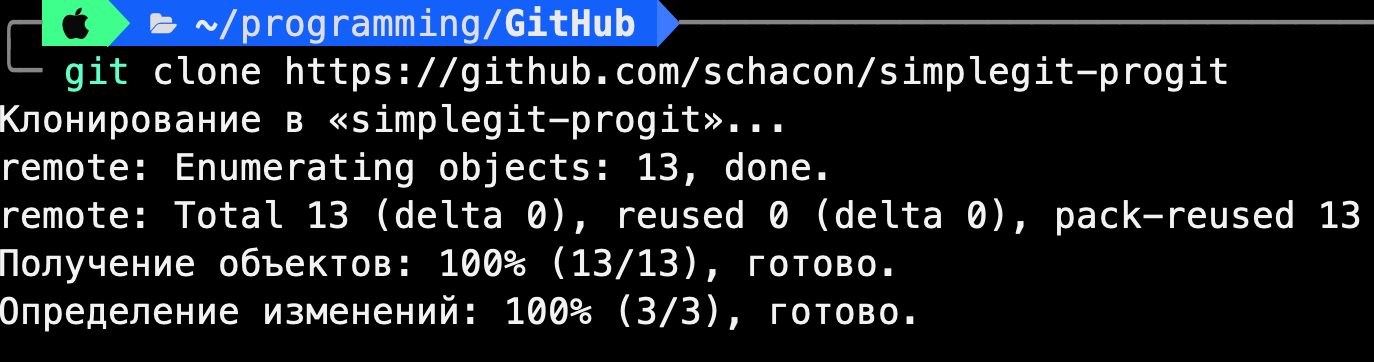
1. Проработаем примеры лабораторной работы и отобразим вывод в консоли:

