# ОГЛАВЛЕНИЕ

[ОГЛАВЛЕНИЕ 2](#_wd7qj5o1xo6z)

[ФОРМУЛИРОВКА ЗАДАНИЙ 3](#_mopfgu3pzw3n)

[ВЫПОЛНЕНИЕ ЗАДАНИЙ 6](#_y4x6kflnrm2x)

[Часть 1 6](#_4qskc9cpwolr)

[Часть 2 11](#_y4gd8kasrepx)

[Часть 3 17](#_ps946n3p8e0o)

[КОНТРОЛЬНЫЕ ВОПРОСЫ 27](#_9m8n6tmzyb3e)

[ВЫВОД 29](#_6pz9i0uyu4xs)

# ФОРМУЛИРОВКА ЗАДАНИЙ

**Цель работы**: получить навыки по работе с командной строкой и git’ом.

**Часть 1. Основные команды Git**

1. Установите и настройте клиент git на своей рабочей станции.

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

3. Внесите изменения в один из файлов.

4. Проиндексируйте изменения и проверьте состояние.

5. Сделайте коммит того, что было проиндексировано в репозиторий. Добавьте к коммиту комментарий.

6. 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.

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

8. Верните рабочий каталог к одному из предыдущих состояний.

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

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

11. Отмените один из коммитов в локальном репозитории.

**Часть 2. Системы управления репозиториями**

1. Создайте аккаунт на GitHub (у кого нет),

2. Создайте репозиторий на GitHub и на локальной машине, согласно выбранной теме проекта,

3. Создайте несколько файлов на локальной машине при помощи консоли,

4. Создайте SSH-ключ для авторизации,

5. Свяжите репозиторий локальной машины с репозиторием на GitHub при помощи консоли,

6. Создайте новую ветку в репозитории с помощью команды, произведите в ней какие-нибудь изменения, а после слейте с веткой master,

7. Выполните цепочку действий в репозитории, согласно вариантам.

| № Варианта | Задание |
| --- | --- |
| 2 | 1) Клонируйте непустой удаленный репозиторий на локальную машину  2) Создайте новую ветку и выведите список всех веток  3) Произведите коммит в ветке master  4) Произведите 3 коммита в новой ветке в разные файлы  5) Выгрузите изменения в удаленный репозиторий  6) Откатите ветку обратно на 2 коммита (в том числе в удаленном репозитории)  7) Выведите в консоли различия между веткой master и новой веткой  8) Перебазируйте новую ветку на master |

**Часть 3. Работа с ветвлением и оформление кода**

1. Сделайте форк репозитория в соответствии с вашим вариантом

2. Склонируйте его на локальную машину

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

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

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

6. Выгрузите все изменения во всех ветках в удаленный репозиторий

7. Проведите еще 3 коммита в ветку branch1

8. Склонируйте репозиторий еще раз в другую директорию

9. В новом клоне репозитории сделайте 3 коммита в ветку branch1

10. Выгрузите все изменения из нового репозитория в удаленный репозиторий

11. Вернитесь в старый клон с репозиторием, выгрузите изменения с опцией --force

12. Получите все изменения в новом репозитории

| № Варианта | Задание |
| --- | --- |
| 2 | https://github.com/google/model\_search |

# ВЫПОЛНЕНИЕ ЗАДАНИЙ

# Часть 1

Клиент git был установлен и настроен.

Создание локального репозитория с несколькими файлами, и внесение изменений в один из них.

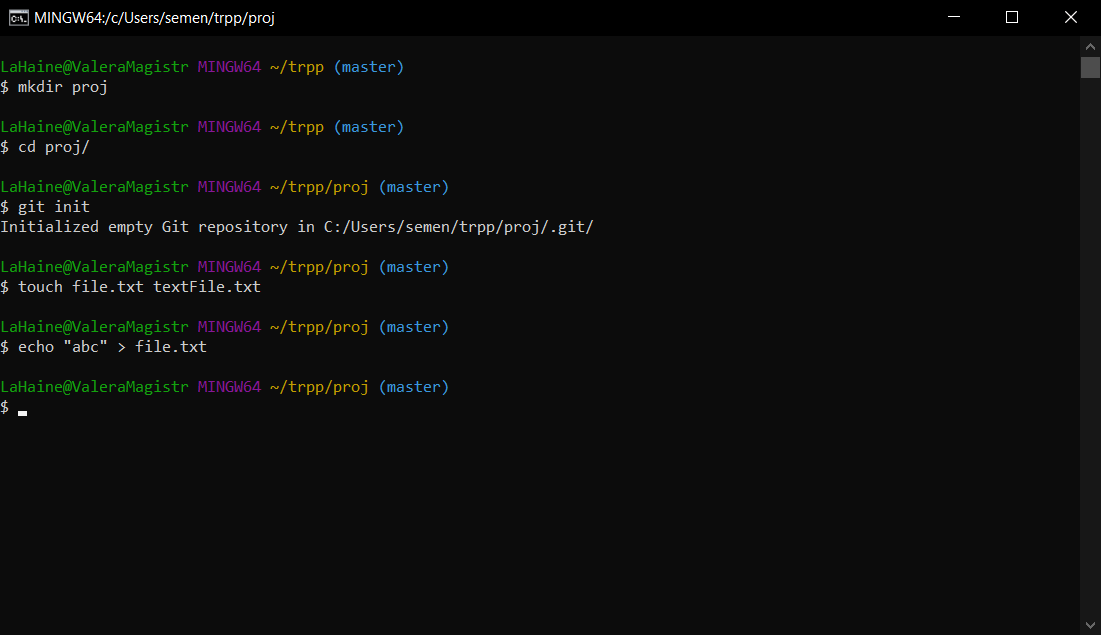


Рисунок 1 – Выполнение заданий 1-3 первой части

Индексирование изменений и проверка состояния, коммит проиндексированных изменений и добавление комментария.

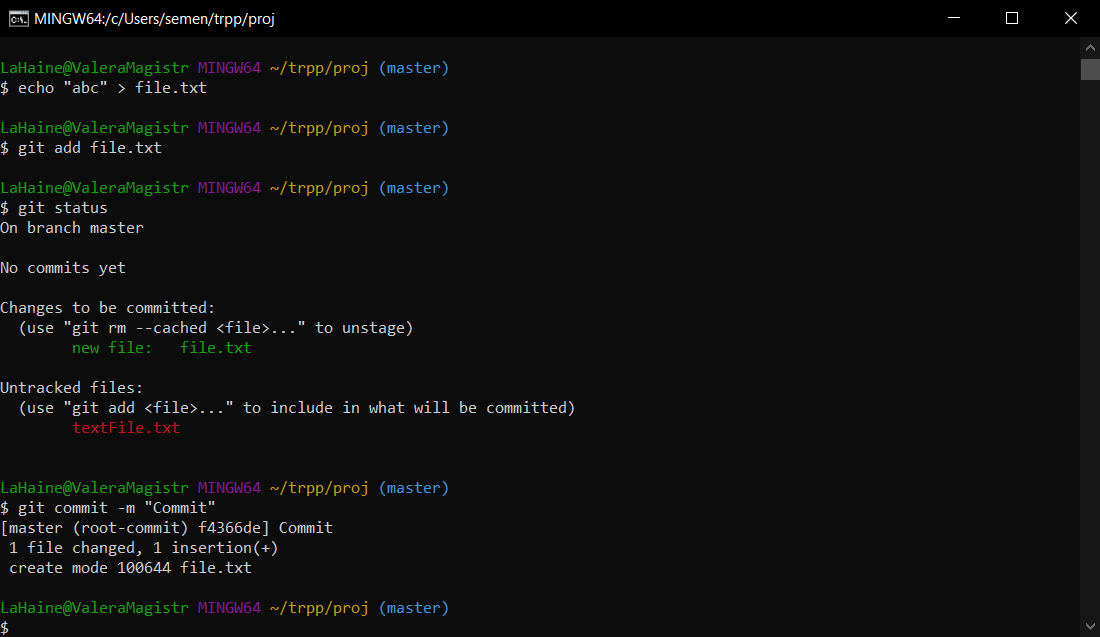


Рисунок 2 – Выполнение заданий 3-5 первой части

Изменение файла, индексирование, изменение файла повторно с проверкой состояния. Создание коммита с изменением. Индексирование второго изменения с проверкой состояния и последующим коммитом.

