|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| *Федеральное государственное бюджетное образовательное учреждение высшего образования* ***«МИРЭА – Российский технологический университет»***  **РТУ МИРЭА** |

Институт Информационных технологий (ИТ)

Кафедра Математического обеспечения и стандартизации информационных технологий (МОСИТ)

|  |
| --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №1** |
| **по дисциплине** |
| **«Технология разработки программных приложений»**  **Тема: «Системы контроля версий»** |

|  |  |
| --- | --- |
| Выполнил студент группы ИКБО-08-21 | Гусев Т. А |
| Принял | Туманова М. Б |

|  |  |  |
| --- | --- | --- |
| Практическая работа выполнена | «\_\_»\_\_\_\_\_\_\_2023г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2023 г. | *(подпись руководителя)* |

Москва 2023

[**1.** **Основные команды git** 3](#_Toc127459537)

[**2.** **Системы управления репозиториями.** 7](#_Toc127459538)

[**3.** **Работа с ветвлением и оформлением кода** 12](#_Toc127459539)

[**4.** **Контрольные вопросы** 16](#_Toc127459540)



# **Основные команды git**

Установите и настройте клиент git на своей рабочей станции.  
и создайте локальный репозиторий и добавьте в него несколько файлов.

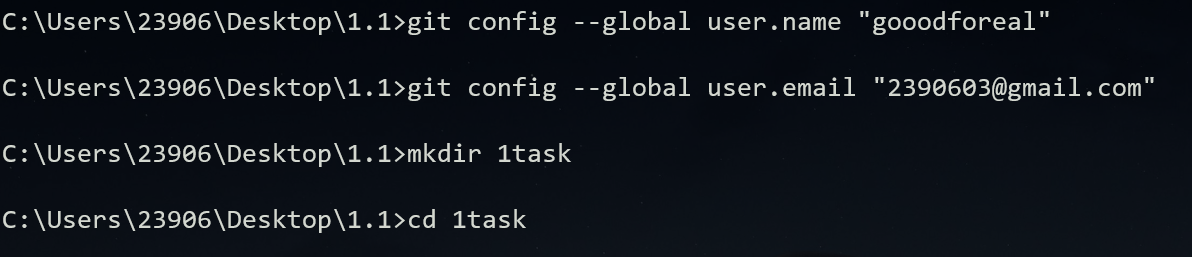


Рисунок 1 – Установка git и создание локального репозитория

Создайте несколько файлов и внесите в них изменения

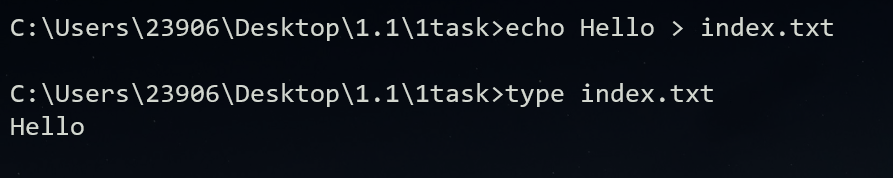


Рисунок 2 – Создание файла

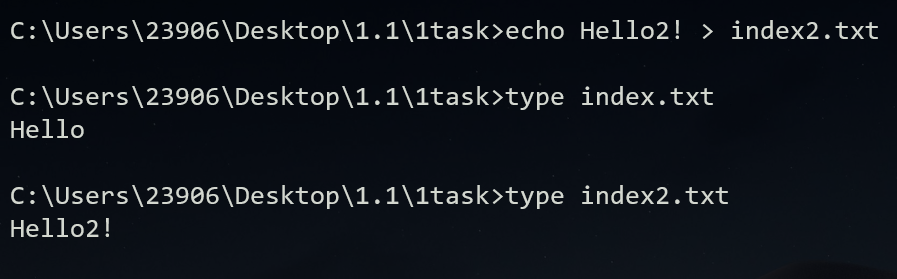


Рисунок 3 – Создание второго файла

Проиндексируйте изменения и проверьте состояние. Сделайте коммит того, что было проиндексировано в репозиторий. Добавьте к  
коммиту комментарий. Измените еще один файл. Добавьте это изменение в индекс git. Измените файл еще раз. Проверьте состояние и произведите коммит проиндексированного изменения. Теперь добавьте второе изменение в индекс, а затем проверьте состояние с помощью команды git status. Сделайте коммит второго изменения.



Рисунок 4 – Индексация изменений и коммит

Просмотрите историю коммитов с помощью команды git log. Ознакомьтесь с параметрами команды и используйте некоторые из них для различного формата отображения истории коммитов.

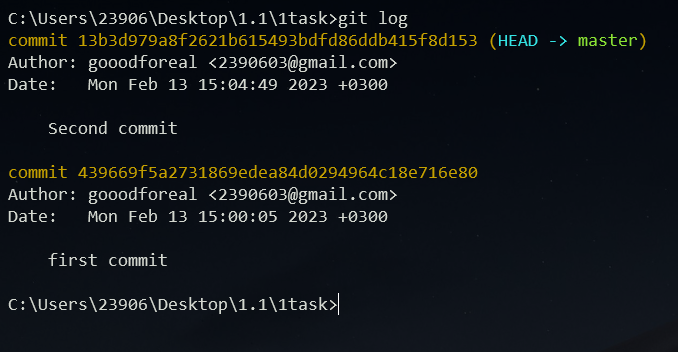


Рисунок 5 – История коммитов

Верните рабочий каталог к одному из предыдущих состояний.

