**ОГЛАВЛЕНИЕ**

[Часть 1. Основные команды git 2](#_Toc127385455)

[Часть 2. Системы управления репозиториями 8](#_Toc127385456)

[Часть 3. Работа с ветвлением и оформление кода 15](#_Toc127385457)

[Ответы на вопросы 21](#_Toc127385458)

# Часть 1. Основные команды git

1. Установите и настройте клиент git на своей рабочей станции

На рисунке 1 показан процесс настройки конфигурации git.

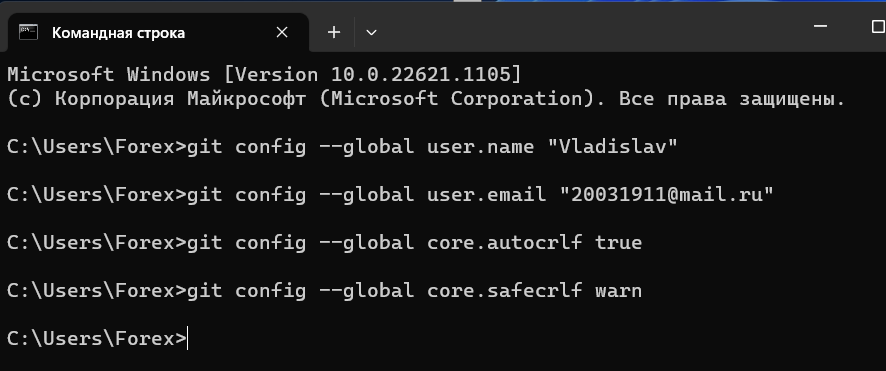


Рисунок 1 – Настройка конфигурации git

1. Создайте локальный репозиторий и добавьте в него несколько файлов.

На рисунке 2 показан процесс инициализации пустого репозитория и добавления файлов в папку проекта.

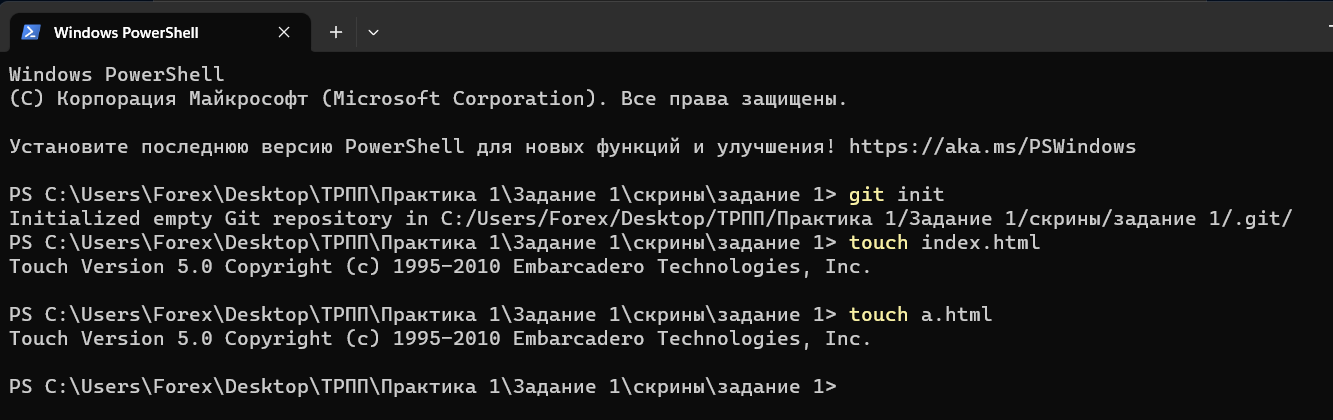


Рисунок 2 – Инициализация репозитория git

1. Внесите изменения в один из файлов.
2. Проиндексируйте изменения и проверьте состояние.
3. Сделайте коммит того, что было проиндексировано в репозиторий. Добавьте к коммиту комментарий.

На рисунке 3 показан процесс изменения файла index.html, индексация изменений и создание коммита.

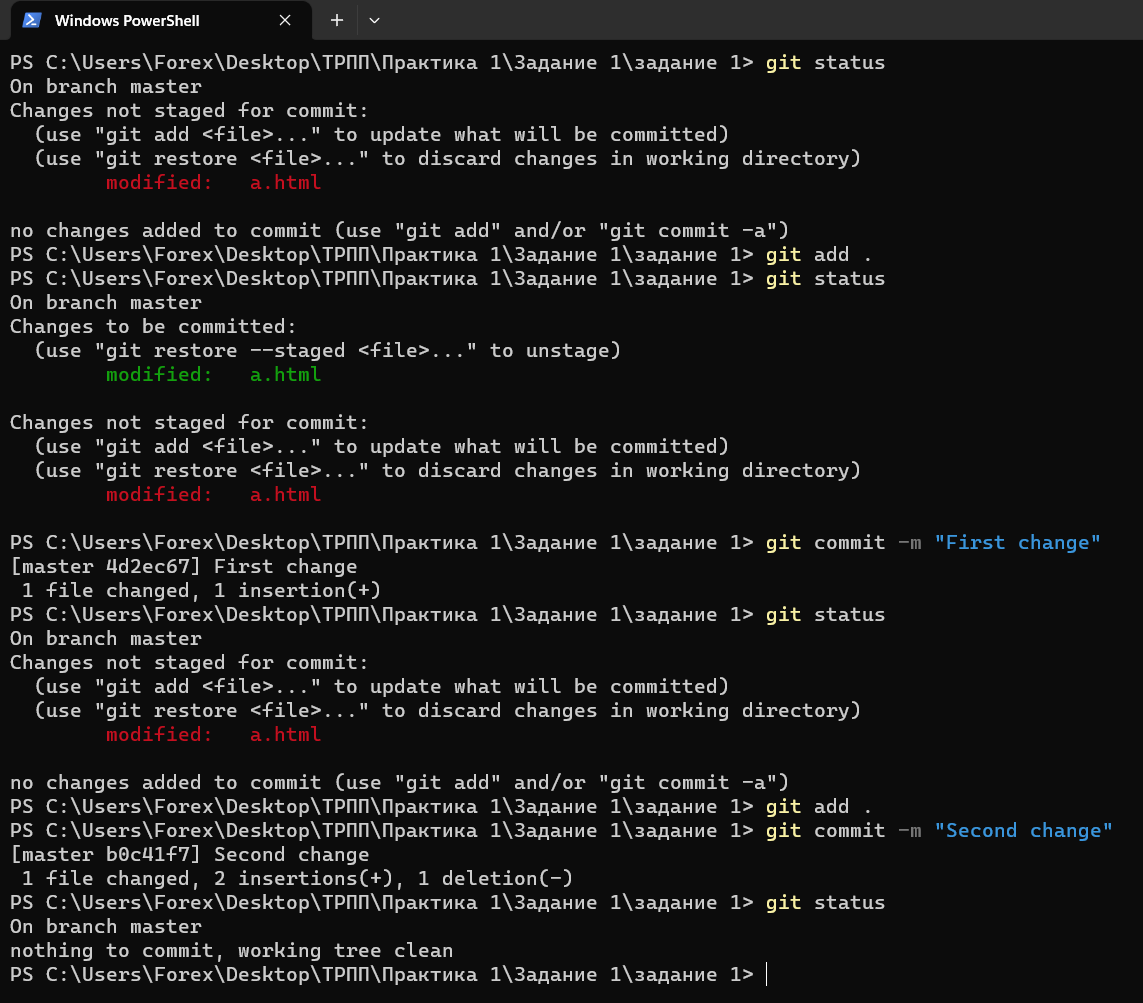


Рисунок 3 – изменение файла и создание коммита

1. 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.

На рисунке 4 показан процесс создания коммитов с разными изменениями одного файла.