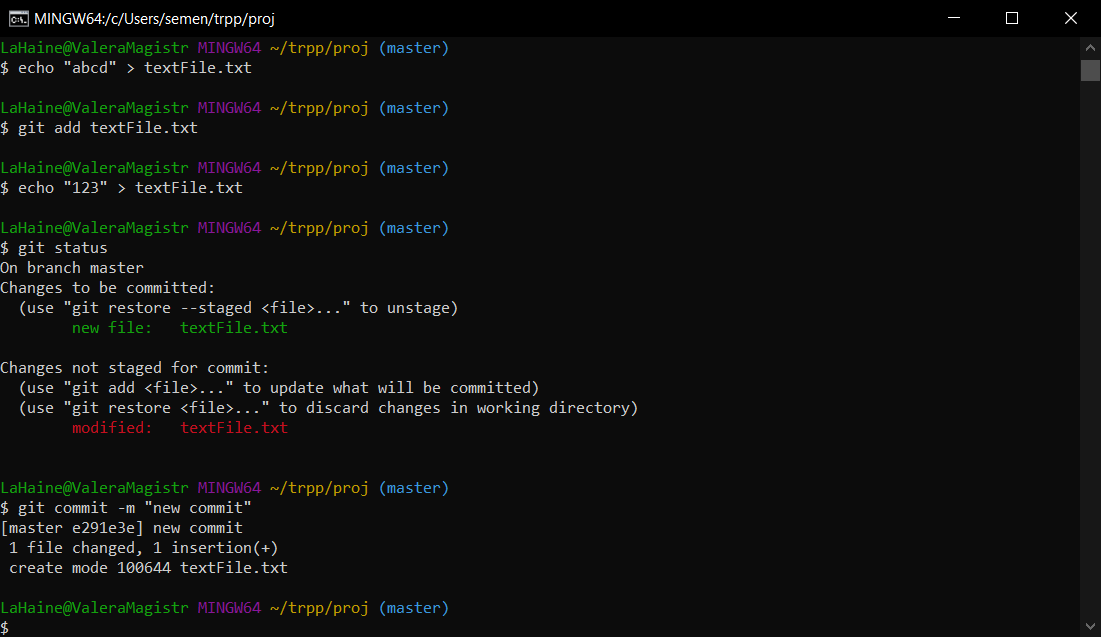


Рисунок 3 – Выполнение задания 6 первой части



Рисунок 4 – Выполнение задания 6 первой части

Просмотр истории коммитов, использование различных форм для отображения истории коммитов.

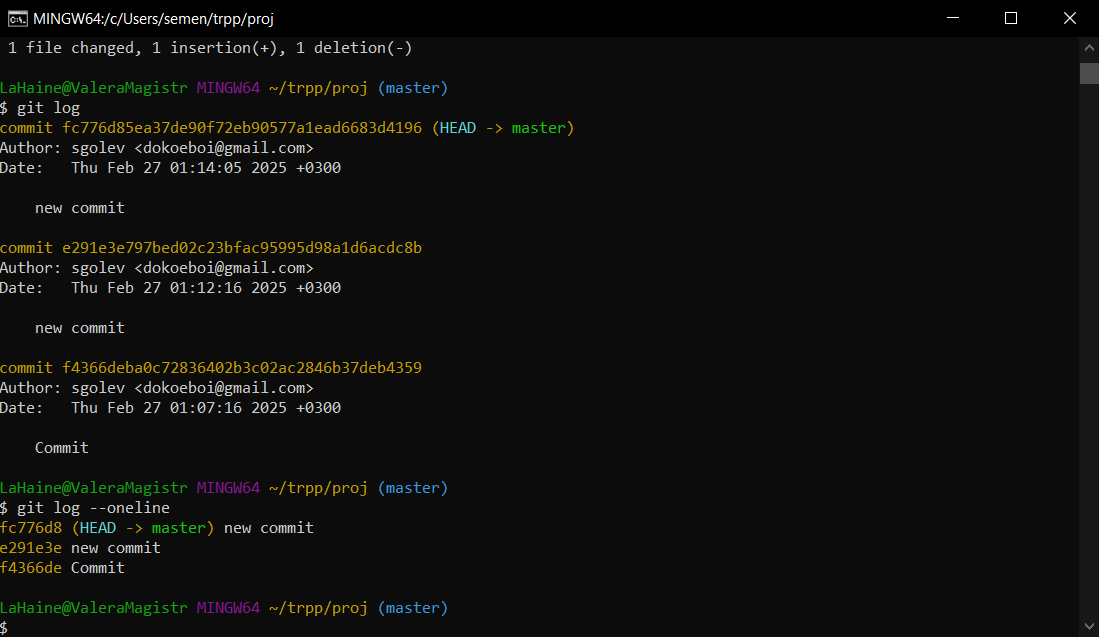


Рисунок 5 – Выполнение задания 7 первой части

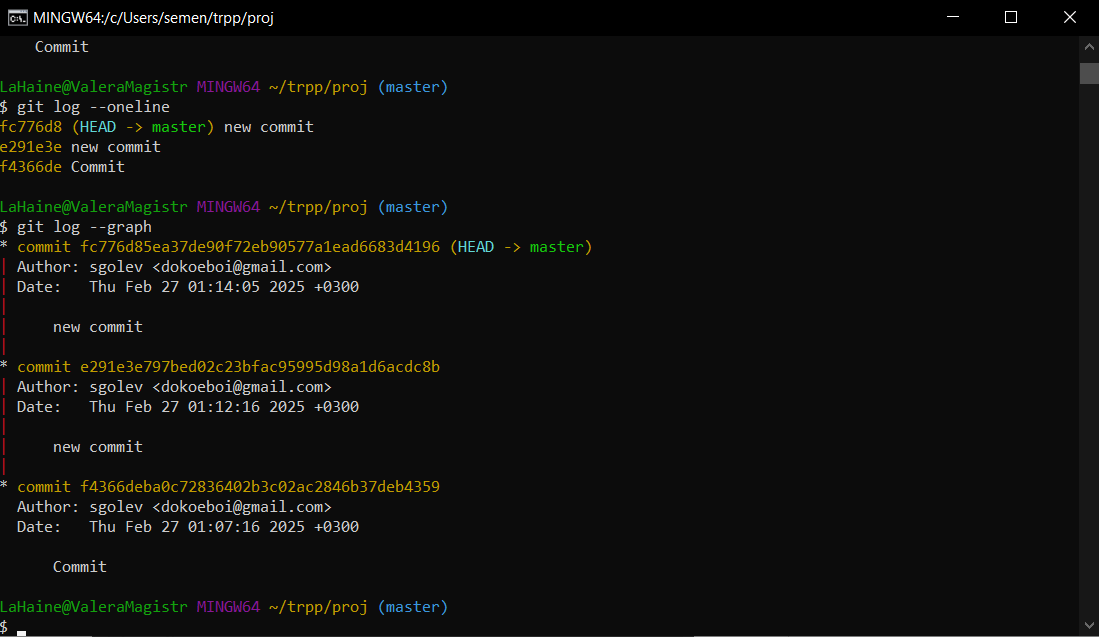


Рисунок 6 – Выполнение задания 7 первой части

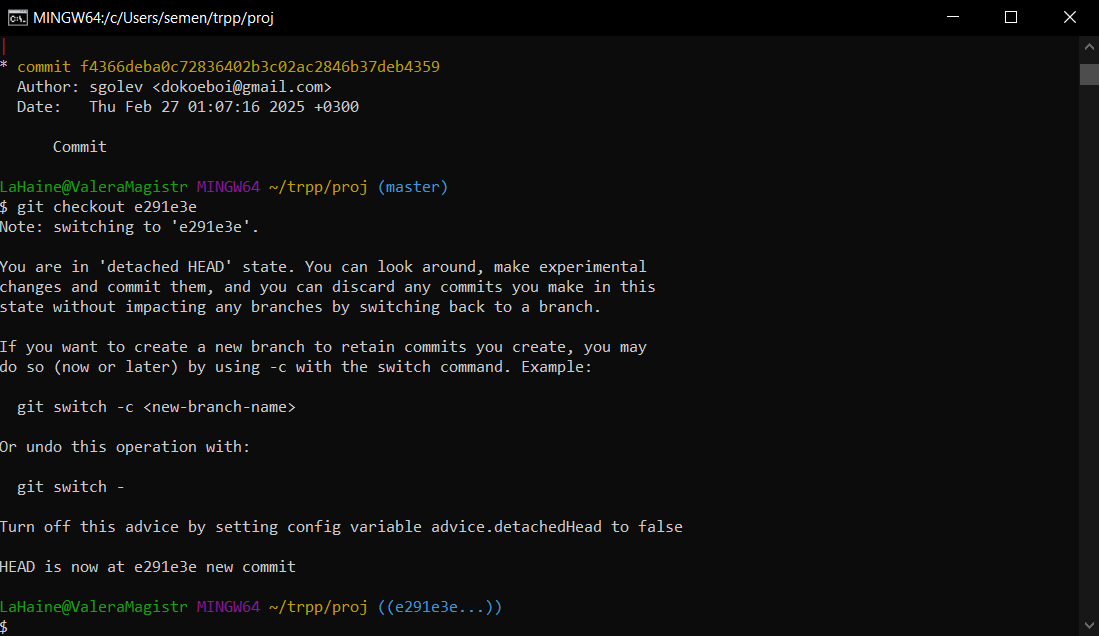
Возвращение рабочего каталога к состоянию на момент второго коммита (хэш : e291e3e).  


Рисунок 7 – Выполнение задания 8 первой части

Изучено создание тегов.

Внесение изменений в файл, отменим это изменение, далее произведены те же действия, но с индексацией изменений.