Рисунок 5 – Клонирование проекта «simplegit» с GitHub

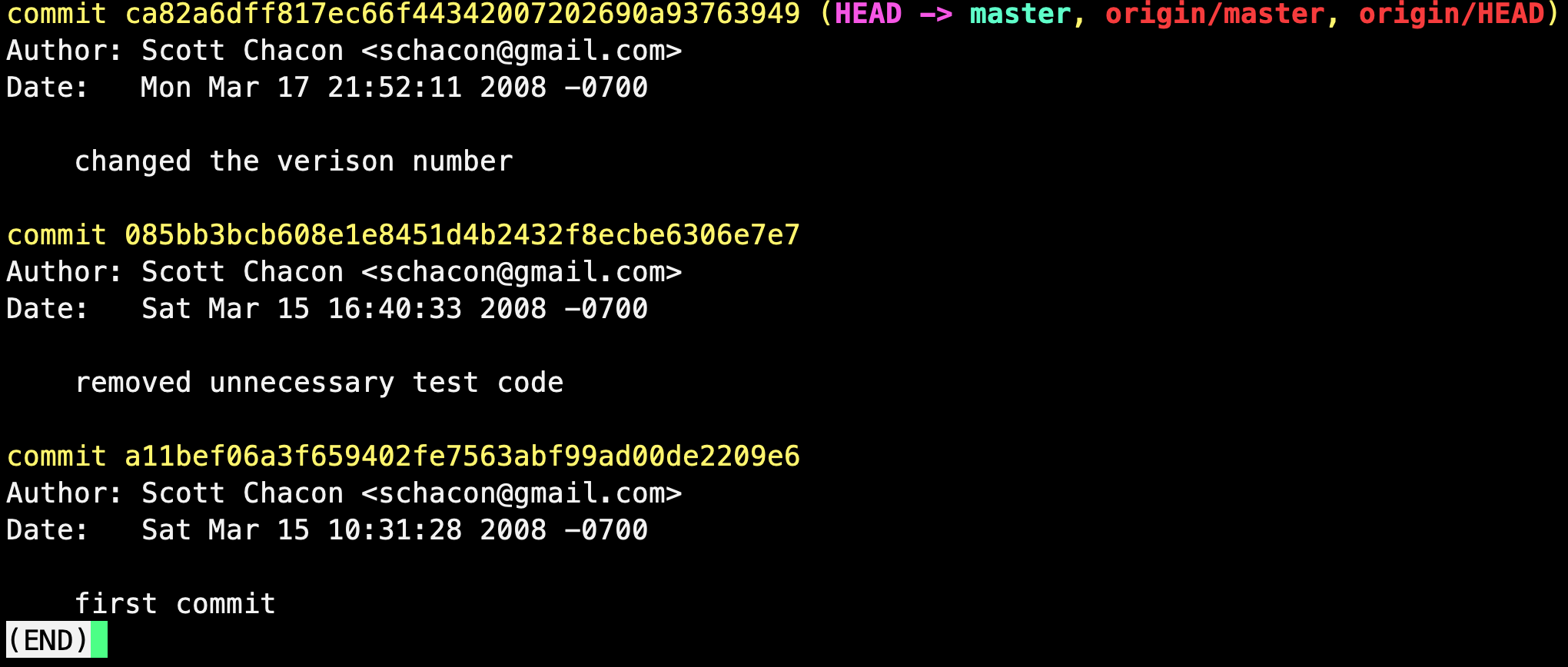


Рисунок 6 – Просмотр истории коммитов

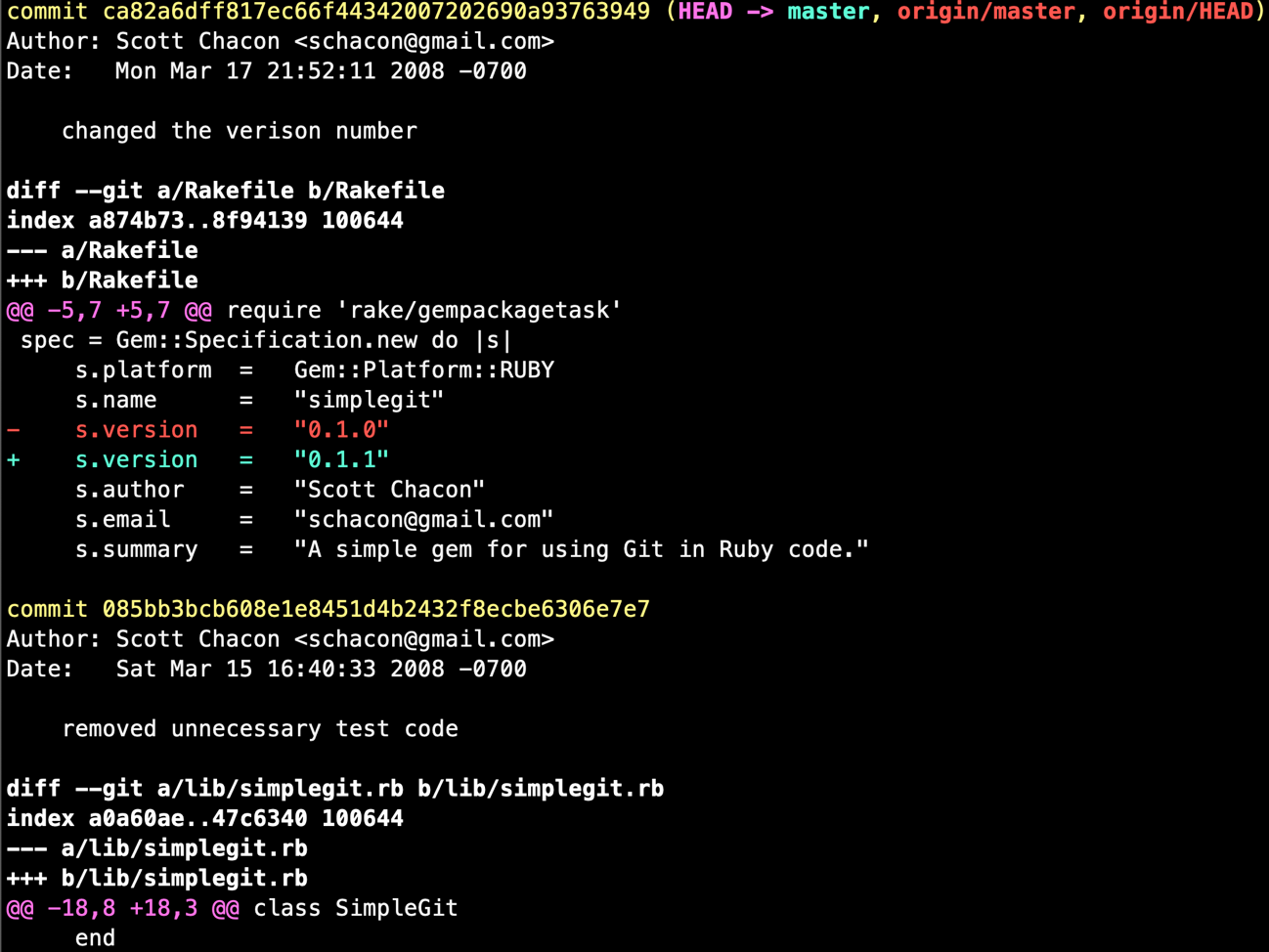


Рисунок 7 – Просмотр изменений в файлах в последних двух коммитах



Рисунок 8 – Просмотр сокращенной статистики каждого коммита

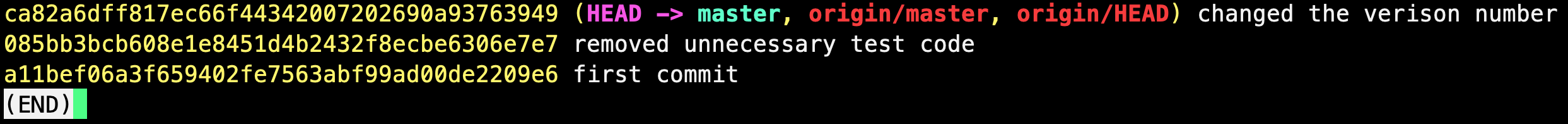