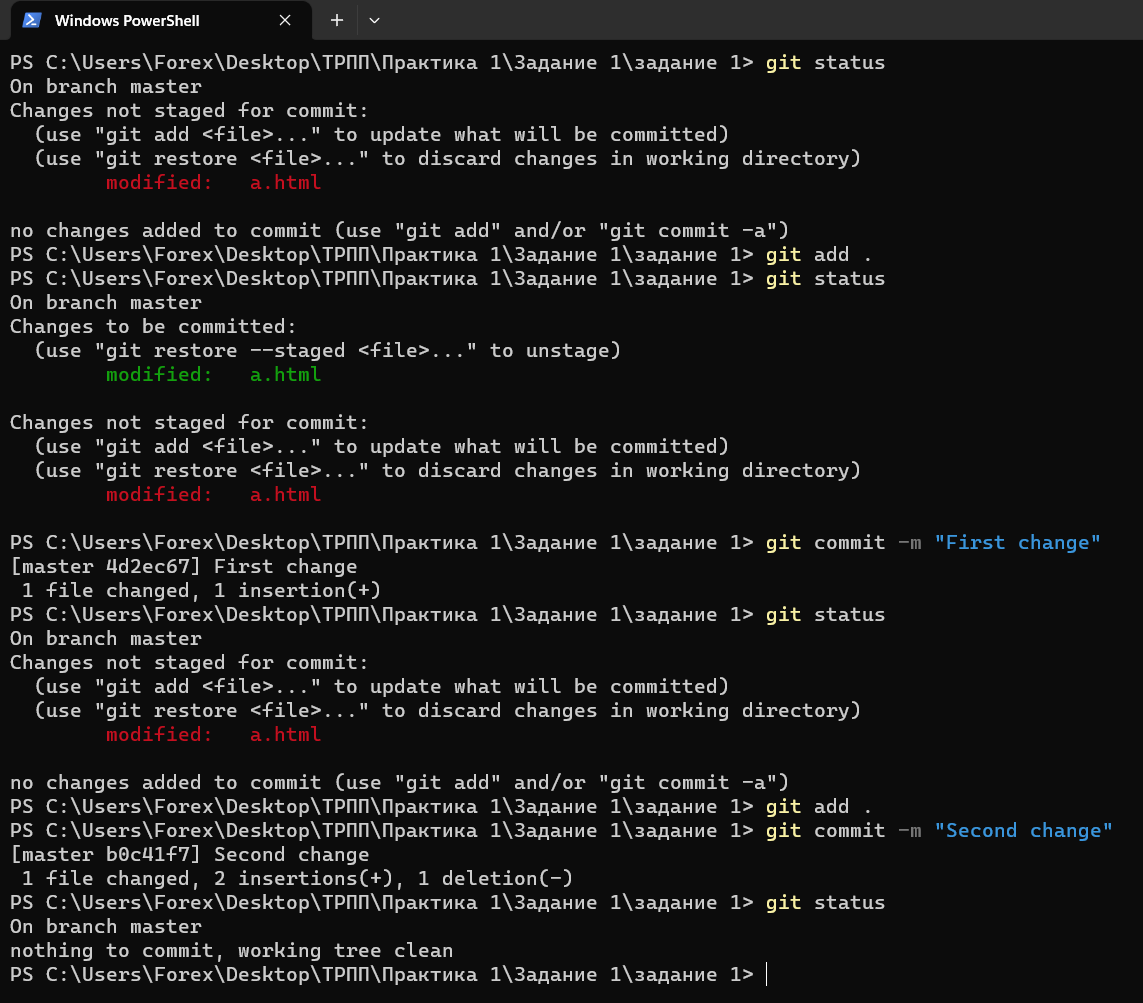


Рисунок 4 – Постепенное изменение файла и создание коммитов

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

На рисунке 5 показана история коммитов репозитория.

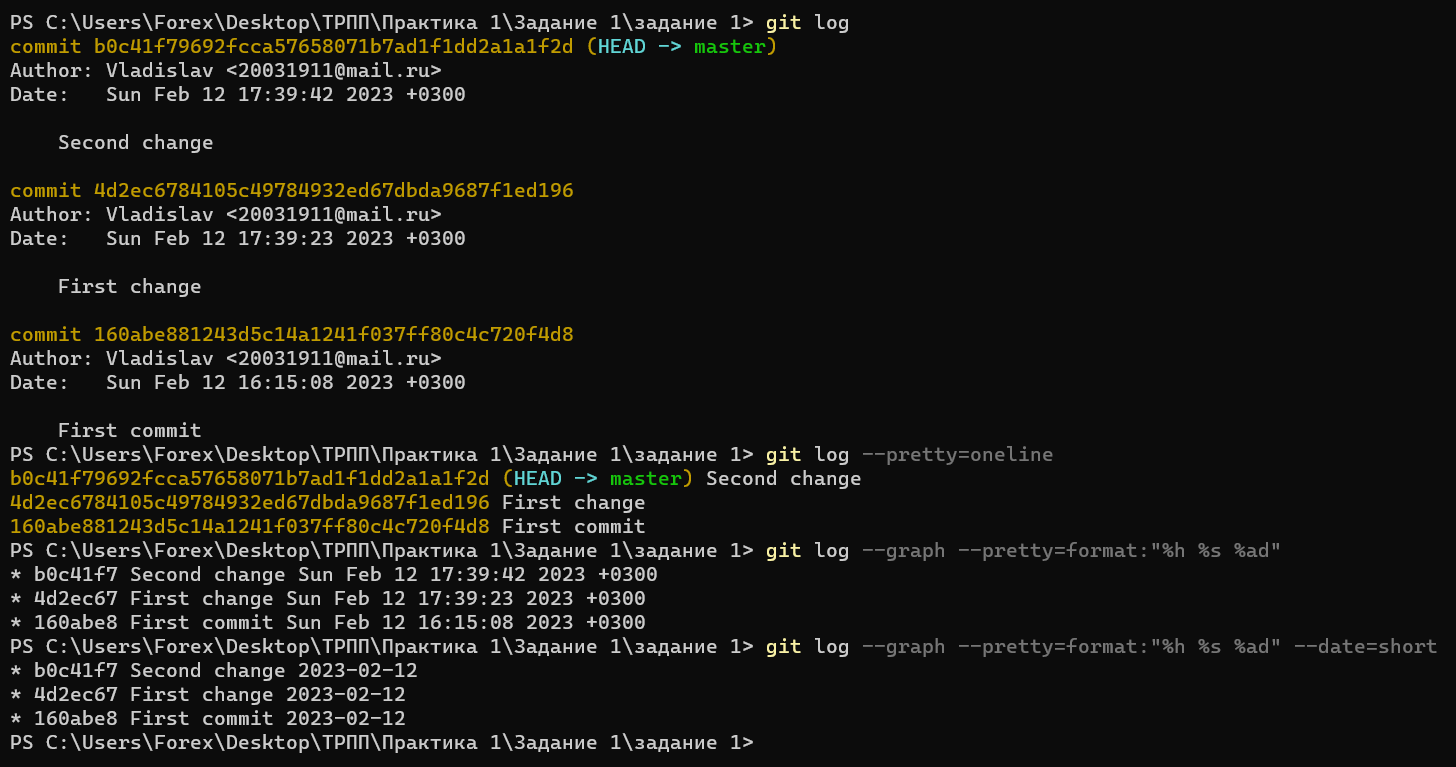


Рисунок 5 – История коммитов

1. Верните рабочий каталог к одному из предыдущих состояний.

На рисунке 6 показан процесс возврата репозитория к одному из предыдущих состояний.

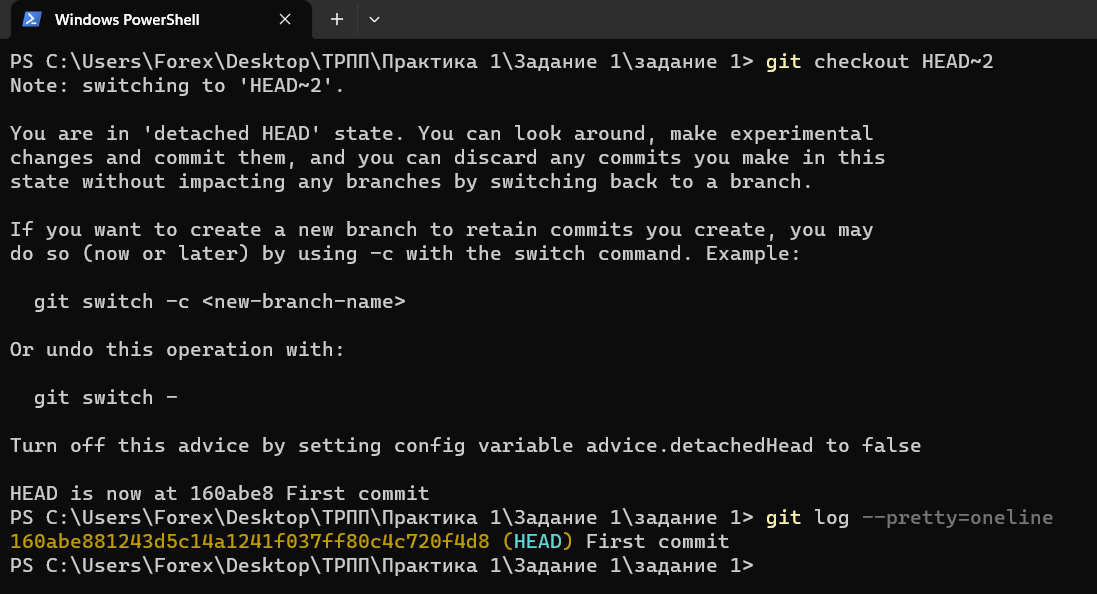


Рисунок 6 – Возврат репозитория к предыдущему состоянию

1. Изучите, как создавать теги для коммитов для использования в будущем.

На рисунке 7 показан процесс создания тегов для коммита.

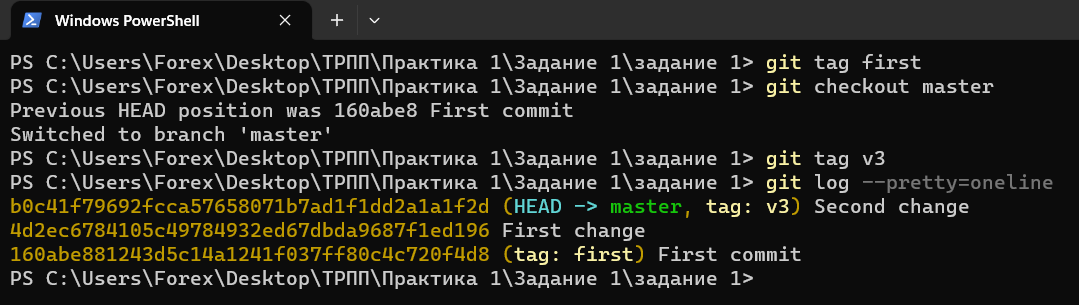


Рисунок 7 – Создание тегов для коммита

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

На рисунке 8 показан процесс отмены изменений до индексации.