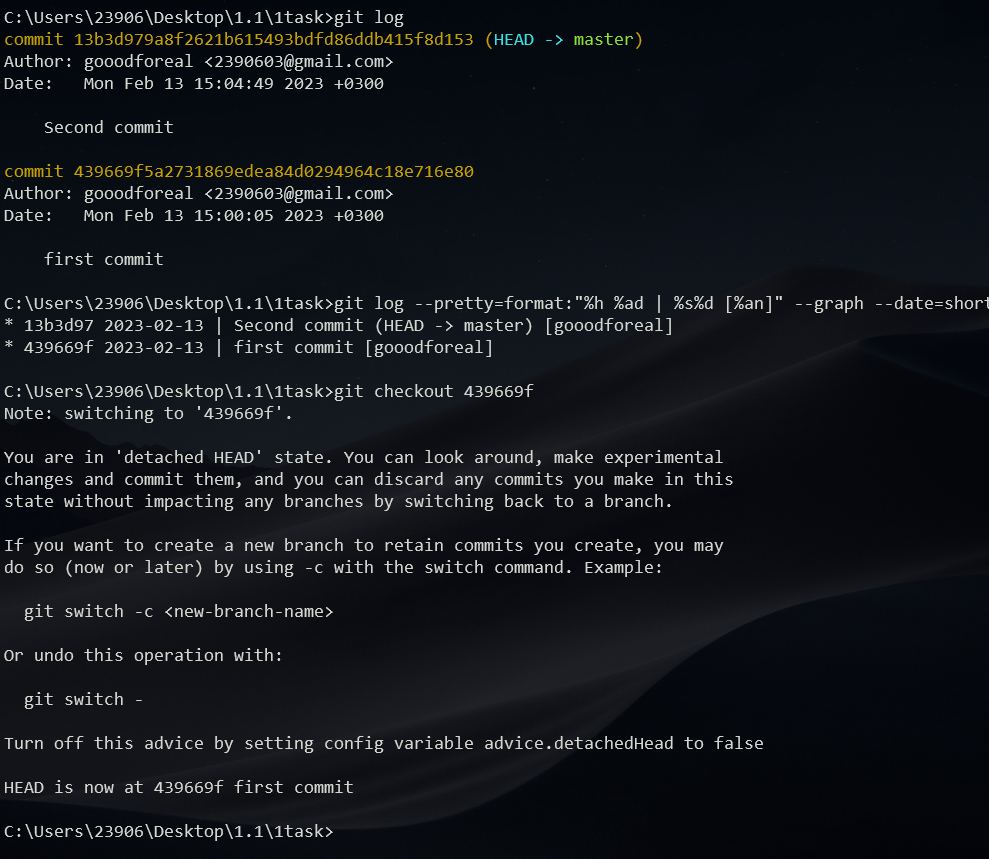


Рисунок 6 – Возвращение к предидущему состоянию

Отмените некоторые изменения в рабочем каталоге (до и после  
индексирования).

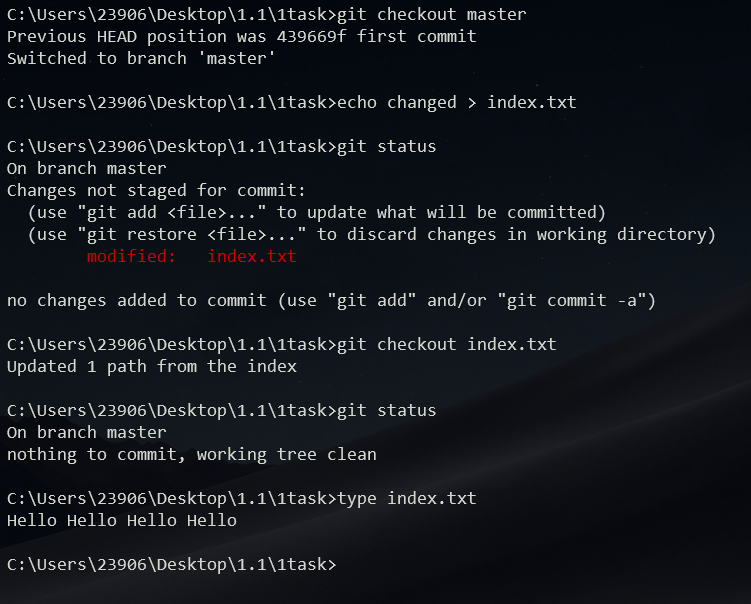


Рисунок 7 – Отмена изменений

Отмените один из коммитов в локальном репозитории.