Рисунок 9 – Изменение формата вывода информации о коммитах (1)

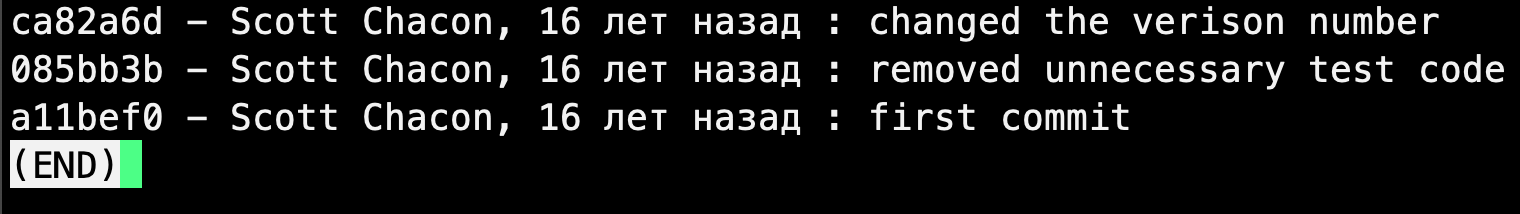


Рисунок 10 – Изменение формата вывода информации о коммитах (2)

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 11 – Вывод коммитов, которые меняли файл Rakefile

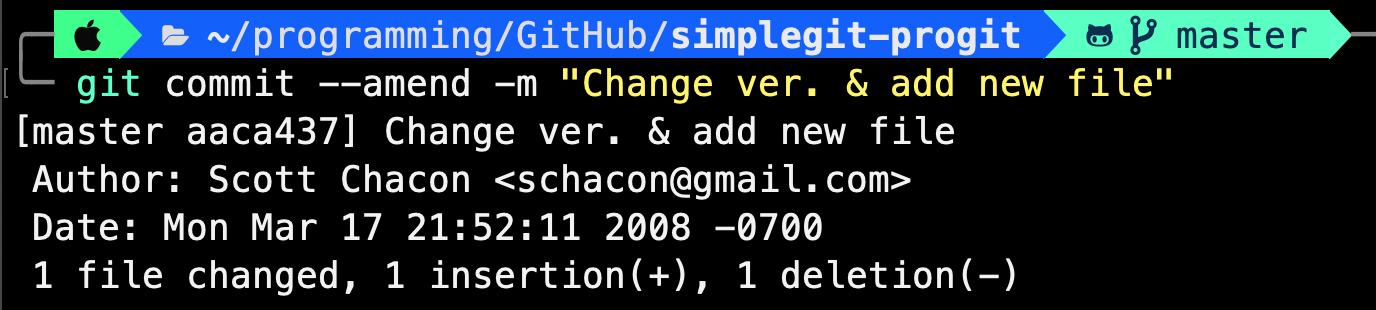


Рисунок 12 – Добавление нового файла и замена предыдущего коммита новым.