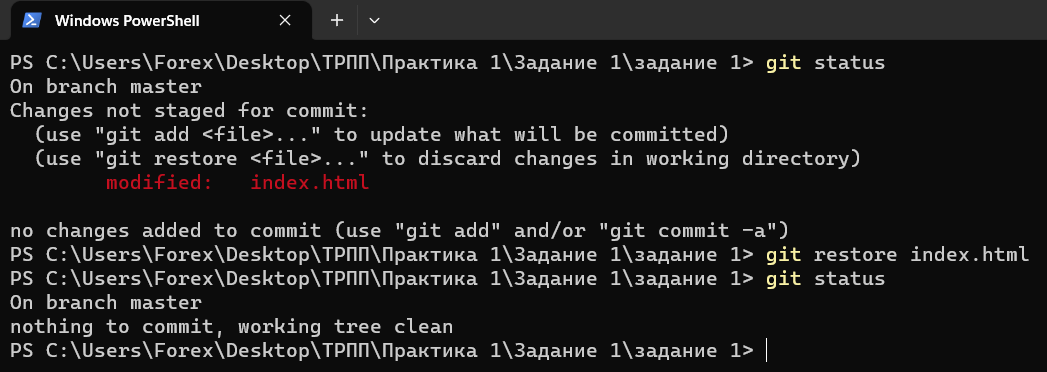


Рисунок 8 – Отмена изменений до индексации

На рисунке 9 показан процесс отмены изменений после индексации.

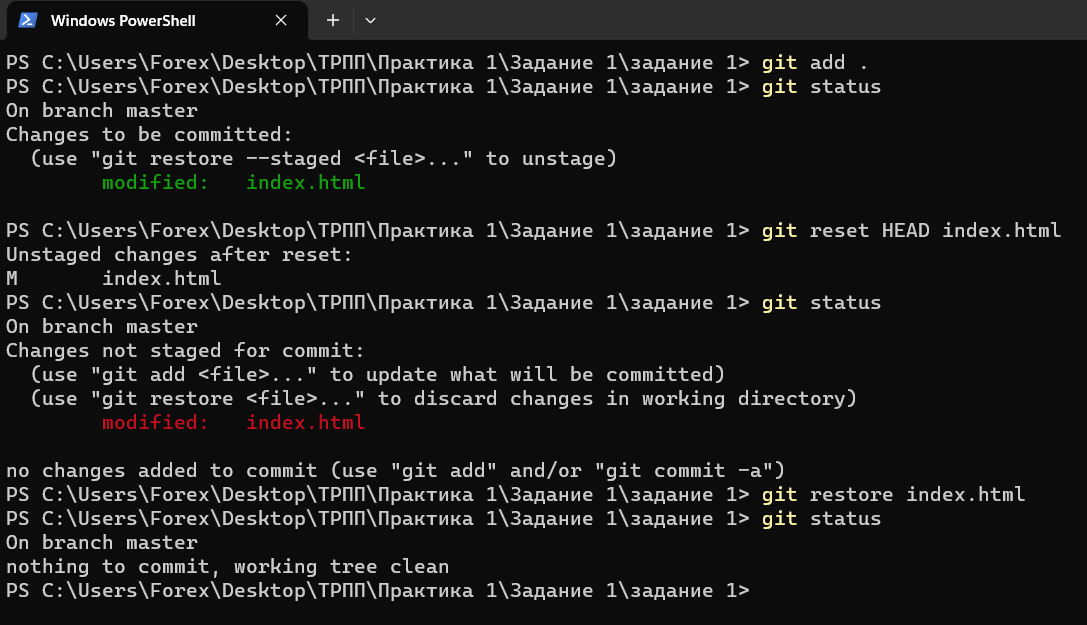


Рисунок 9 – отмена проиндексированных изменений

1. Отмените один из коммитов в локальном репозитории.

На рисунке 10 показан процесс удаления последнего коммита.

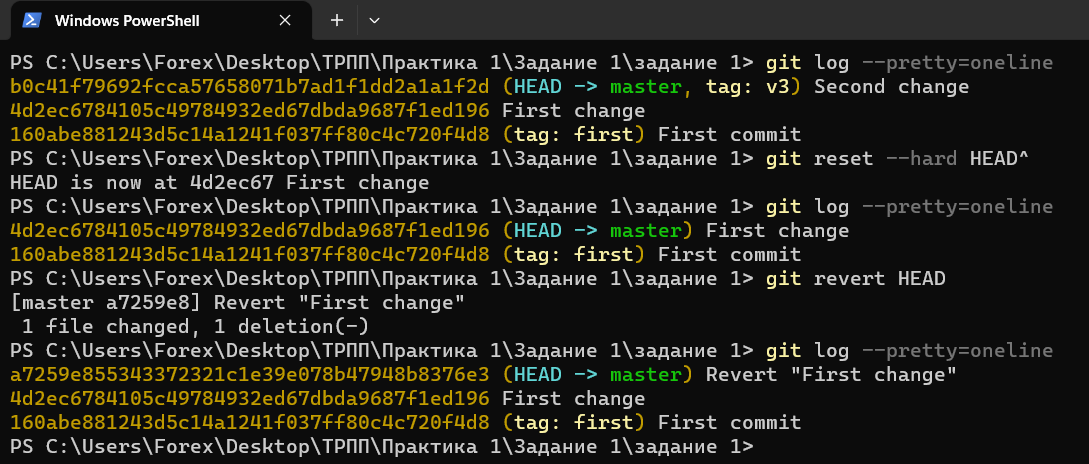


Рисунок 10 – Удаление последнего коммита

# Часть 2. Системы управления репозиториями

**Вариант 6**

1. **Создайте аккаунт на GitHub**

На рисунке 11 показан созданный аккаунт на GitHub.

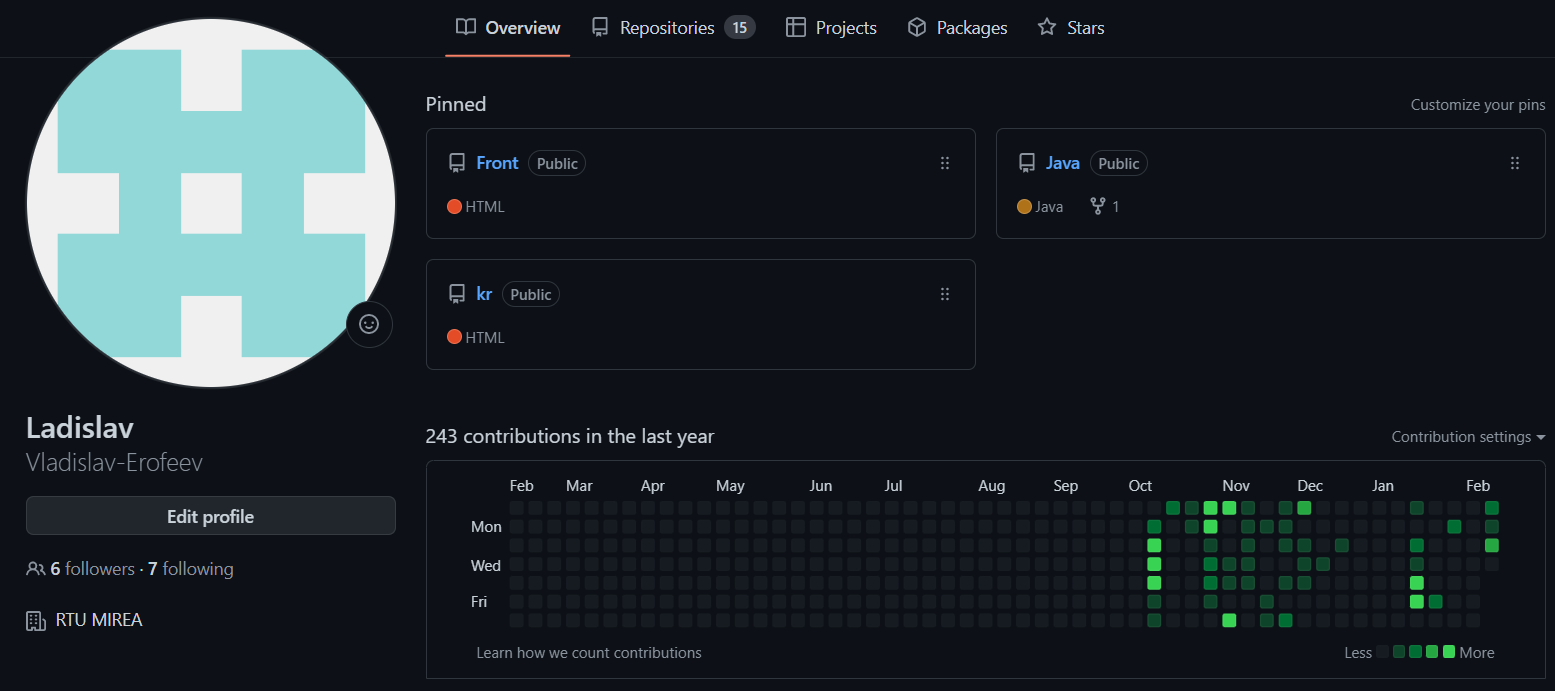


Рисунок 11 – Аккаунт GitHub

1. **Создайте репозиторий согласно варианту**

На рисунке 12 показан процесс создания удалённого репозитория.

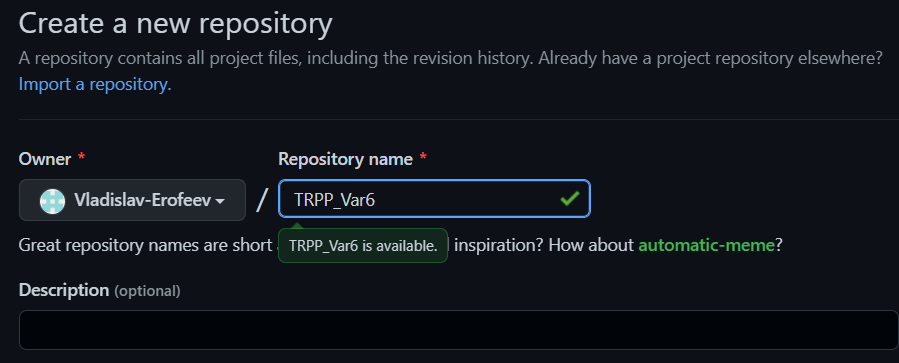


Рисунок 12 – Создание удалённого репозитория

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

На рисунке 13 показан процесс загрузки существующего репозитория на GitHub.