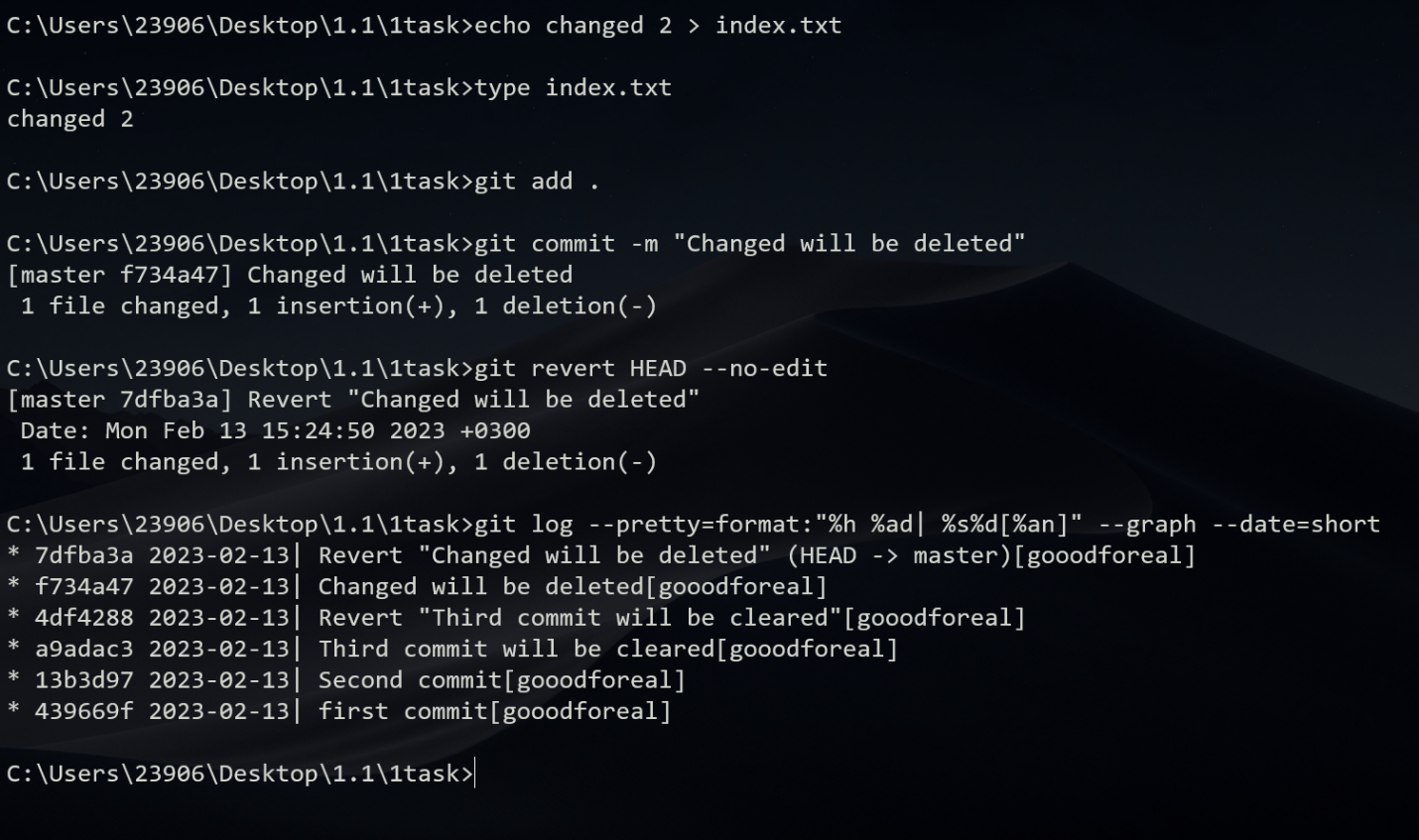


Рисунок 8 – Отмена коммита

# **Системы управления репозиториями.**

Создайте аккаунт и репозиторий на GitHub.

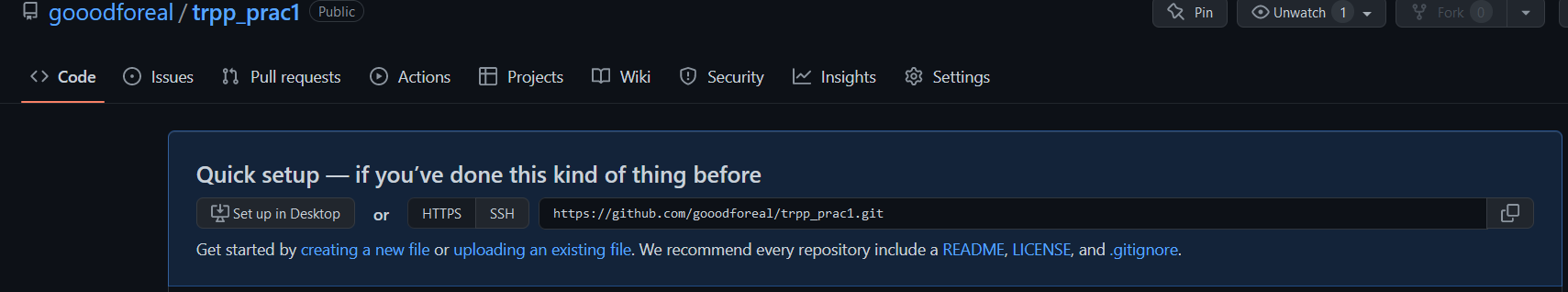


Рисунок 9 – Создание репозитория на GitHub

Создайте новый локальный репозиторий с несколькими файлами на рабочей станции и загрузите его содержимое на GitHub.