Рисунок 8 – Выполнение задания 10 первой части

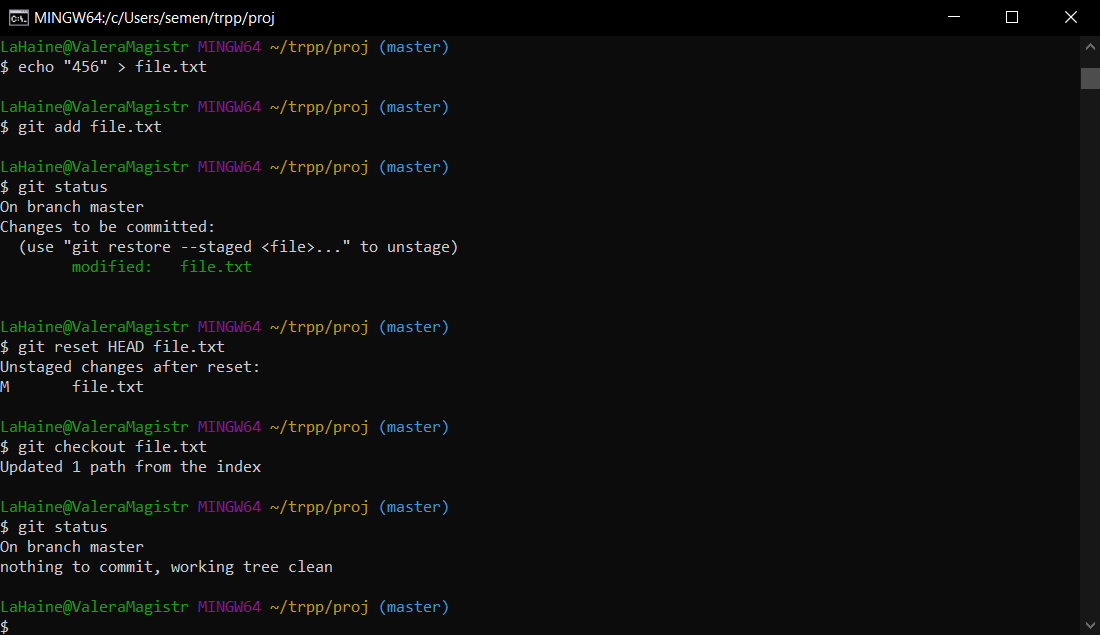


Рисунок 9 – Выполнение задания 10 первой части

Внесение изменений в файл, индексирование и создания коммита с изменениями. Далее отмена этого коммита.



Рисунок 10 – Выполнение задания 11 первой части

# Часть 2

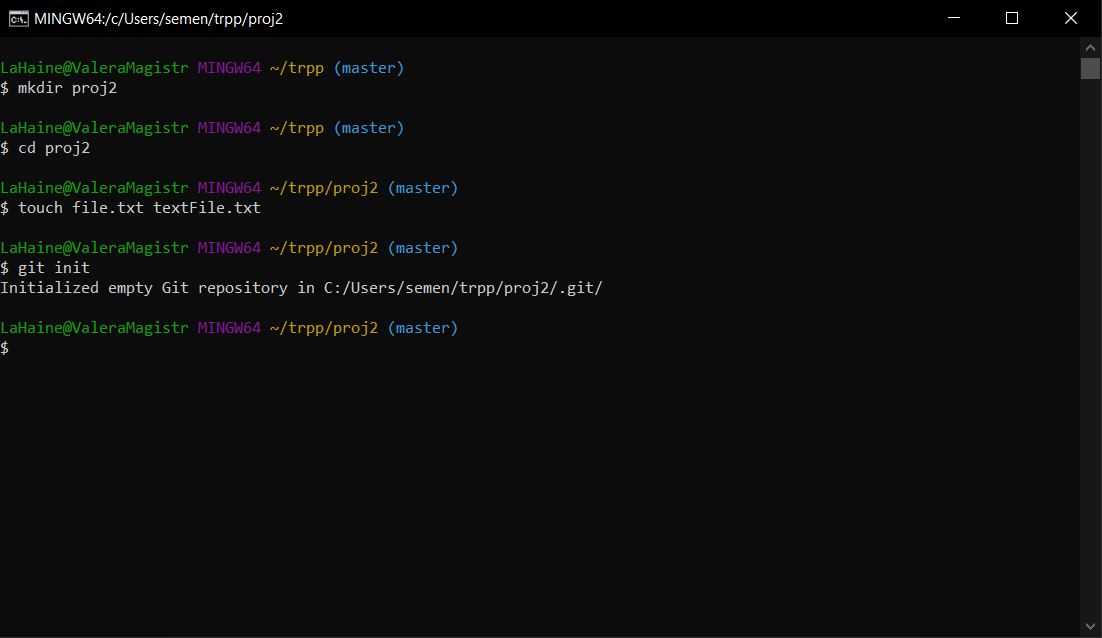
Аккаунт на GitHub был создан, также создан локальный репозиторий с парой файлов.  


Рисунок 11 – Выполнение заданий 1-3 второй части

Создан репозиторий на GitHub.

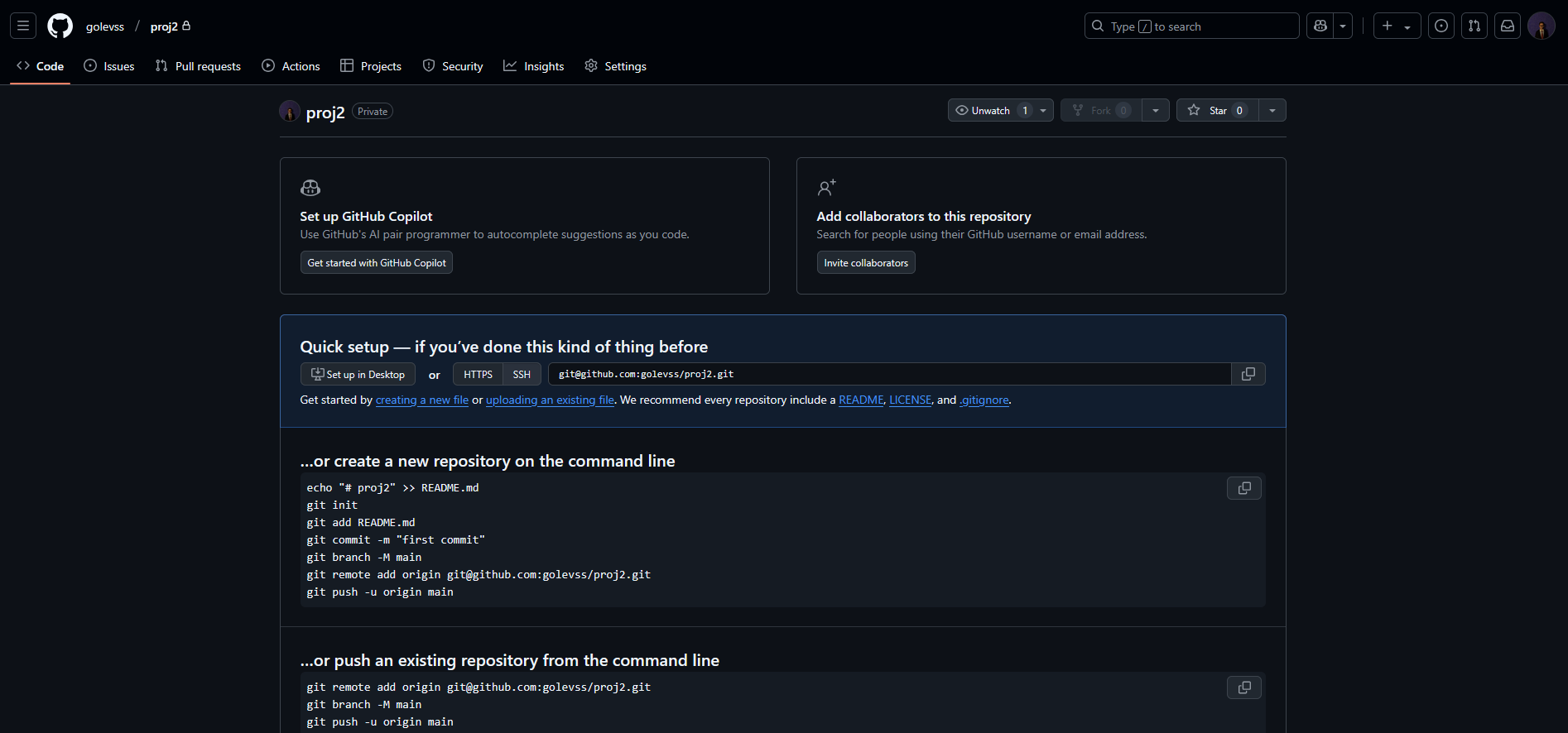


Рисунок 12 – Выполнение задания 2 второй части

Связывание локального репозитория, и репозитория созданного на GitHub.