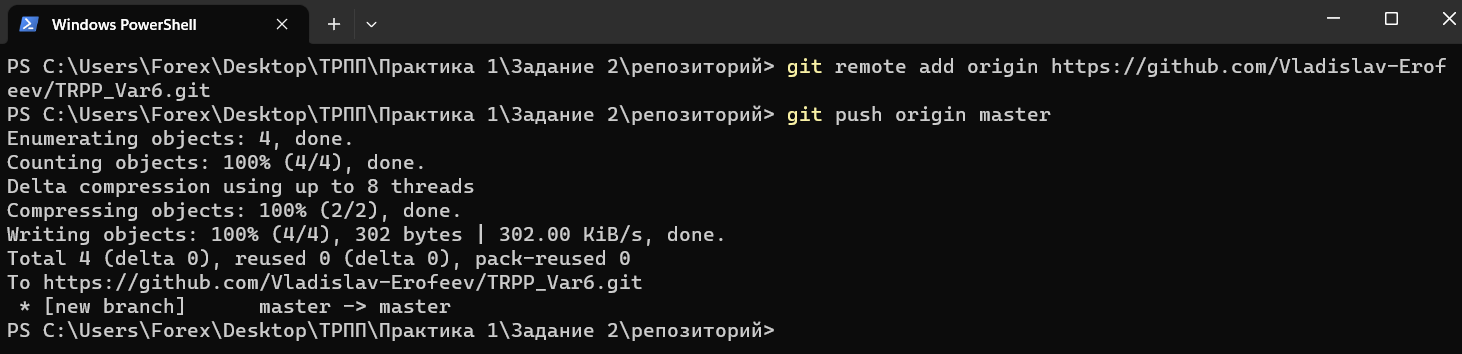


Рисунок 13 – Загрузка репозитория на GitHub

1. **Чтобы избежать ввода логина и пароля, создайте SSH-ключ для авторизации**

На рисунке 14 показан процесс генерации SSH ключа.

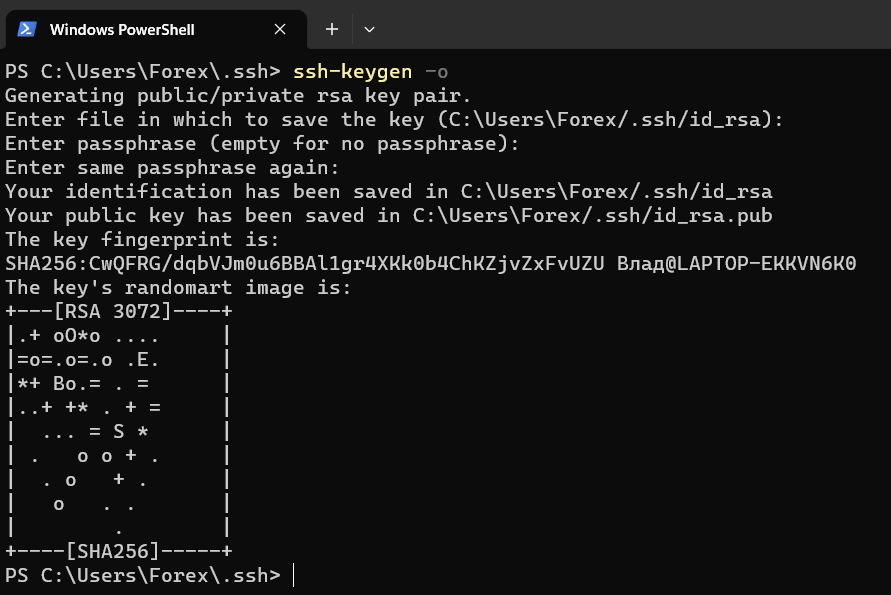


Рисунок 14 – Генерация SSH ключа

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

На рисунке 15 показан процесс создания новой ветки и слияние её с веткой master.

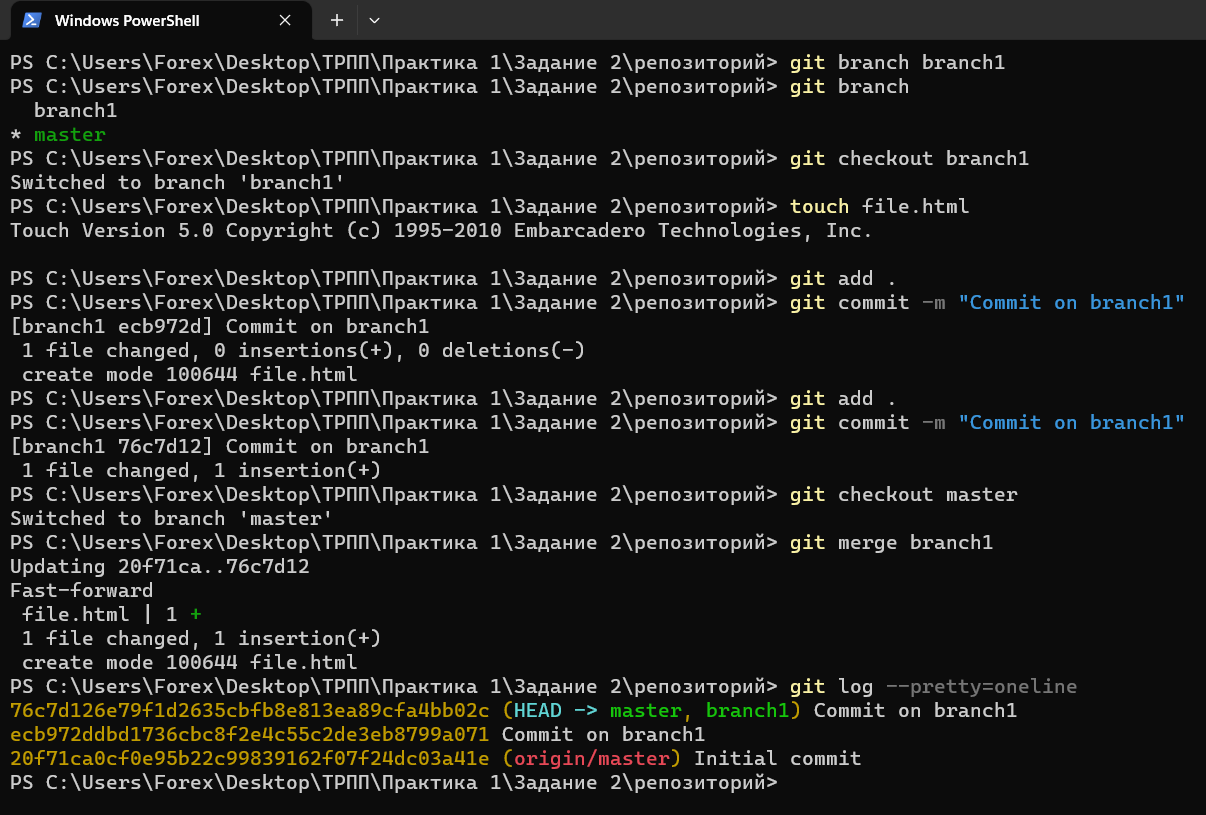


Рисунок 15 – Слияние новой ветки с веткой master

1. **Клонируйте непустой удаленный репозиторий на локальную машину**

На рисунке 16 показан процесс клонирования удалённого репозитория.

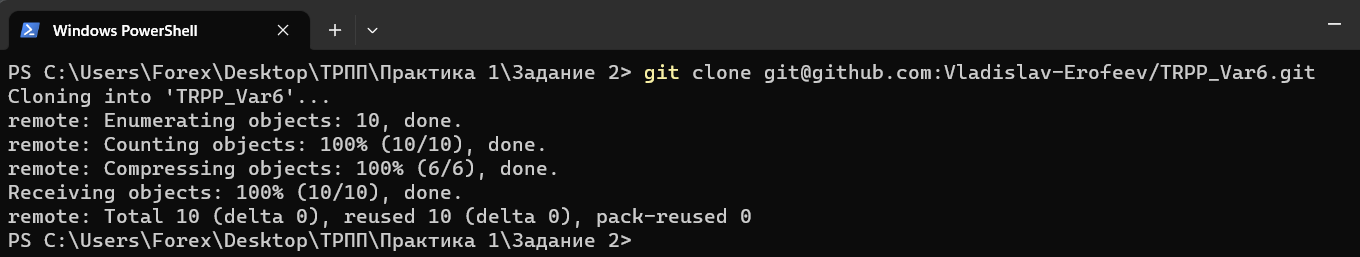


Рисунок 16 – Клонирование репозитория

1. **Создайте новую ветку и выведите список всех веток**

На рисунке 17 показан процесс создания новой ветки и вывод списка всех веток.

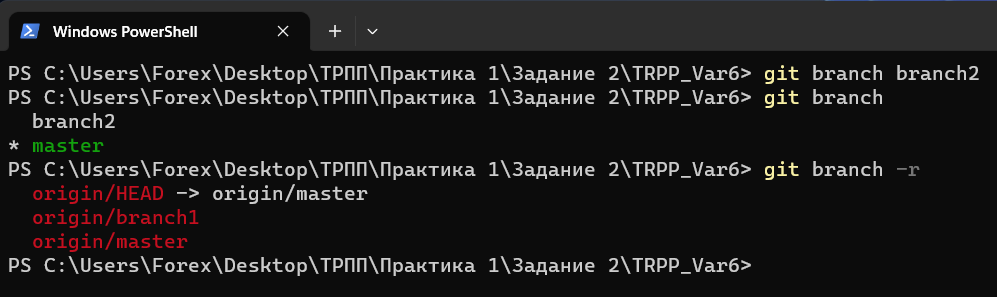


Рисунок 17 – Создание новой ветки и вывод списка веток

1. **Произведите 3 коммита в новой ветке в новых файлах**

На рисунке 18 показан процесс создания трёх коммитов в новой ветке.