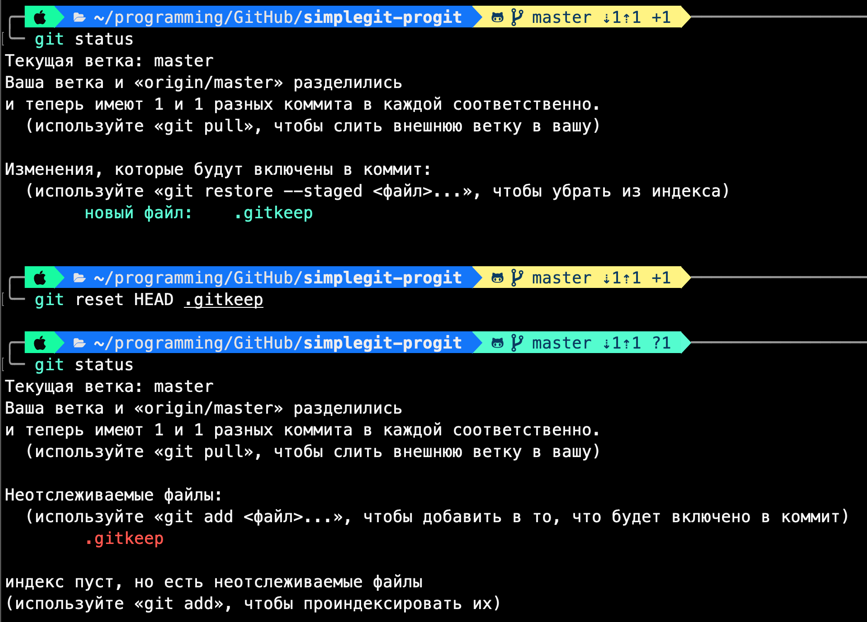


Рисунок 13 – Отмена индексации файла

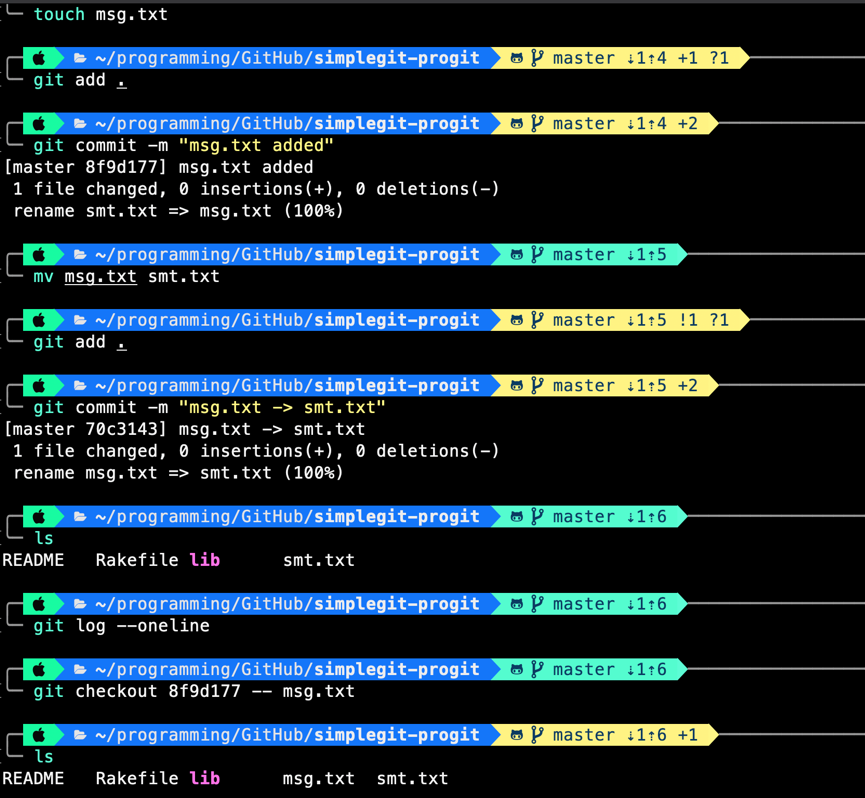


Рисунок 14 – Выполнение отката изменений



Рисунок 15 – Просмотр названий доступных удаленных репозиториев

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 16 – Просмотр адресов для чтения и записи, привязанных к репозиторию

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 17 – Добавление удаленного репозитория

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 18 – Получение изменений с удаленного репозитория

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 19 – Просмотр удаленного репозитория

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 20 – Переименование удаленного репозитория