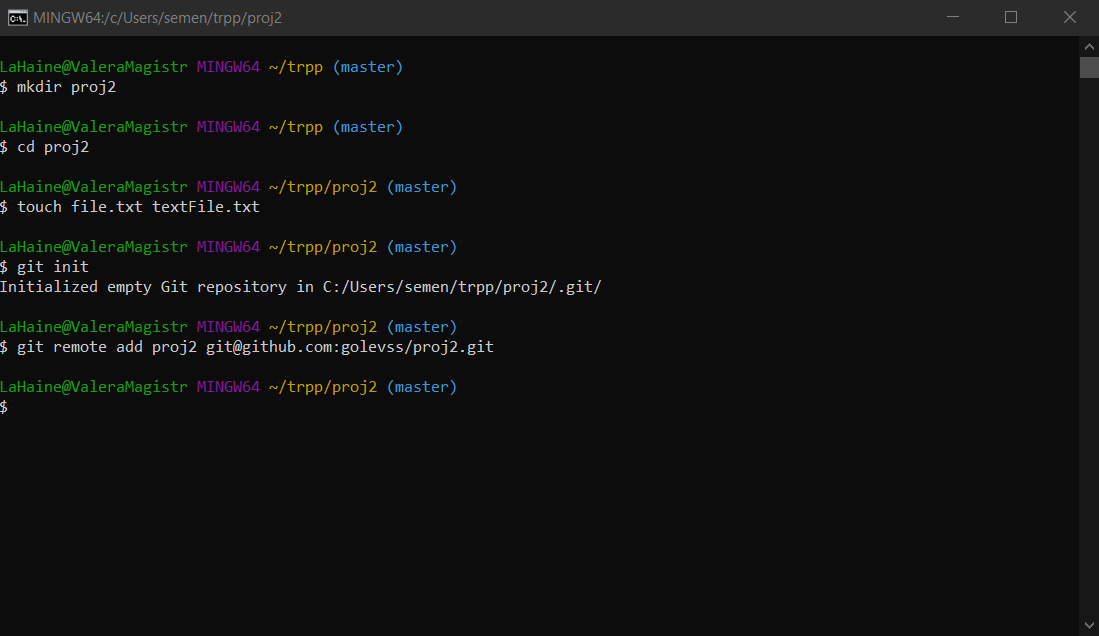


Рисунок 13 – Выполнение заданий 4-5 второй части

Создание новой ветки в репозитории, произведение изменений и сливание данной ветки с веткой master.

# 

Рисунок 14 – Выполнение задания 6 второй части

Клонирование непустого локального репозитория.



Рисунок 15 – Выполнение пункта 1 из заданий варианта второй части

Создание новой ветки, коммит в ветке master.

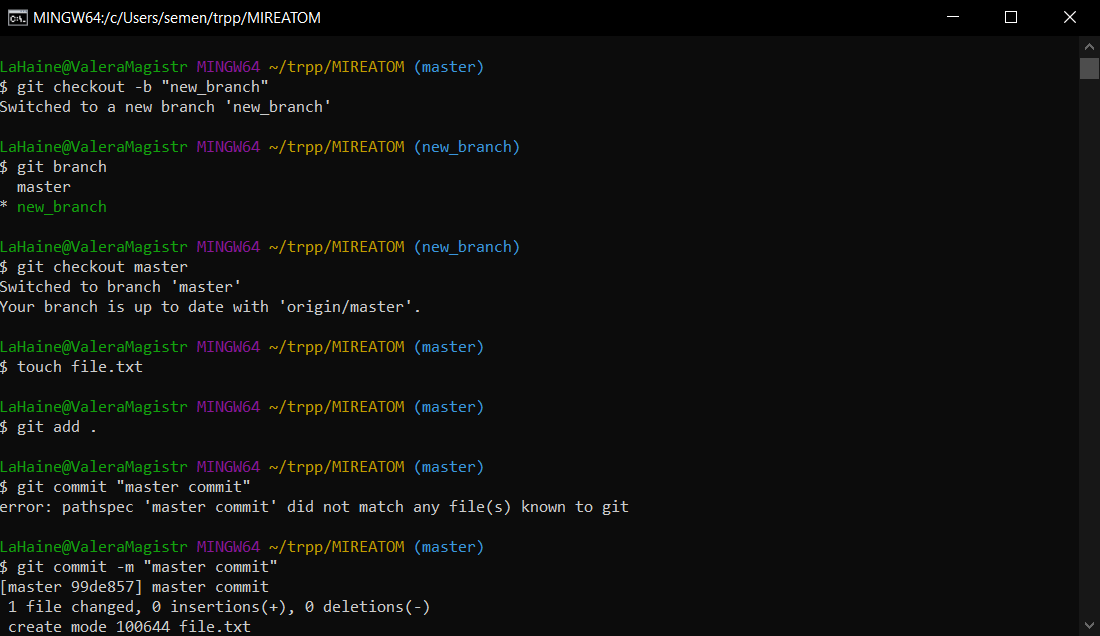


Рисунок 16 – Выполнение пункта 2 из заданий варианта второй части

Произведение трёх коммитов в новой ветке в разные файлы.

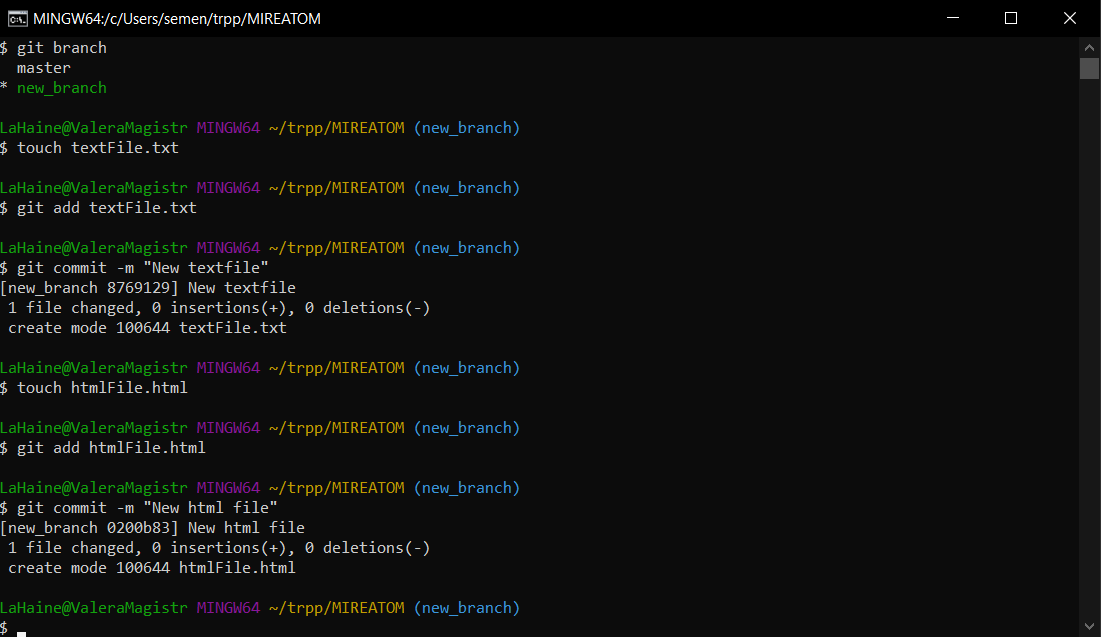


Рисунок 17 – Выполнение пункта 3 из заданий варианта второй части

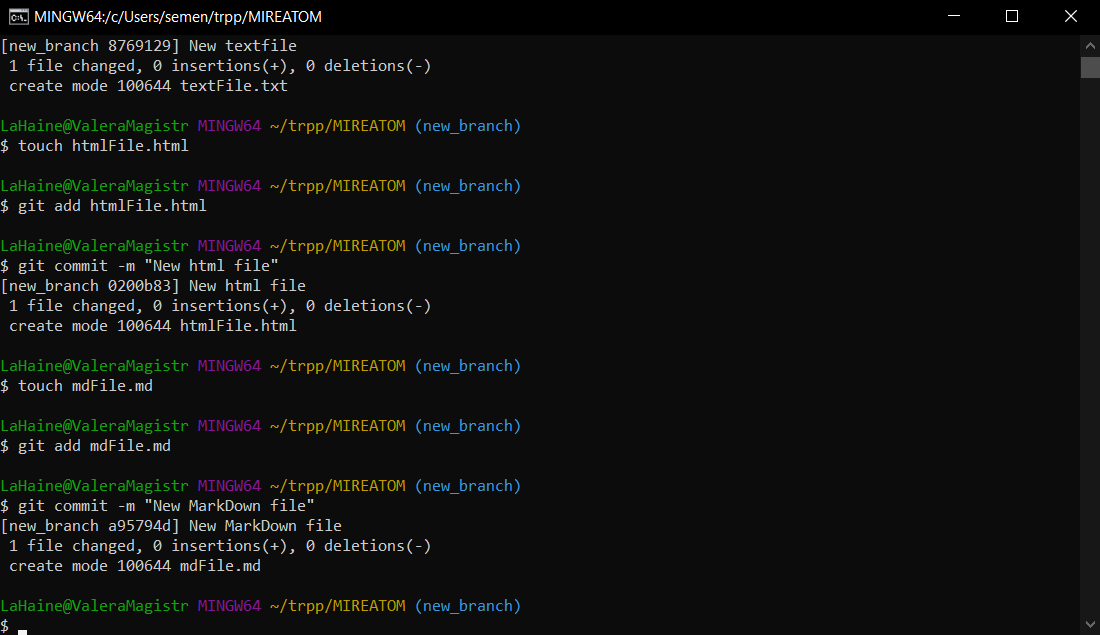


Рисунок 18 – Выполнение пункта 3 из заданий варианта второй части

Выгрузка изменений в удаленный репозиторий.