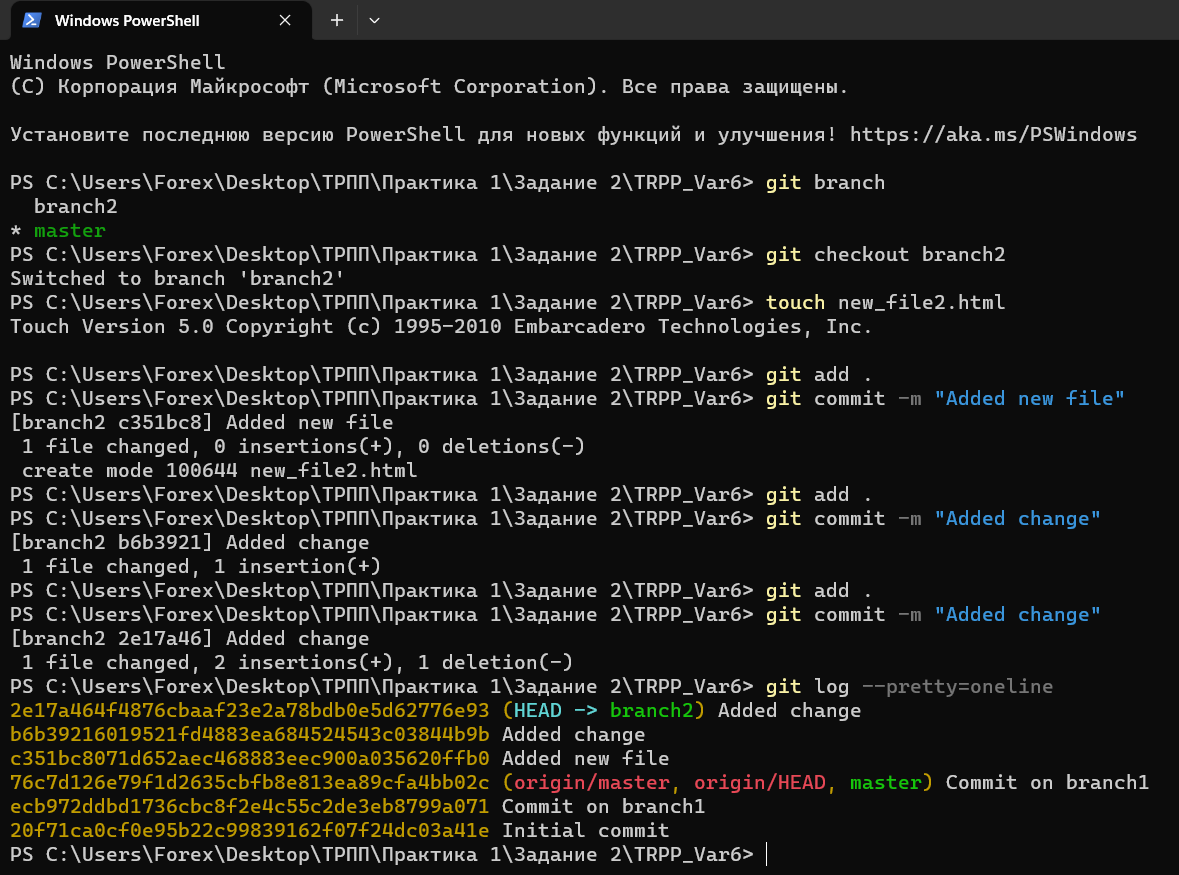


Рисунок 18 – Создание трёх коммитов в новой ветке

1. **Выгрузите изменения в удаленный репозиторий**

На рисунке 19 показан процесс выгрузки изменений в удалённый репозиторий.

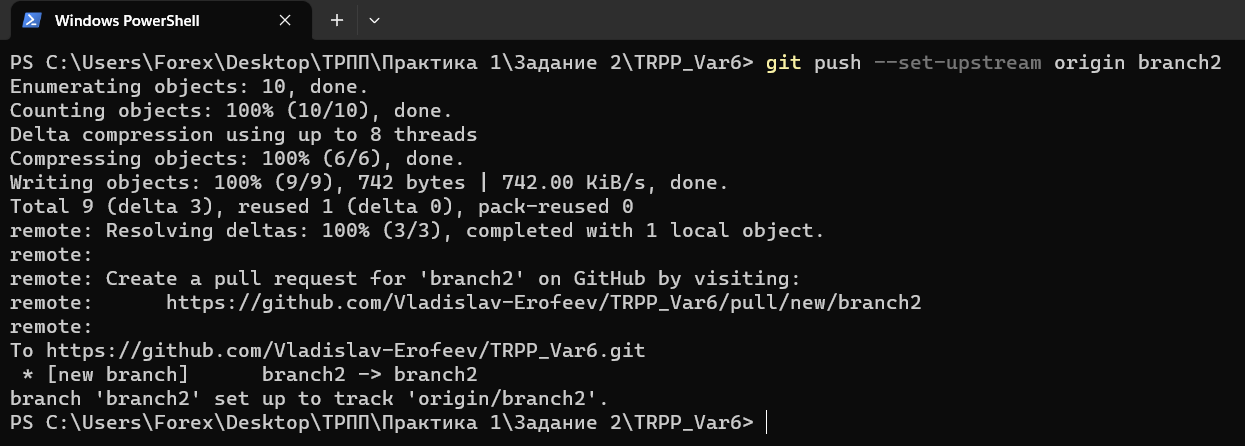


Рисунок 19 – Выгрузка изменений в удалённый репозиторий

1. **Произведите какие-нибудь изменения в файле который существует в ветке master, но не коммитьте их**

На рисунке 20 показано не закоммиченное изменение в файле, который существует в ветке master.

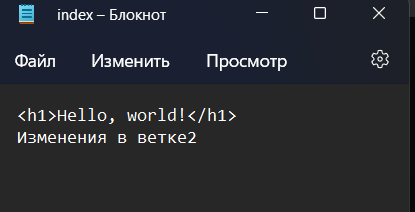


Рисунок 20 – Изменение на ветке branch2

1. **Выполните gis stash, переключитесь на ветку master, выполните gis stahs pop**

На рисунке 21 показан процесс переноса не закоммиченных изменений с branch2 на master.

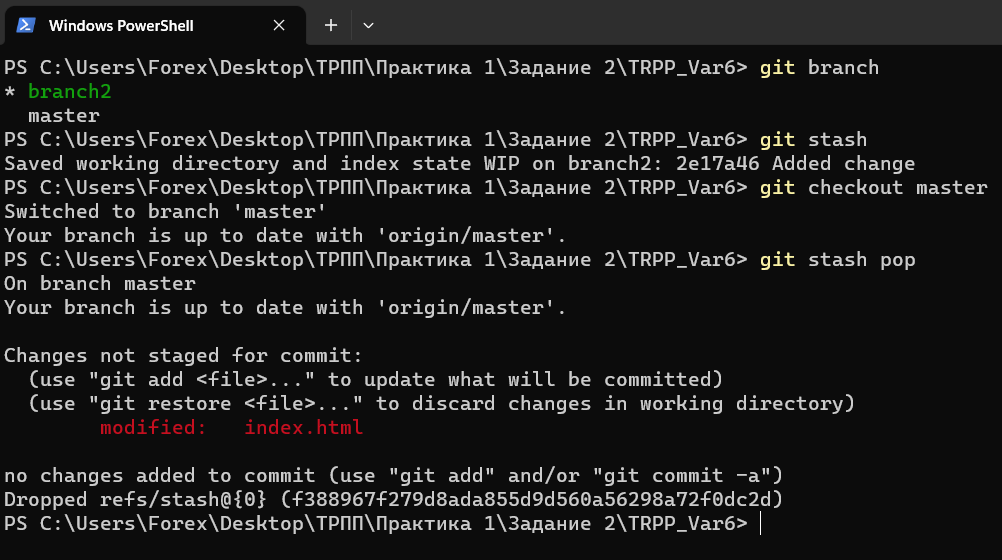


Рисунок 21 – Перенос изменений с ветки branch2 на master

1. **Выведите в консоли различия между веткой master и новой веткой**

На рисунке 22 показаны различия между веткой branch2 и master.

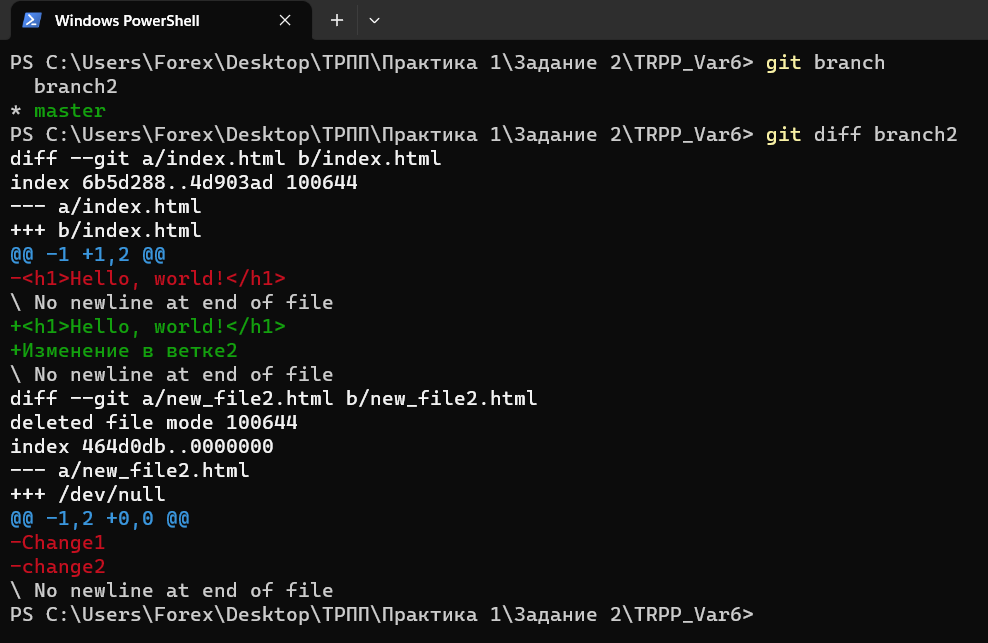


Рисунок 22 – Сравнение веток branch2 и master

1. **Слейте новую ветку с master при помощи merge**

На рисунке 23 показан процесс слияние ветки branch2 и master.