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 21 – Удаление удаленного репозиторияИзображение выглядит как текст, снимок экрана, Шрифт, Графика

Автоматически созданное описание

Рисунок 22 – Создание аннотированного тега

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 23 – Просмотр тега и привязанного к нему коммиту

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 24 – Удаление тега

1. Напишем небольшую программу на Python и зафиксируем изменения не менее 7 раз, а также отметим 3 тэгами. Подтвердим фиксирование изменений командой git:



Рисунок 25 – Подтверждение наличия изменений

1. Посмотрим содержимое коммита HEAD

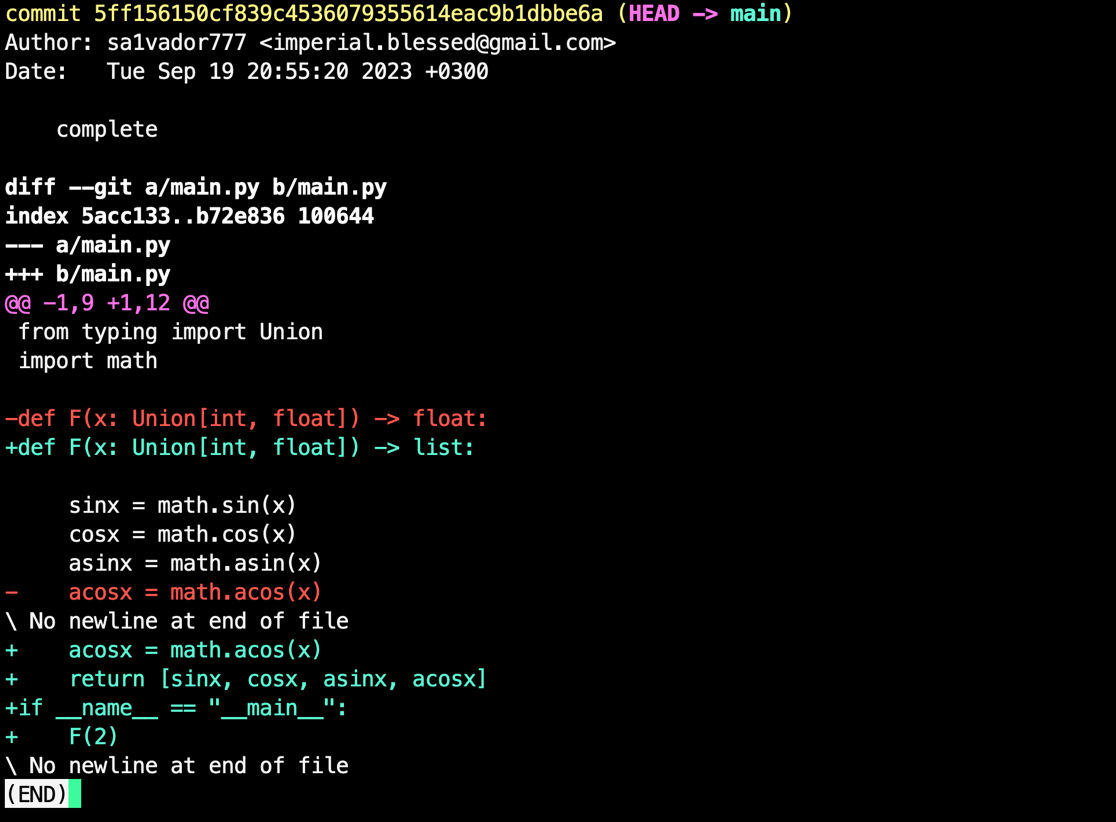


Рисунок 26 – Содержимое коммита HEAD

**Ответы на контрольные вопросы:**

1. Как выполнить историю коммитов в Git? Какие существуют дополнительные опции для просмотра истории коммитов?