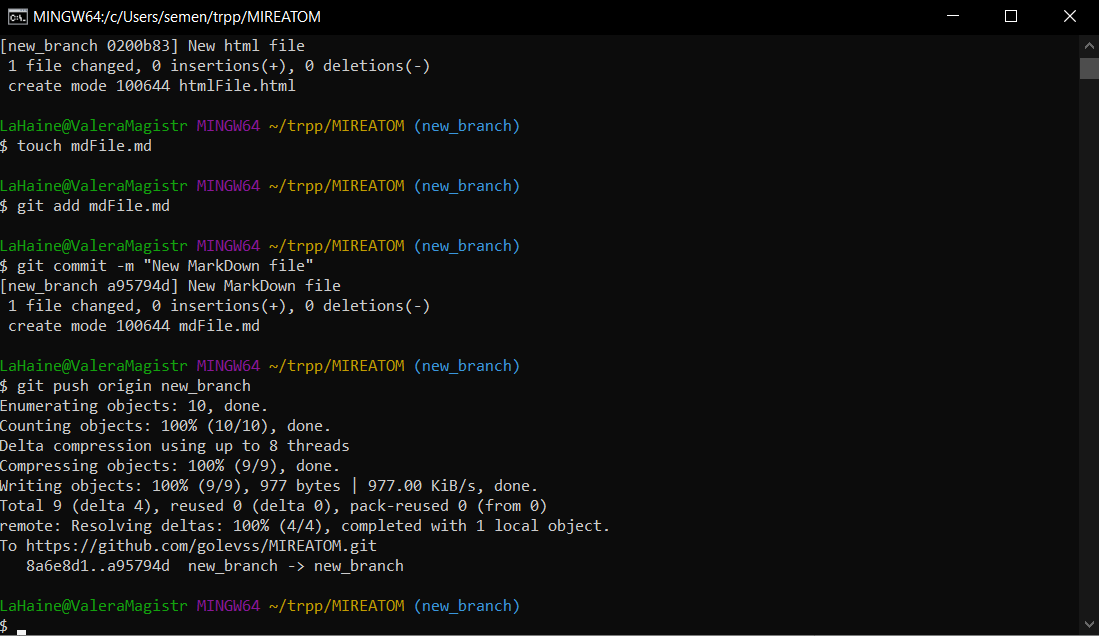


Рисунок 18 – Выполнение пункта 4 из заданий варианта второй части

Возвращение ветки на 2 коммита обратно, в том числе в удалённом репозитории.

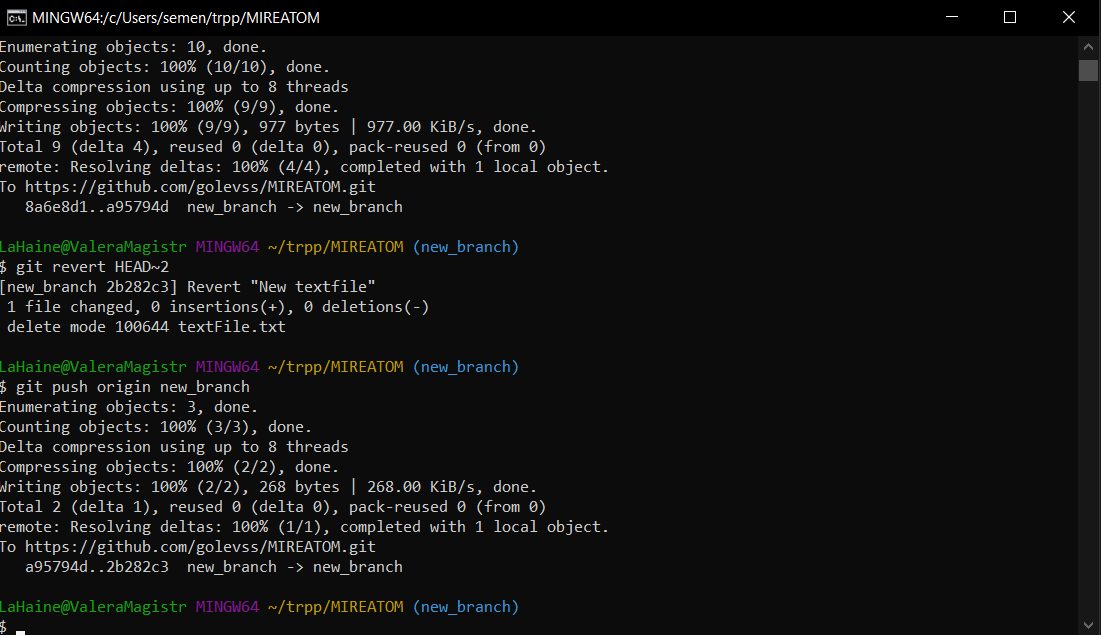


Рисунок 19 – Выполнение пункта 5 из заданий варианта второй части

Вывод различий между двумя ветками.

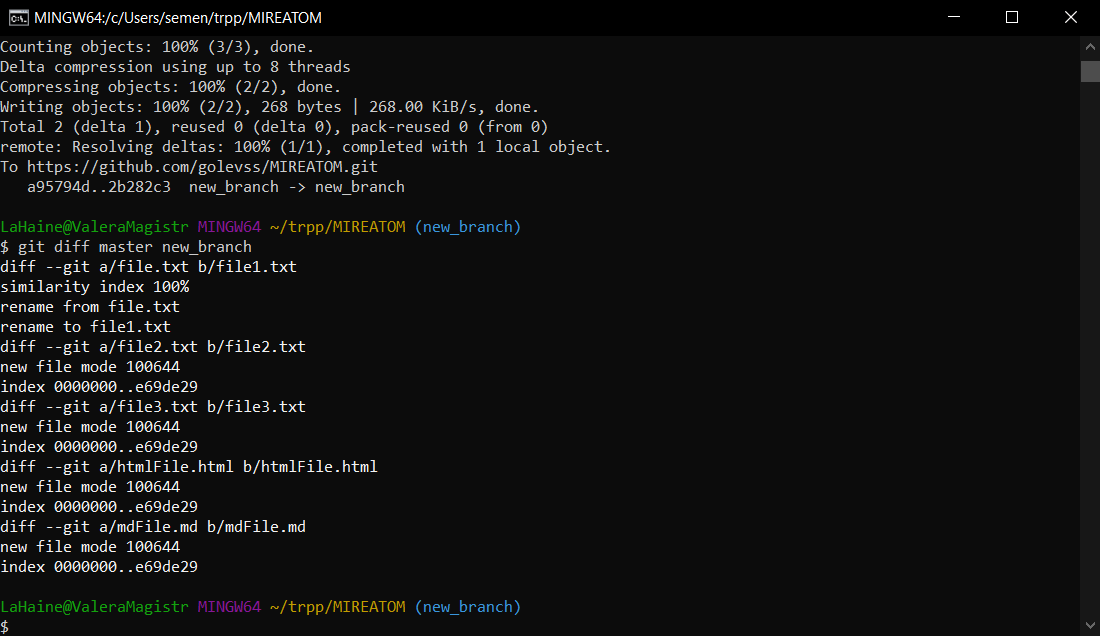


Рисунок 20 – Выполнение пункта 6 из заданий варианта второй части

Перебазирование новой ветки на ветку master.

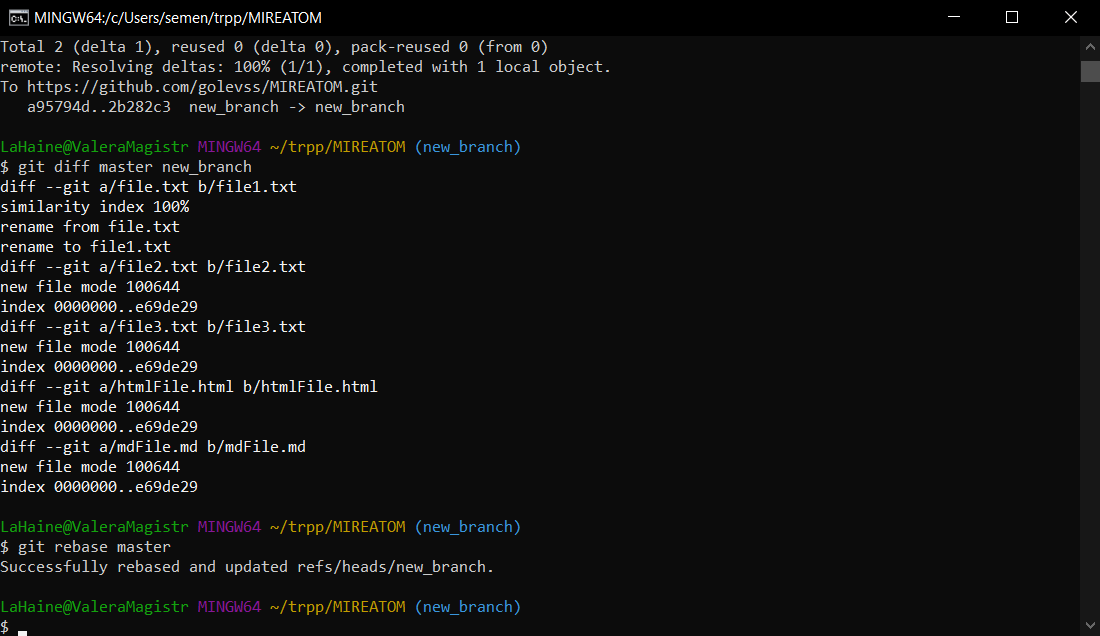


Рисунок 21 – Выполнение пунктов 7-8 из заданий варианта второй части

# Часть 3

Сделан форк репозитория указанного в задании варианта.