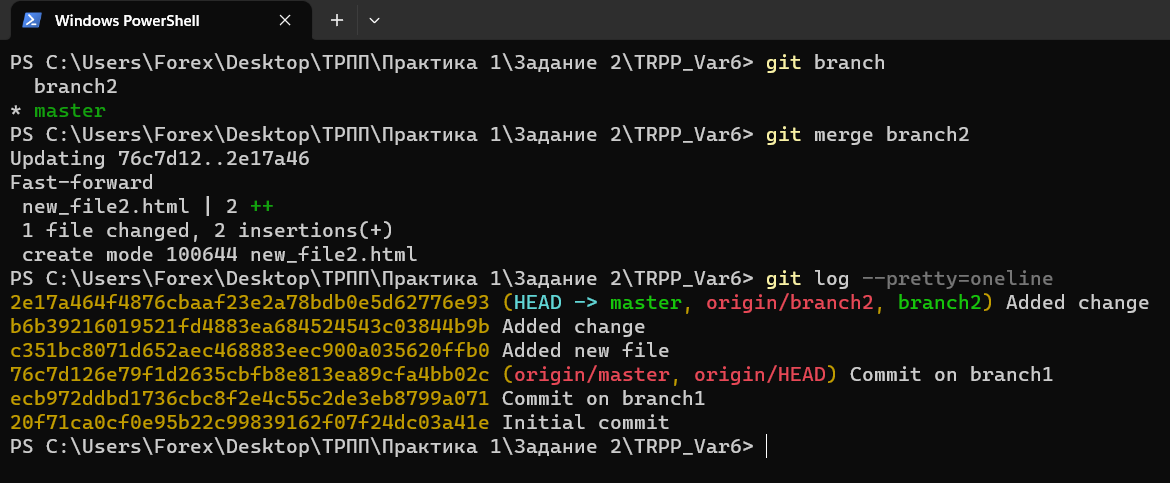


Рисунок 23 – слияние ветки branch2 и master

# Часть 3. Работа с ветвлением и оформление кода

**Вариант 7**

1. **Сделайте форк репозитория в соответствии с Вашим вариантом**

На рисунке 24 показан процесс создания форка репозитория.

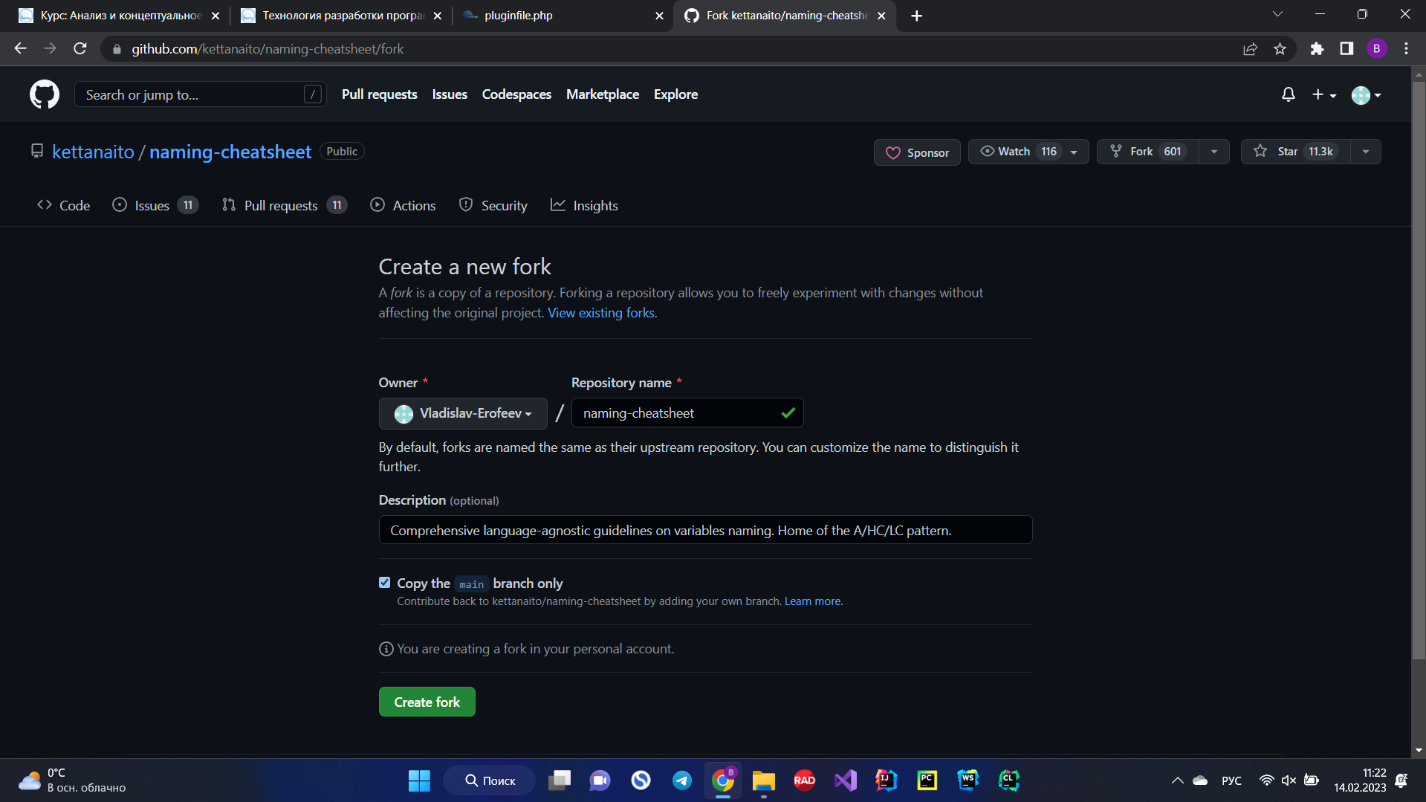


Рисунок 24 – Форк репозитория

1. **Склонируйте его (форк репозитория) на локальную машину.**

На рисунке 25 показан процесс клонирования репозитория на локальную машину.



Рисунок 25 – Клонирование репозитория на локальную машину

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

На рисунке 26 показан процесс создания двух веток branch1 и branch2 от последнего коммита в master.

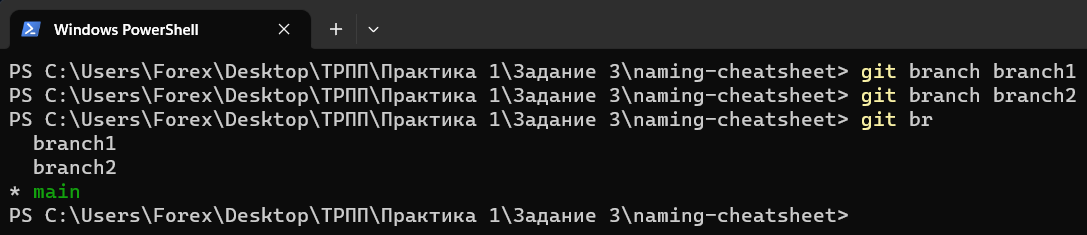


Рисунок 26 – Создание двух веток от последнего коммита

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

На рисунке 27 показаны 3 коммита на каждой из веток.

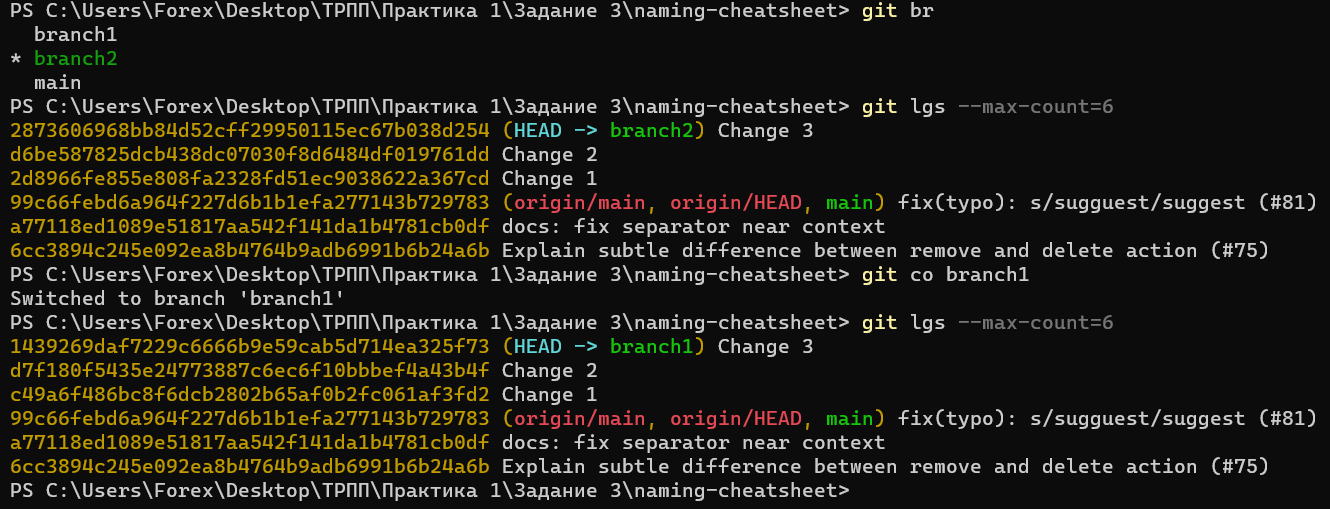


Рисунок 27 – Создание 3-ёх коммитов на ветках

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

На рисунке 28 показан процесс слияния ветки branch и branch2 и разрешения при этом конфликтов.