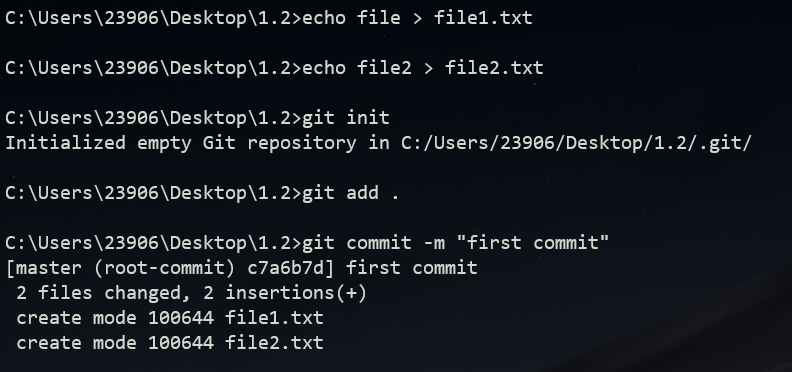


Рисунок 10 – Создание репозитория и файлов

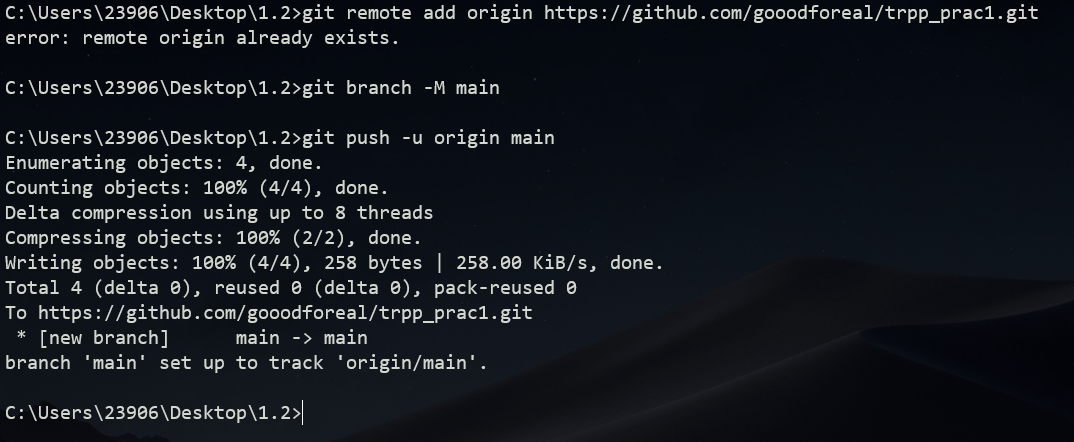


Рисунок 11 – Выгрузка репозитория

Создайте в репозитории новую ветку. Произведите в ней несколько изменений и слейте с веткой master.

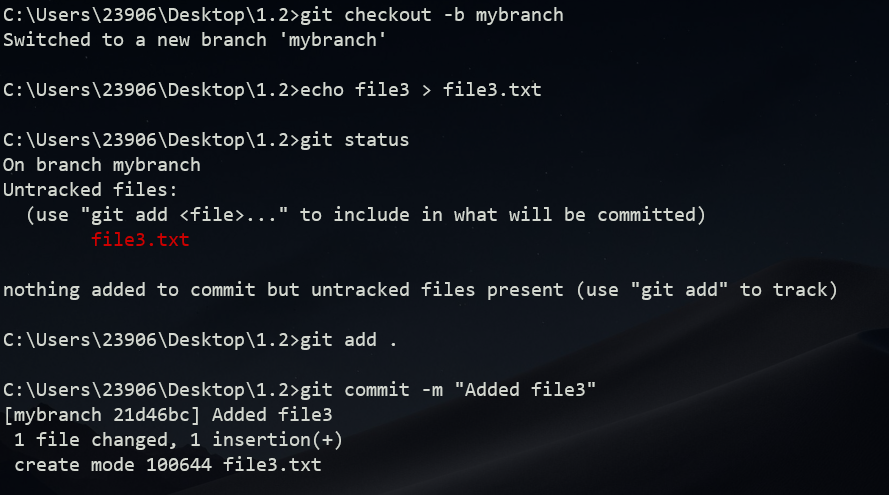


Рисунок 12 – Создание новой ветки

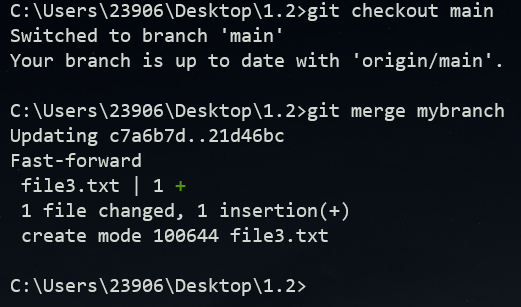


Рисунок 13 – Слияние веток

Выполните цепочку действий в репозитории, согласно варианту. *Вариант 7*

Клонируйте непустой удаленный репозиторий на локальную машину  
Создайте две новых ветки и выведите список всех веток.