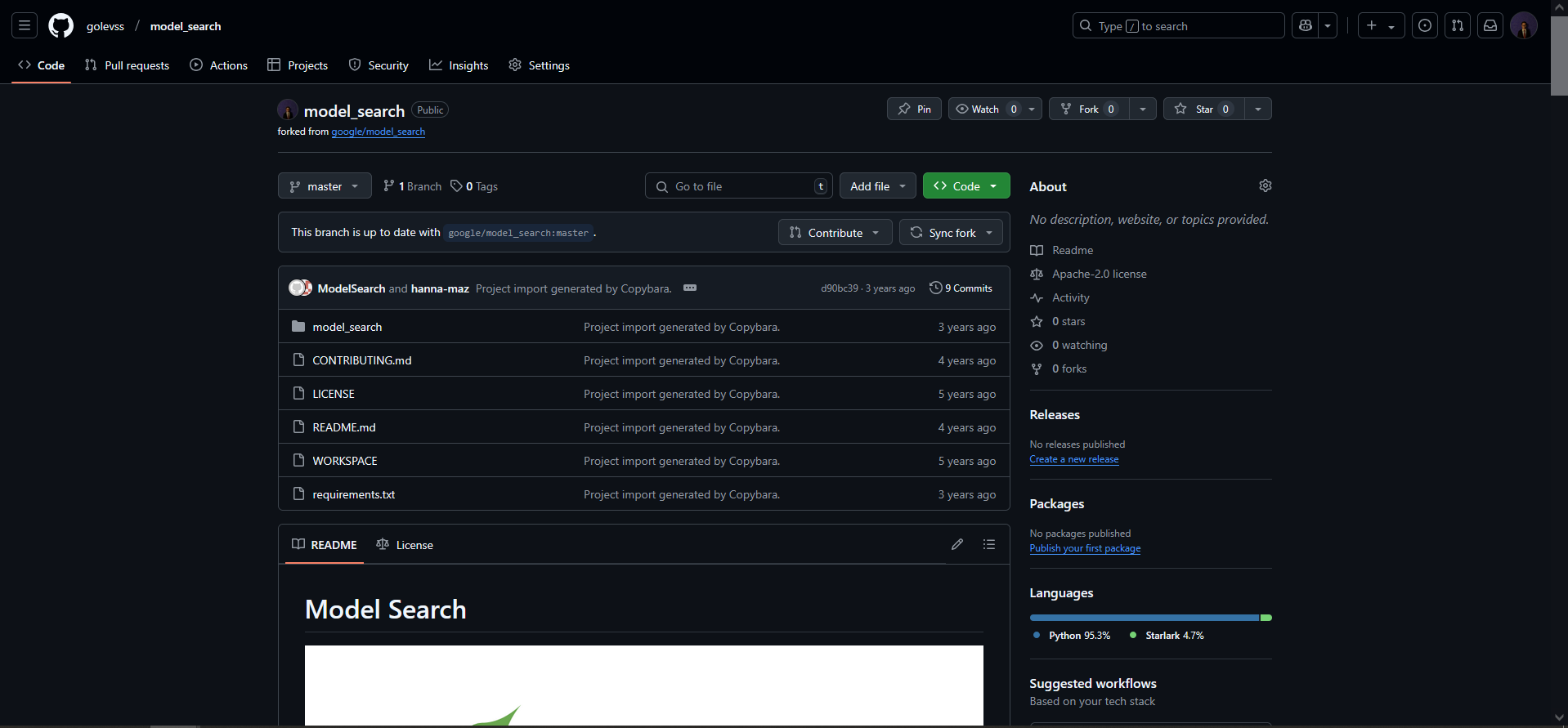


Рисунок 22 – Выполнение задания 1 третьей части

Репозиторий клонирован на локальную машину.

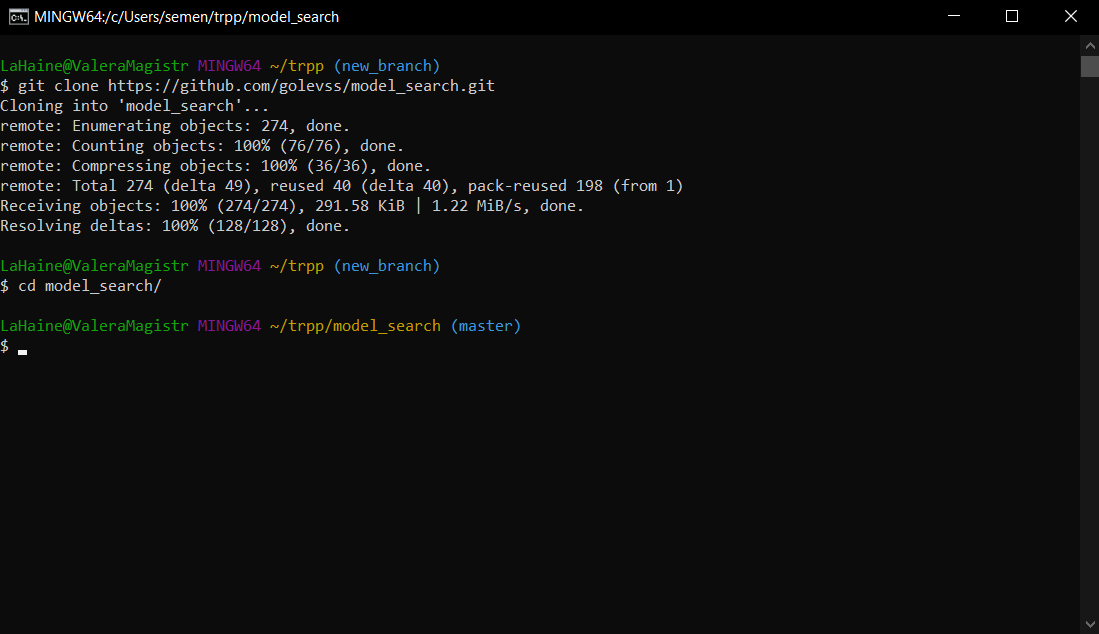


Рисунок 23 – Выполнение задания 2 третьей части

Создадим две новые ветки и проведём в каждой по три коммита.