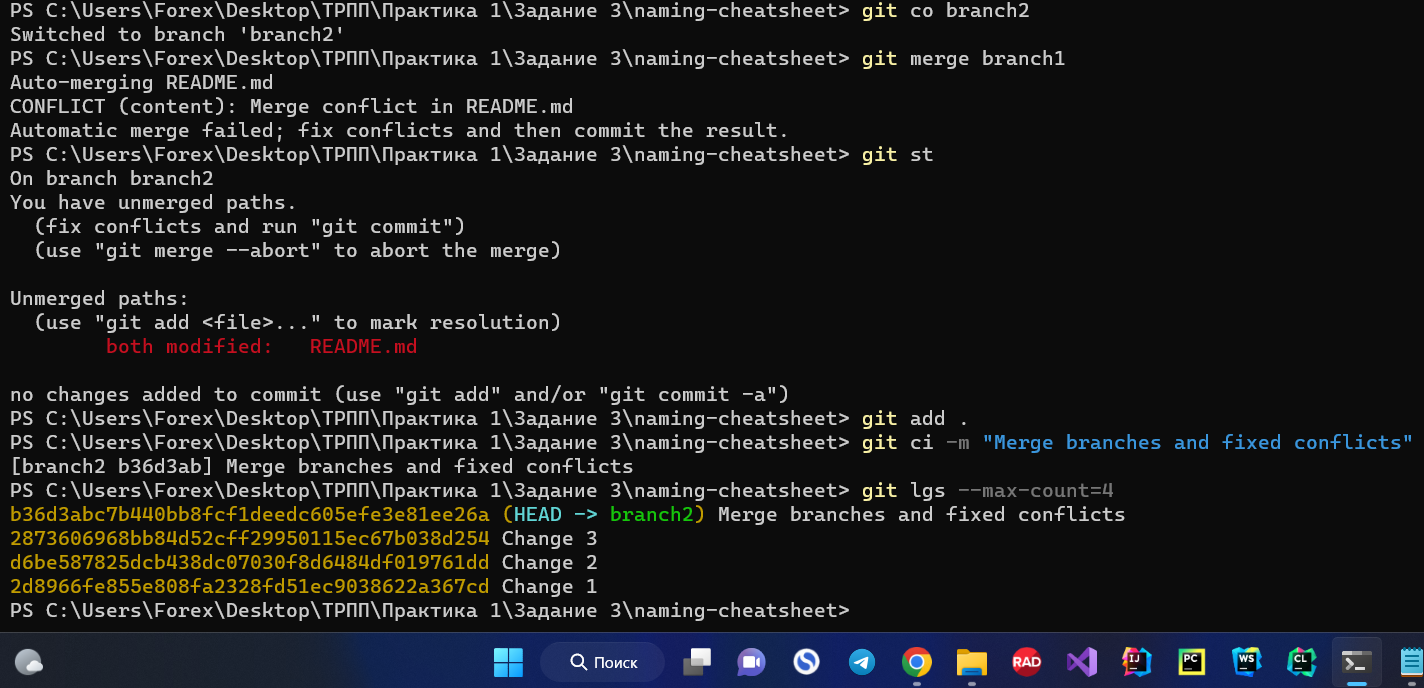


Рисунок 28 – Слияние двух веток

1. **Выгрузите все изменения во всех ветках в удаленный репозиторий.**

На рисунке 29 показан процесс выгрузки всех изменений в удалённый репозиторий.

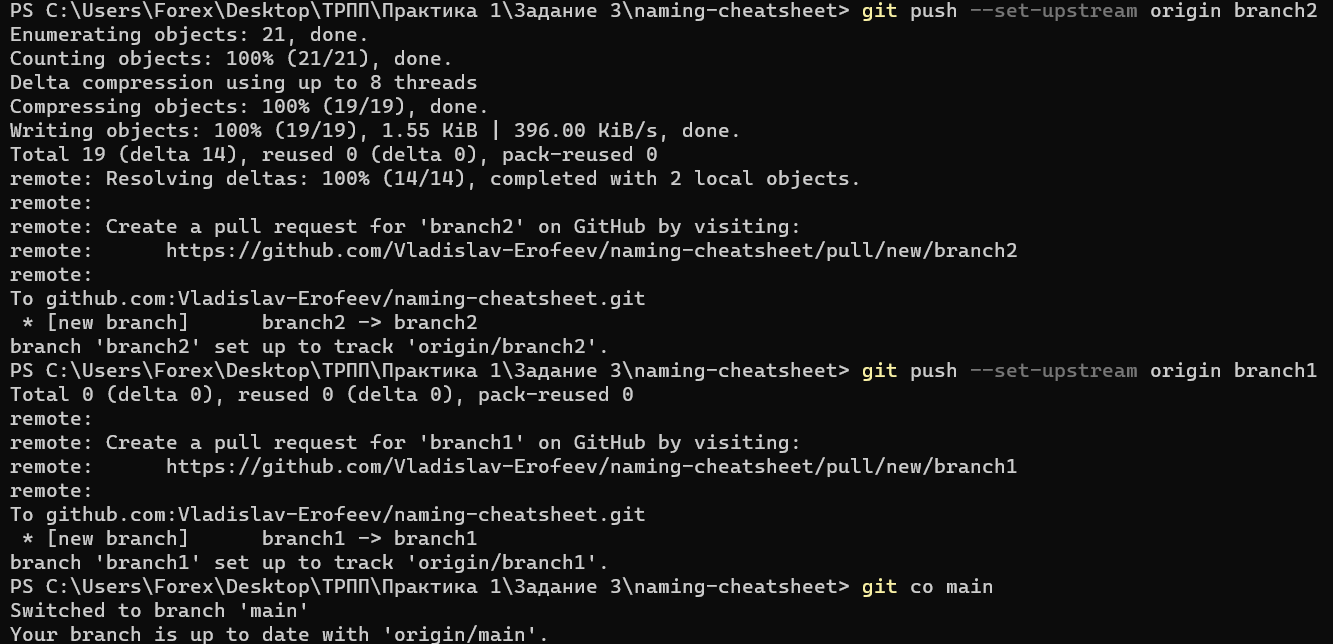


Рисунок 29 – Выгрузка изменений в удалённый репозиторий

1. **Проведите еще 3 коммита в ветку branch1**

На рисунке 30 показан процесс создания 3-ёх коммитов на ветке branch1.

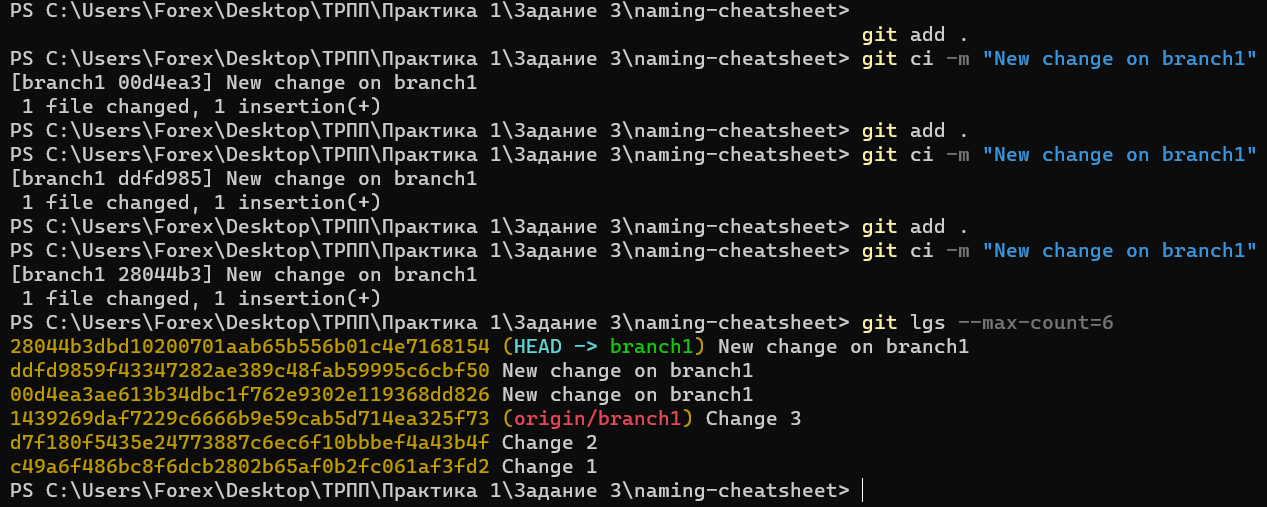


Рисунок 30 – 3 коммита на ветке branch1

1. **Склонируйте репозиторий еще раз в другую директорию**

На рисунке 31 показан процесс клонирования репозитория в другую директорию.



Рисунок 31 – Клонирование репозитория

1. **В новом клоне репозитории сделайте 3 коммита в ветку branch1**

На рисунке 32 показан процесс создания 3-ёх коммитов на ветке branch1 в новом репозитории.