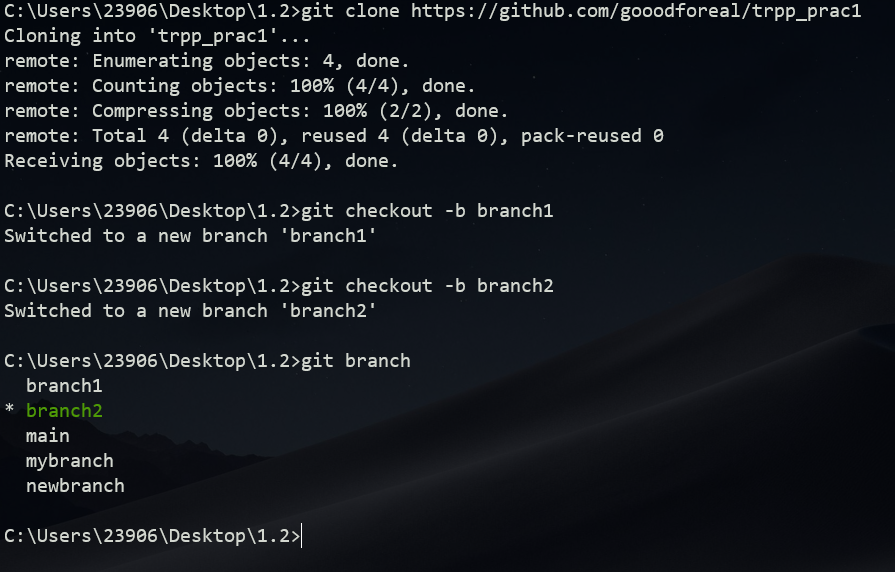


Рисунок 14 – Клонирование репозитория и создание веток

Проведите три коммита в первой ветке

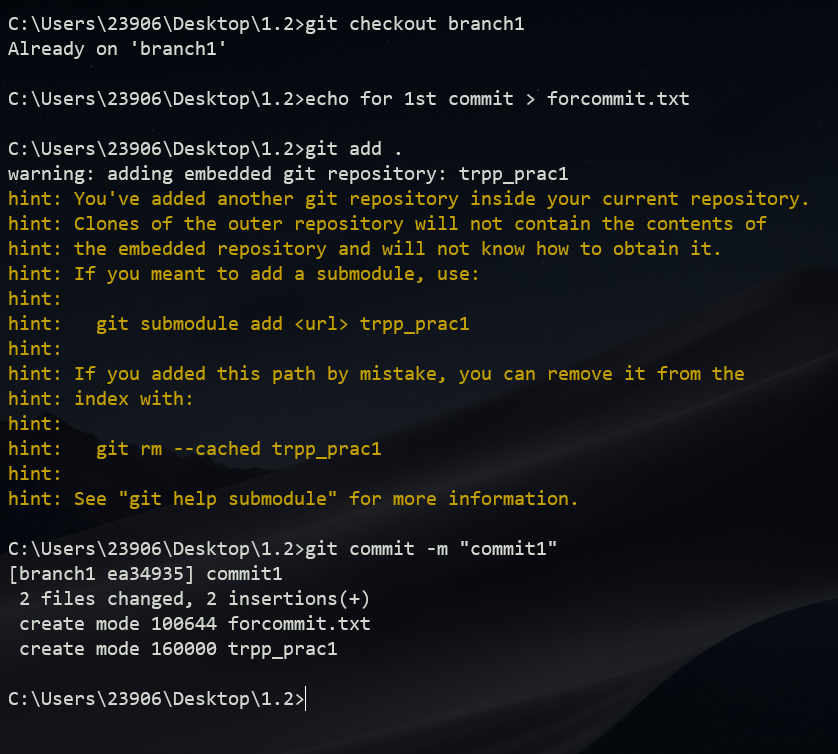


Рисунок 15 – Первый коммит в ветке

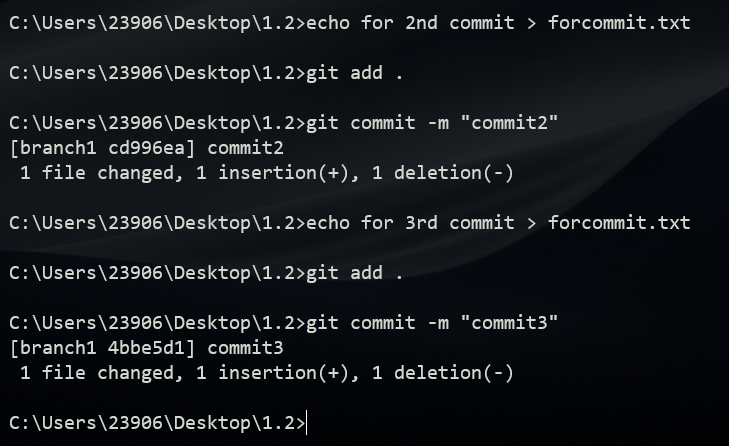


Рисунок 16 – Коммиты в первой ветке

Переместите (cherry pick) первый коммит во вторую ветку. Выгрузите изменения в удаленный репозиторий.