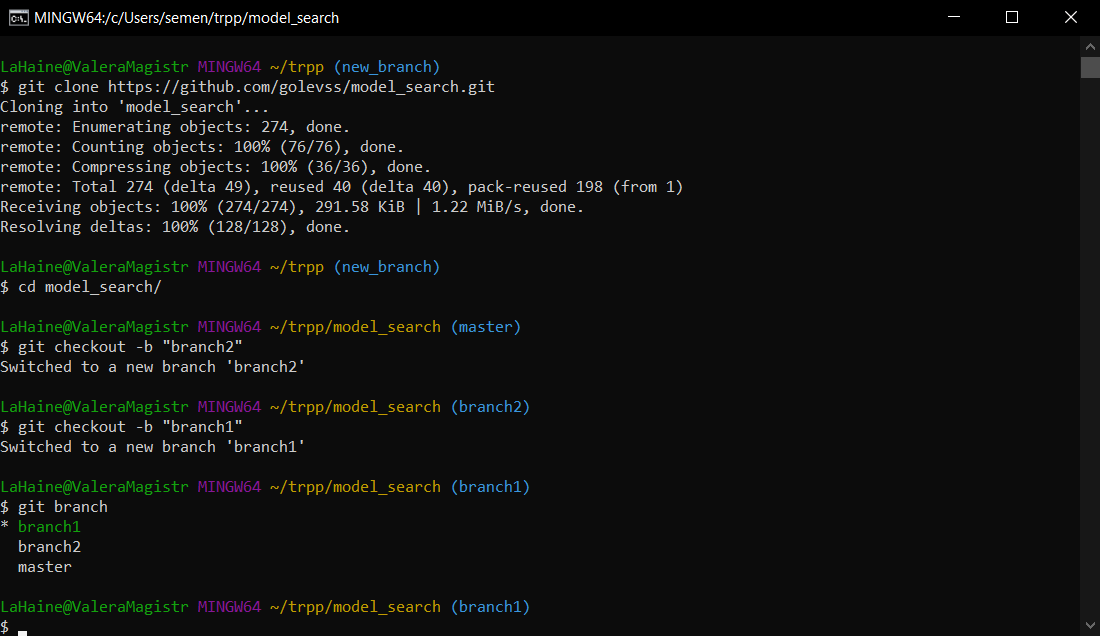


Рисунок 24 – Выполнение задания 3 третьей части

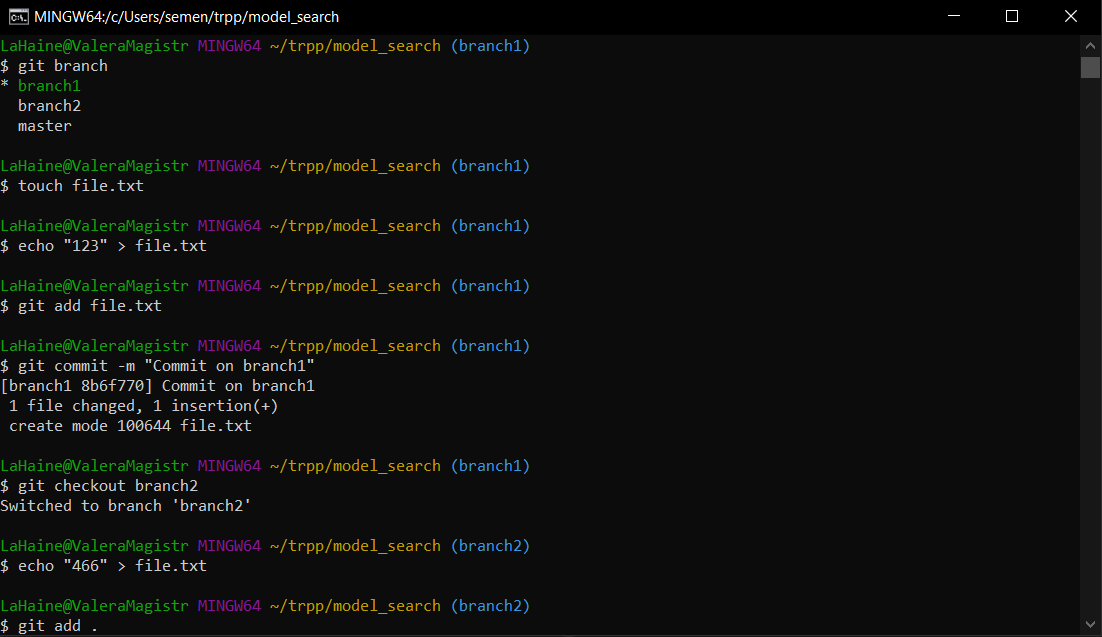


Рисунок 25 – Выполнение задания 4 третьей части

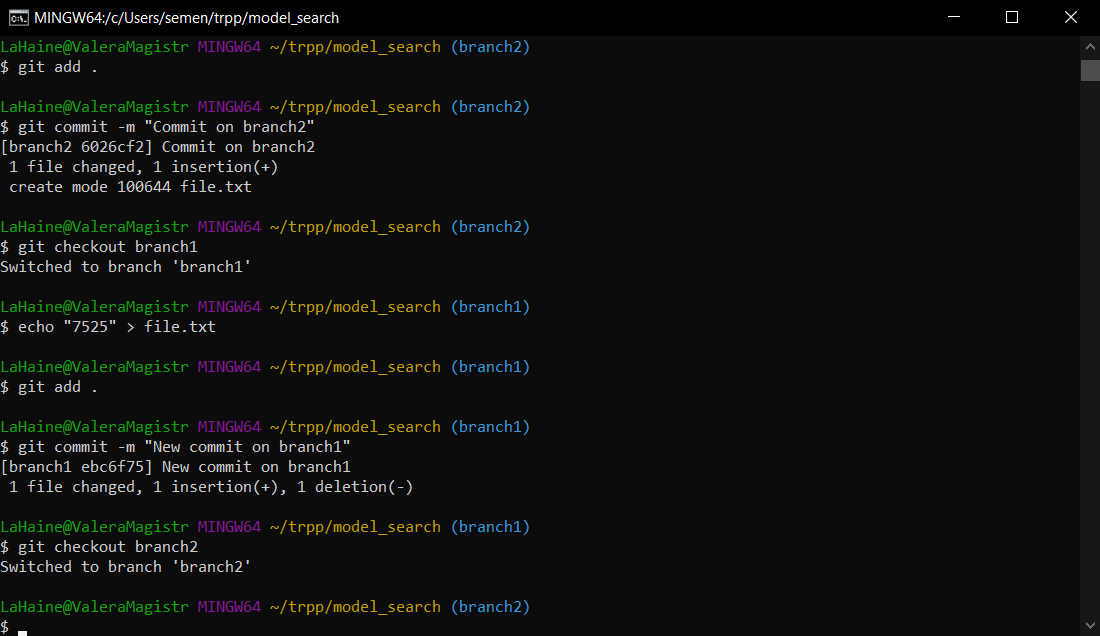


Рисунок 26 – Выполнение задания 4 третьей части

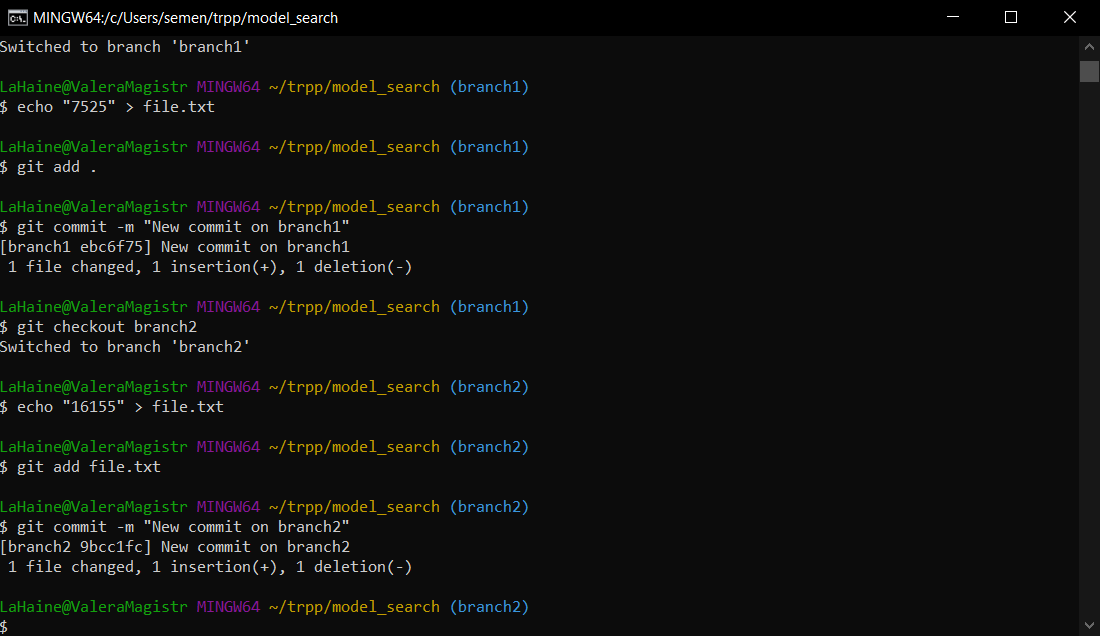


Рисунок 27 – Выполнение задания 4 третьей части

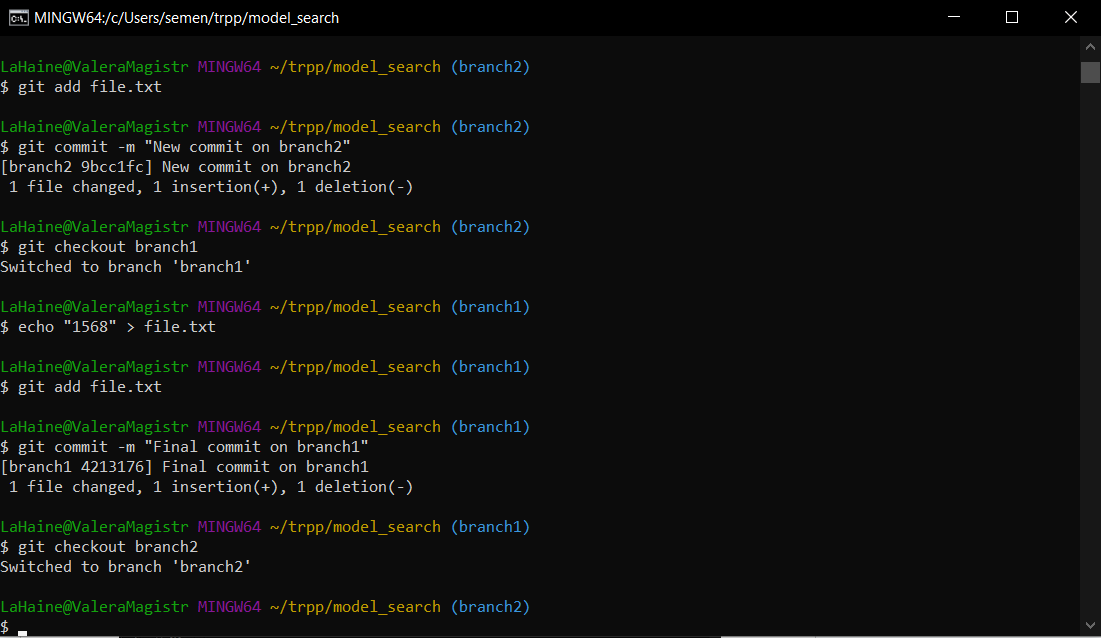


Рисунок 28 – Выполнение задания 4 третьей части

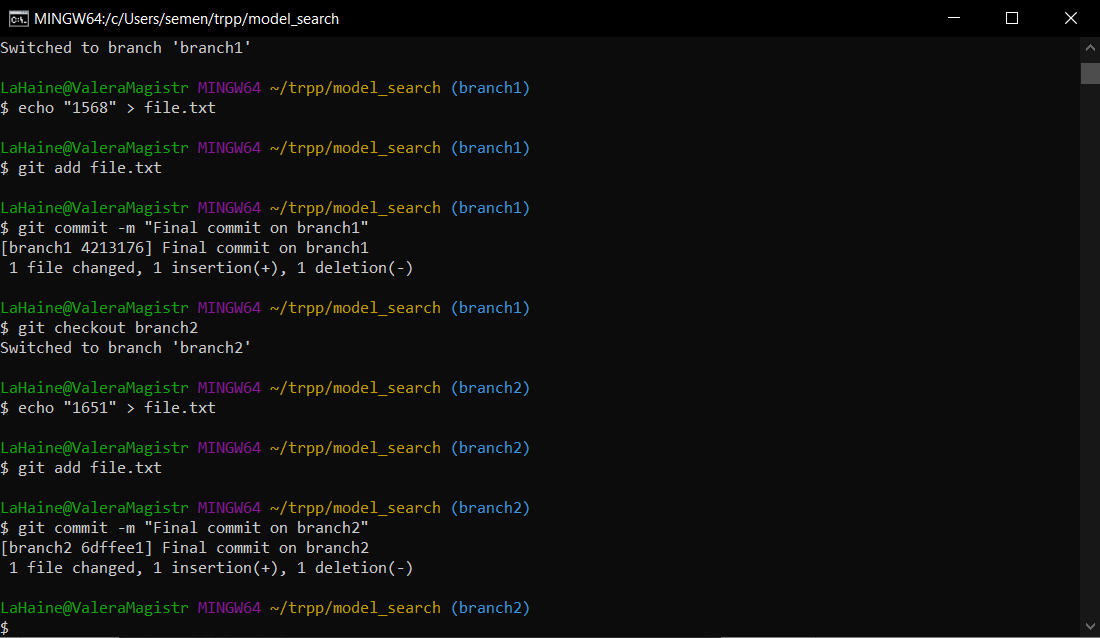