Для просмотра истории коммитов используется команда git log. Существуют такие дополнительные опции как: -p/--patch (разница в файлах, внесенная в каждый коммит), -*n* (*n* – число, вывод последних *n* коммитов), --stat (сокращенная статистика для каждого коммита), --pretty (изменение формата вывода) и т. д.

1. Как ограничить вывод при просмотре истории коммитов?

Для ограничения вывод при просмотре используются следующие опции: --after/--before (ограничения вывода до определенной даты), --grep (фильтрация по ключевым словам), --author (фильтрация по автору коммита), -- path\_to\_file (показывает коммиты, которые изменили данный файл) и т. д.

1. Как внести изменения в уже сделанный коммит?

Для изменения сделанного коммита необходимо внести и зафиксировать изменения, а после воспользоваться командой git commit --amend

1. Как отменить индексацию файла в Git?

Для отмены индексации файла необходимо воспользоваться командой git reset HEAD <file\_name> или git restore --staged <file\_name>.

1. Как отменить изменения в файле?

Чтобы отменить изменения в файле необходимо воспользоваться командой git checkout <branch\_hash\_code> -- <file\_name>.

1. Что такое удаленный репозиторий Git?

Это версии проекта, сохраненные в интернете или где-либо еще в сети. Пользователь может иметь несколько удаленных репозиториев, каждый из которых может быть доступен для чтения или чтения-записи. Взаимодействие с другими пользователями предполагает управление удаленными репозиториями, а также отправку и получение данных из них.

1. Как выполнить просмотр удаленных репозиториев данного локального репозитория?

Для того, чтобы просмотреть список настроенных удаленных репозиториев можно запустить команду git remote. Origin – имя по умолчанию, которое Git даёт серверу, с которого производилось клонирование.

1. Как добавить удаленный репозиторий для данного локального репозитория?

Для добавления репозитория необходимо воспользоваться командой git remote add <shortname> <url>.

1. Как выполнить отправку/получение изменений с удаленного репозитория?

Для получения изменений с удаленного репозитория необходимо воспользоваться командой git fetch <remote\_name>. Если ветка настроена на отслеживание удаленной ветки, то можно использовать команду git pull чтобы автоматически получить изменения из удаленной ветки и слить их со своей текущей.

Для того, чтобы поделиться собственными наработками с удаленным репозиторием необходимо воспользоваться командой git push <remote\_name> <branch name>.

1. Как выполнить просмотр удаленного репозитория?

Чтобы просмотреть удаленный репозиторий необходимо воспользоваться командой git shot <remote\_name>.

1. Каково назначение тегов в Git?

Теги помогают отмечать наиболее важные моменты в «жизни» проекта. Обычно с их помощью отмечают моменты, когда проект меняет свою версию.

1. Как осуществляется работа с тегами Git?

Чтобы просмотреть список созданных тегов, нужно воспользоваться командой git tag. Чтобы создать аннотированный тег используется команда git tag -a <tag\_name> -m <comment>. Для отправки на удаленный репозиторий используется команда git push <remote\_name> <tag\_name> или git push <remote\_name> --tags, чтобы отправить все теги. Для удаления тега используется команда git tag -d <tag\_name>. Удаление с удаленного репозитория – git push <remote\_name> --delete <tagname>. Переход на тег осуществляется командой git checkout <tag\_name>.

1. Самостоятельно изучите назначение флага --prune в командах git fetch и git push. Каково назначение этого флага?

В команде git push флаг --prune удаляет удаленные ссылки на теги или ветки, которые не существуют на локальном репозитории. В команде git fetch --prune удаляет ссылки на теги или ветки, которые больше не существуют в удаленном репозитории.