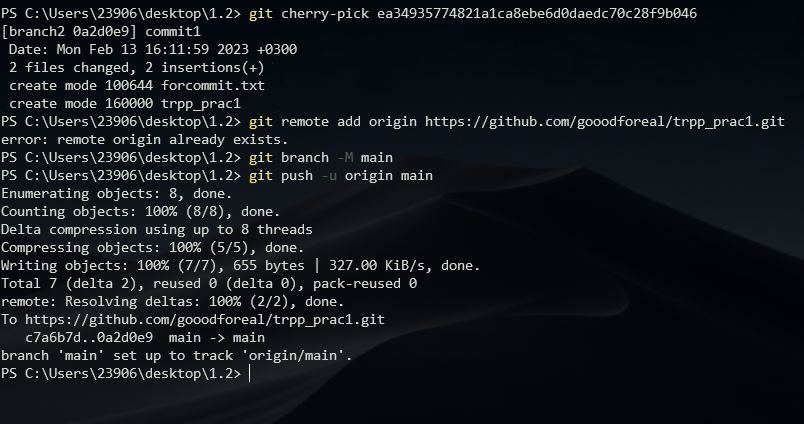


Рисунок 17 – Cherry pick и выгрузка

Выведите в консоли различия между веткой master и новыми ветками

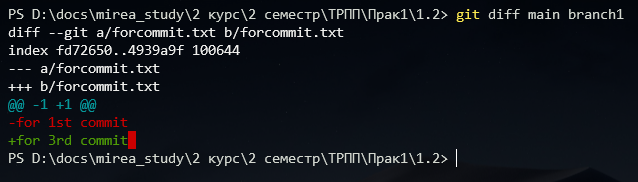


Рисунок 18 – Различия между main и branch1

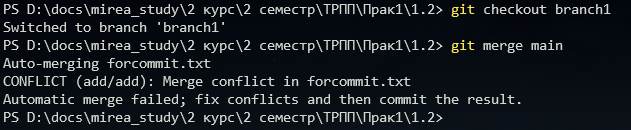


Рисунок 19 – Слияние main и branch1

# **Работа с ветвлением и оформлением кода**

*Вариант 7*. (https://github.com/kettanaito/naming-cheatsheet)

Сделайте форк репозитория в соответствии с вашим вариантом.

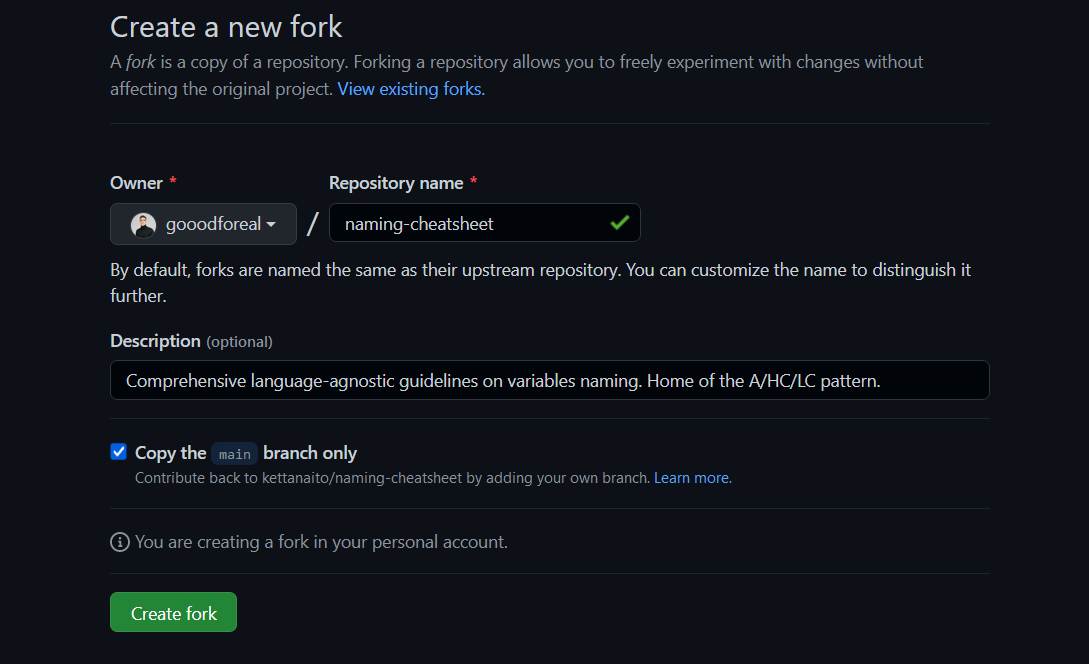


Рисунок 20 – Форк репозитория

Склонируйте его на локальную машину

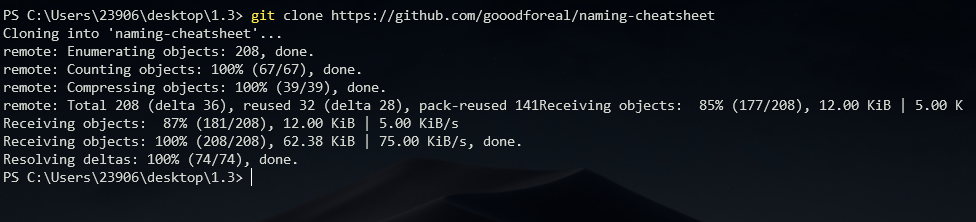


Рисунок 21 – Клонирование репозитория

Создайте две ветки branch1 и branch2 от последнего коммита в master'е

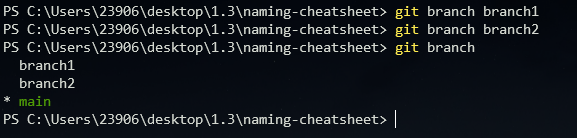


Рисунок 22 – Создание веток

Проведите по 3 коммита в каждую из веток, которые меняют один и тот же кусочек файла