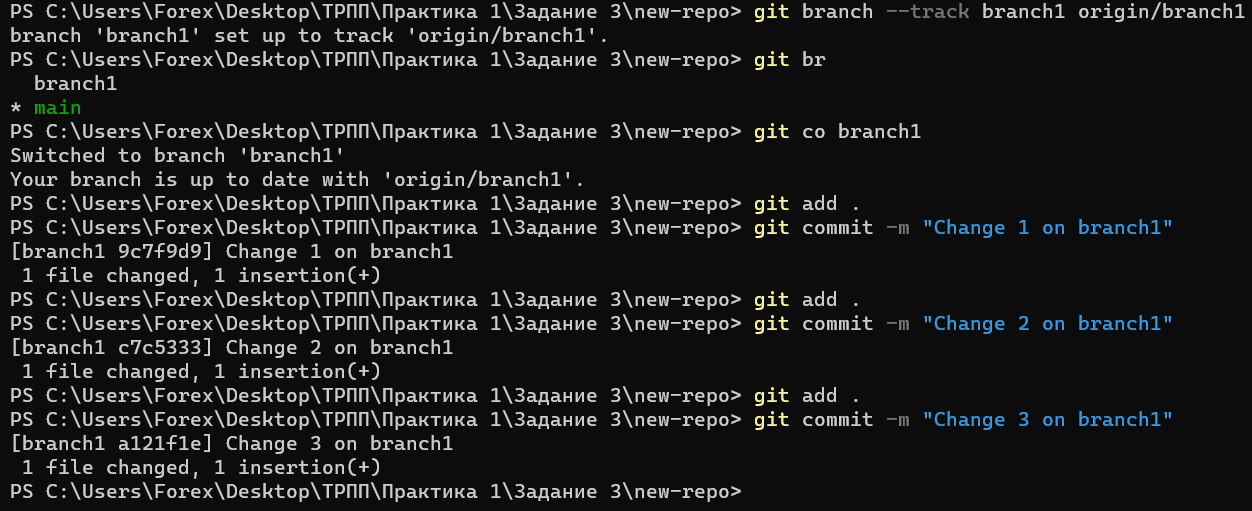


Рисунок 32 – Создание коммитов в новом репозитории

1. **Выгрузите все изменения из нового репозитория в удаленный репозиторий**

На рисунке 33 показан процесс выгрузки изменений в удалённые репозиторий.

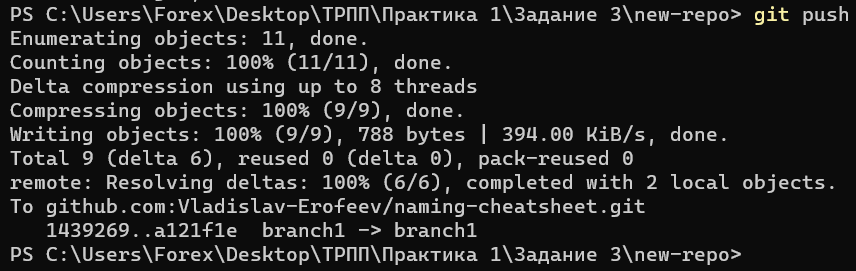


Рисунок 33 – Выгрузка изменений в удалённый репозиторий

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

На рисунке 34 показан процесс выгрузки изменений с тегом –force.

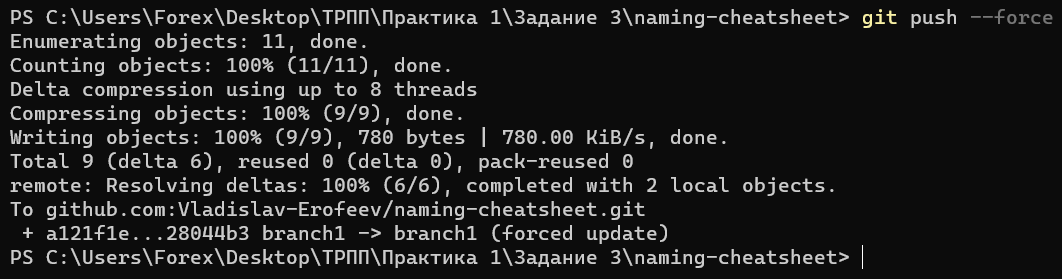


Рисунок 34 – Выгрузка в удалённый репозиторий с тегом –force

1. **Получите все изменения в новом репозитории**

На рисунке 35 показан процесс получения изменений из удалённого репозитория.

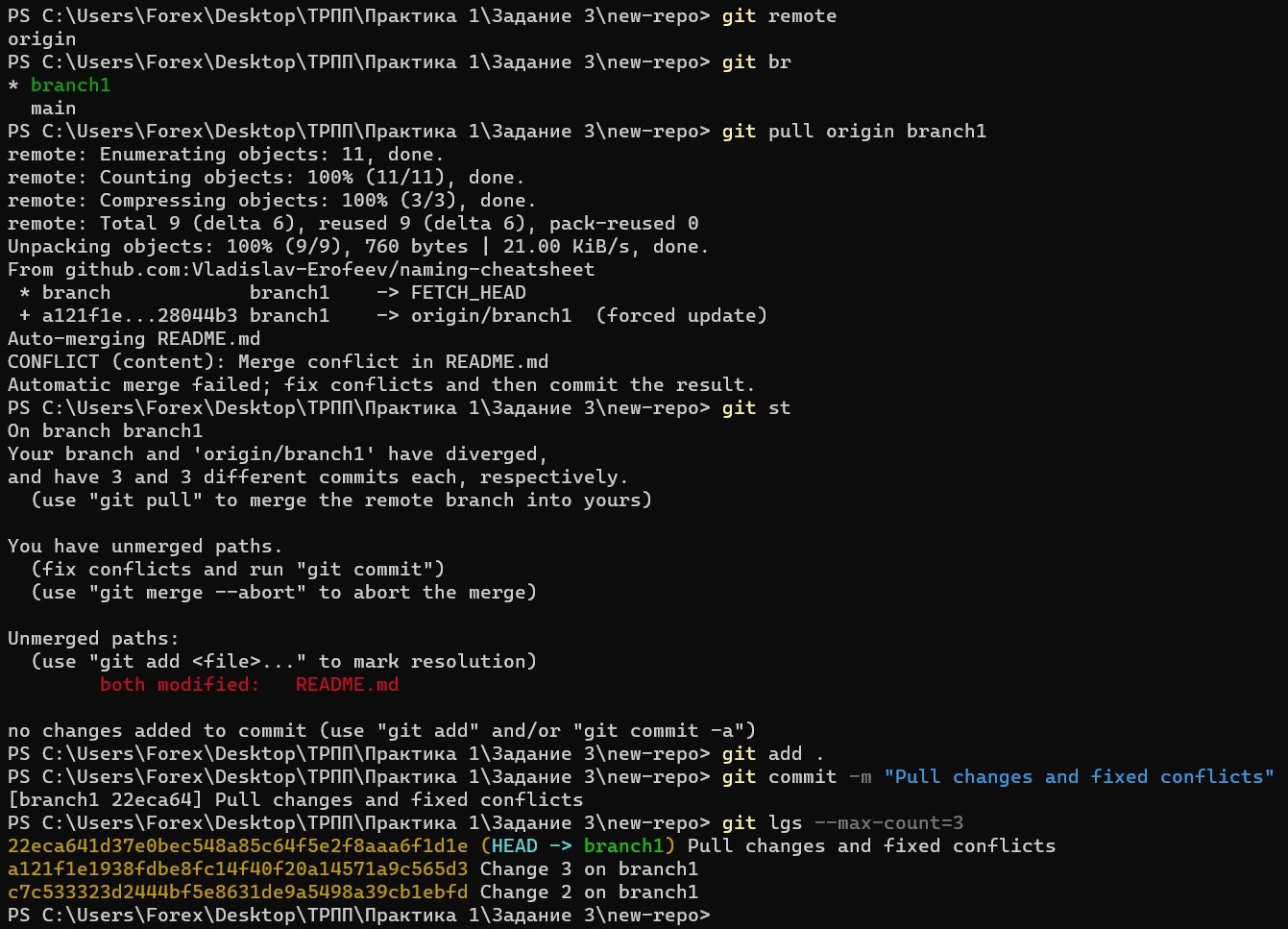


Рисунок 35 – Получение изменений из удалённого репозитория

# Ответы на вопросы

**2. К какому типу систем контроля версий относится Git?**

Git относится к распределённой системе контроле версий.

**3. Что такое репозиторий Git?**

Репозиторий – место, где система контроля версий хранит свои метаданные и базу данных объектов проекта.

Локальный репозиторий — репозиторий, расположенный на локальном компьютере разработчика в каталоге. Именно в нём происходит разработка и фиксация изменений, которые отправляются на удалённый репозиторий.

Удалённый репозиторий — репозиторий, находящийся на удалённом сервере. Это общий репозиторий, в который приходят все изменения и из которого забираются все обновления.

**4. Что такое коммит?**

Коммиты – основные конструктивные элементы временной шкалы 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.

**5. Что такое ветка в репозитории Git?**

Ветка (Branch) — это параллельная версия репозитория. Она включена в этот репозиторий, но не влияет на главную версию, тем самым позволяя свободно работать в параллельной. Когда вы внесли нужные изменения, то вы можете объединить их с главной версией.

**11. Что делает команда git status?**

Команда git status отображает состояние рабочего каталога и раздела проиндексированных файлов. С её помощью можно проверить индексацию изменений и увидеть файлы, которые не отслеживаются Git.

**12. 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.

**18. Что сделает команда "git branch" без какого-либо параметра?**

Команда git branch без параметров выведет список всех веток локального репозитория. Также она выделит зелёным цветом ветку, на которой находится указатель HEAD.

**27. Как сделать ветку с названием my\_branch?**

Для создания ветки с именем my\_branch необходимо написать команду git branch my\_branch в консоль, либо можно написать команду git checkout -b my\_branch, которая создаст новую ветку и переключит указатель HEAD на неё.