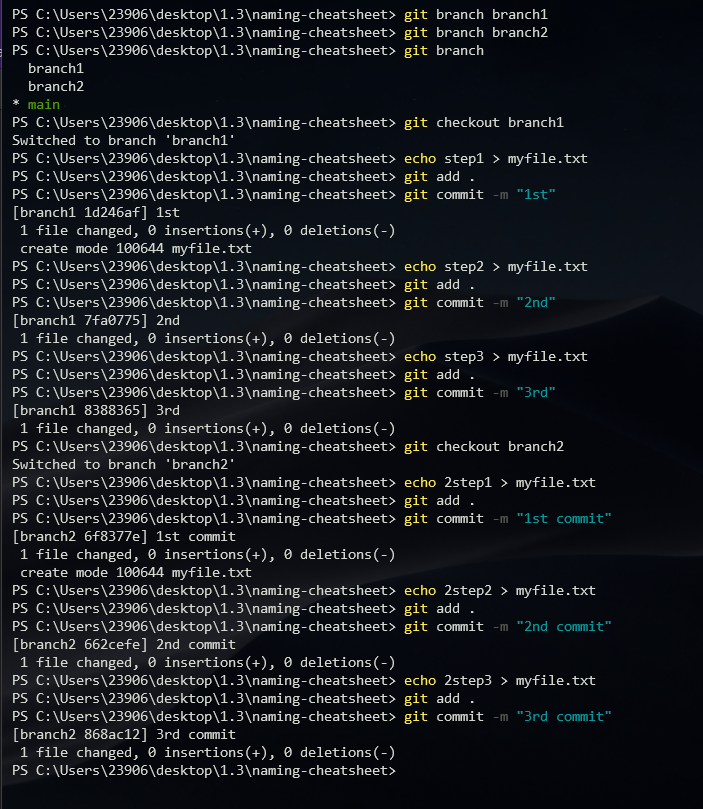


Рисунок 23 – Проведение коммитов

Выполните слияние ветки branch1 в ветку branch2, разрешив конфликты при этом

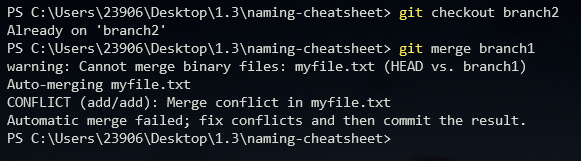


Рисунок 24 – Слияние веток

Выгрузите все изменения во всех ветках в удаленный репозиторий



Рисунок 25 – Выгрузка на GitHub

Проведите еще 3 коммита в ветку branch1

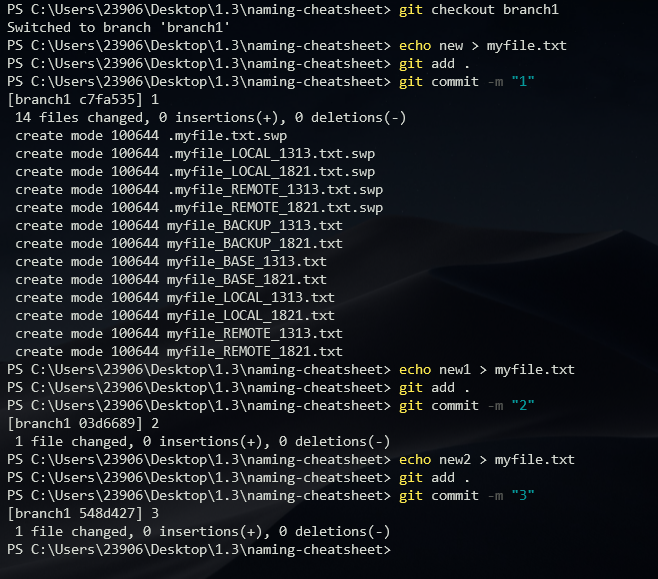


Рисунок 26 – Проведение коммитов

Склонируйте репозиторий еще раз в другую директорию

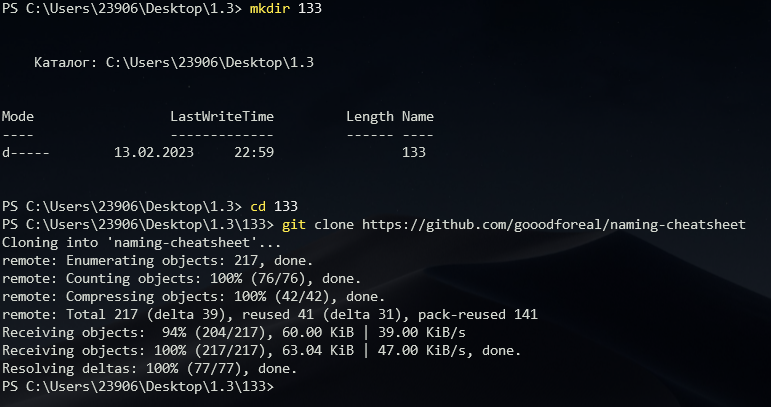


Рисунок 27 – Клонирование репозитория в другую директорию

В новом клоне репозитории сделайте 3 коммита в ветку branch1

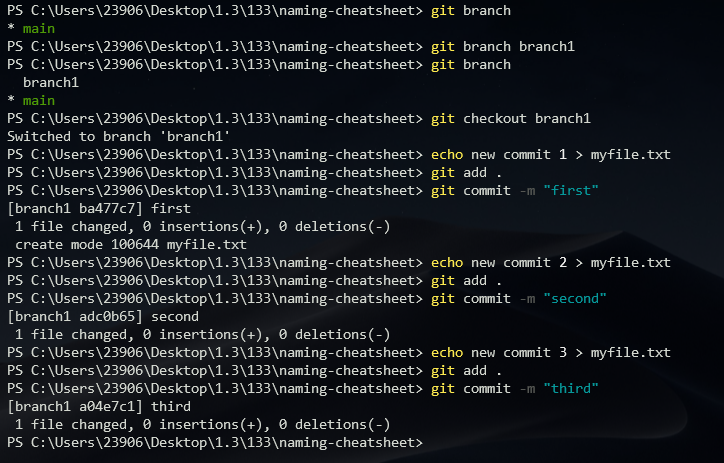


Рисунок 28 – Проведение коммитов

Выгрузите все изменения из нового репозитория в удаленный репозиторий

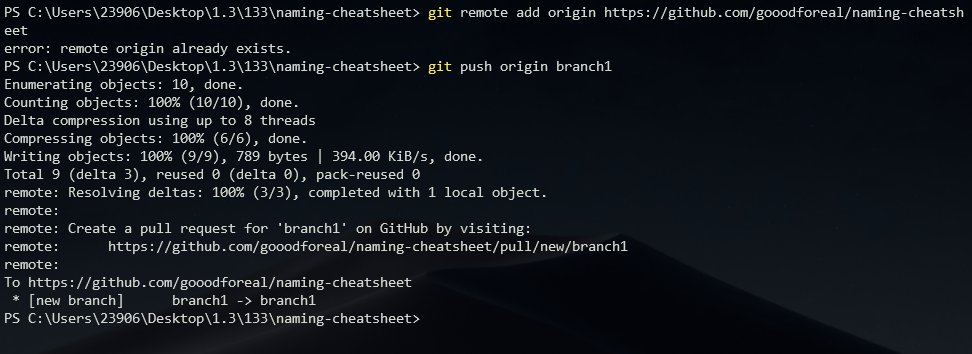


Рисунок 29 – Выгрузка изменений

Вернитесь в старый клон с репозиторием, выгрузите изменения с опцией –force

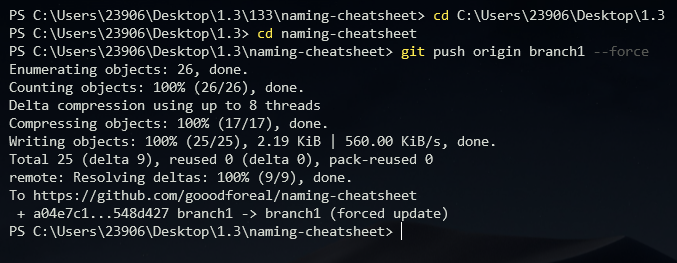


Рисунок 30 – Выгрузка старого клона с опцией -force

# **Контрольные вопросы**

1. Что такое GitHub?

GitHub — крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Веб-сервис основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub.

1. Что такое коммит?

Коммиты — основные конструктивные элементы временной шкалы проекта Git. Их можно рассматривать как снимки состояния или контрольные точки на временной шкале проекта Git. Коммиты создаются с помощью команды git commit, которая делает снимок состояния проекта на текущий момент времени.

1. Что делает команда git diff?

Git diff. Команда git diff используется для вычисления разницы между любыми двумя Git деревьями. Это может быть разница между вашей рабочей копией и индексом (собственно git diff ), разница между индексом и последним коммитом ( git diff --staged ), или между любыми двумя коммитами ( git diff master branchB ).

1. Что делает команда git status?

Команда git status отображает состояние рабочего каталога и раздела проиндексированных файлов. С ее помощью можно проверить индексацию изменений и увидеть файлы, которые не отслеживаются Git. Информация об истории коммитов проекта не отображается при выводе данных о состоянии.

1. Что делает команда git add?

Команда git add добавляет изменение из рабочего каталога в раздел проиндексированных файлов. Она сообщает Git, что вы хотите включить изменения в конкретном файле в следующий коммит.

1. Что делает команда git log?

Одним из основных и наиболее мощных инструментов для этого является команда git log . По умолчанию (без аргументов) git log перечисляет коммиты, сделанные в репозитории в обратном к хронологическому порядке — последние коммиты находятся вверху.

1. Что такое форк репозитория?

Fork - это копия репозитория. С помощью него создается точная копия оригинального репозитория, только на сервисе GitHub. В копии репозитория можно вносить свои собственные изменения, редактировать файлы или удалять директории.

1. Как добавить новую директорию в Git?

Добавить пустую директорию нельзя. Команда git add «видит» только файлы и пути, в которых они лежат. Чтобы сохранить пустую директорию, создайте в ней пустой файл . gitkeep и добавьте его в git.

# **Вывод**

В ходе практической работы были освоены базовые навыки работы с git

и его командами. Также был создан GitHub. По окончании выполнения задания был освоен навык работы с удаленным репозиторием.