Рисунок 29 – Выполнение задания 4 третьей части

Слияние двух веток.

Исправление вручную ошибок при слиянии.

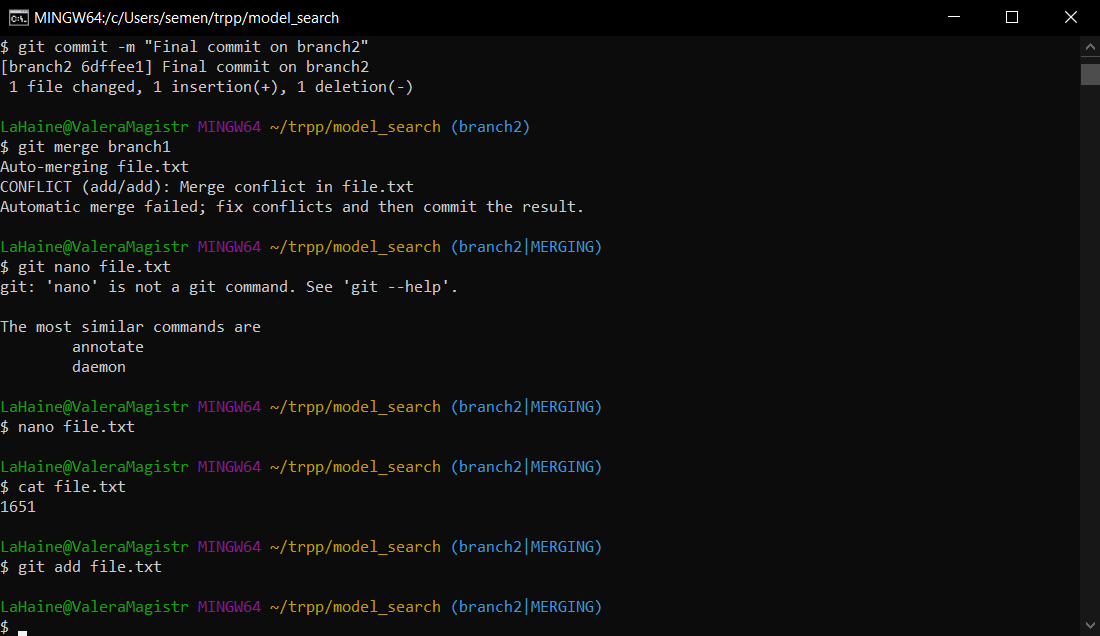


Рисунок 30 – Выполнение задания 5 третьей части



Рисунок 31 – Выполнение задания 5 третьей части

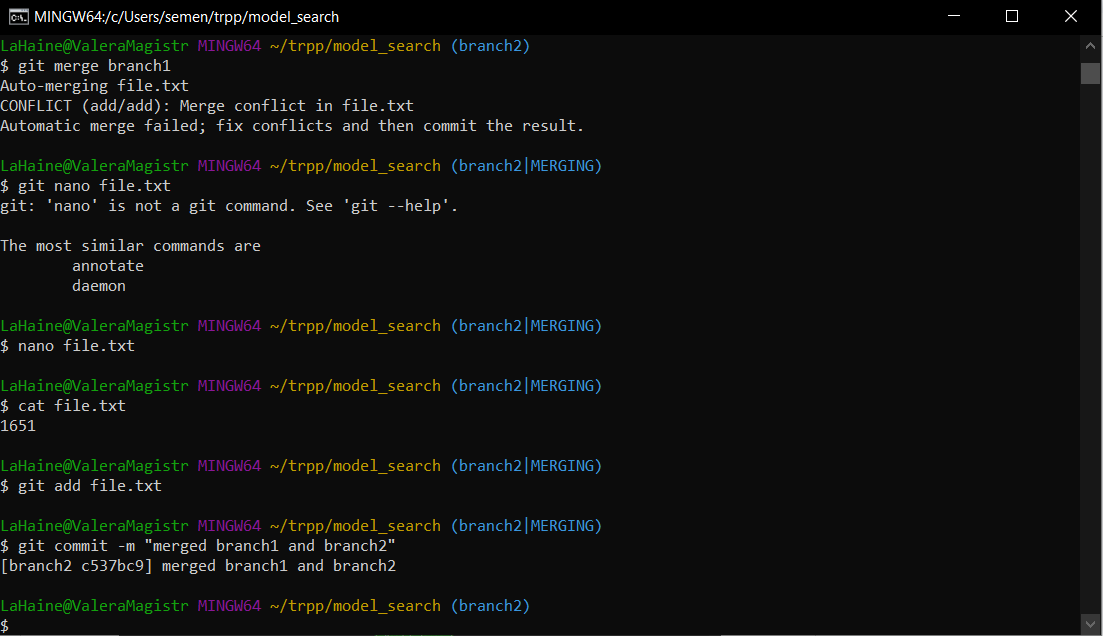


Рисунок 32 – Выполнение задания 5 третьей части

Выгрузка изменений из каждой ветки в удалённый репозиторий.

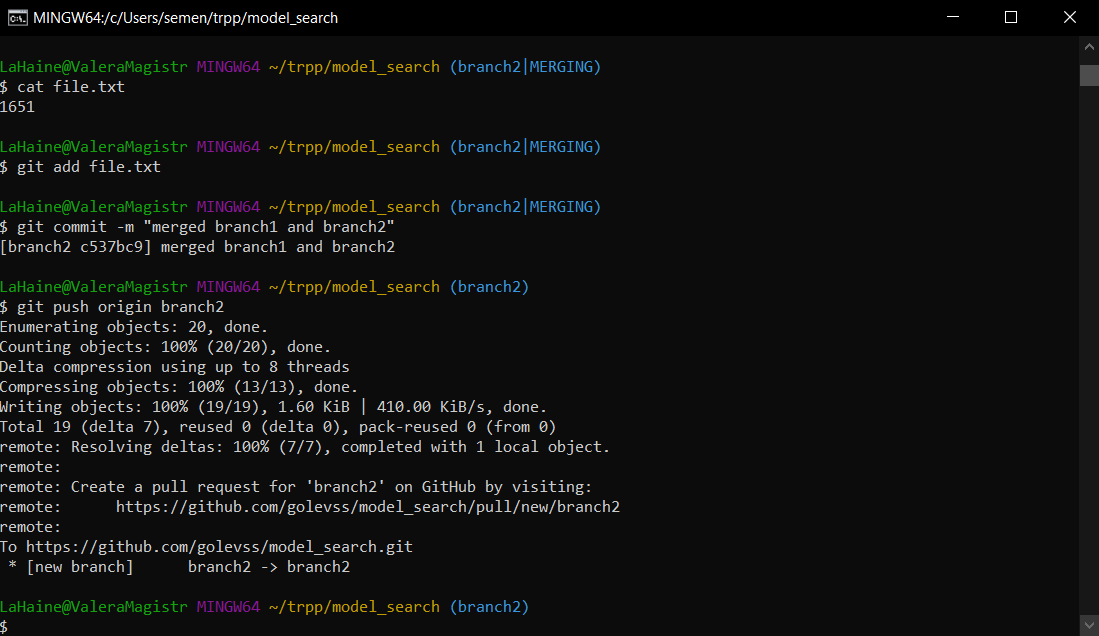


Рисунок 33 – Выполнение задания 6 третьей части

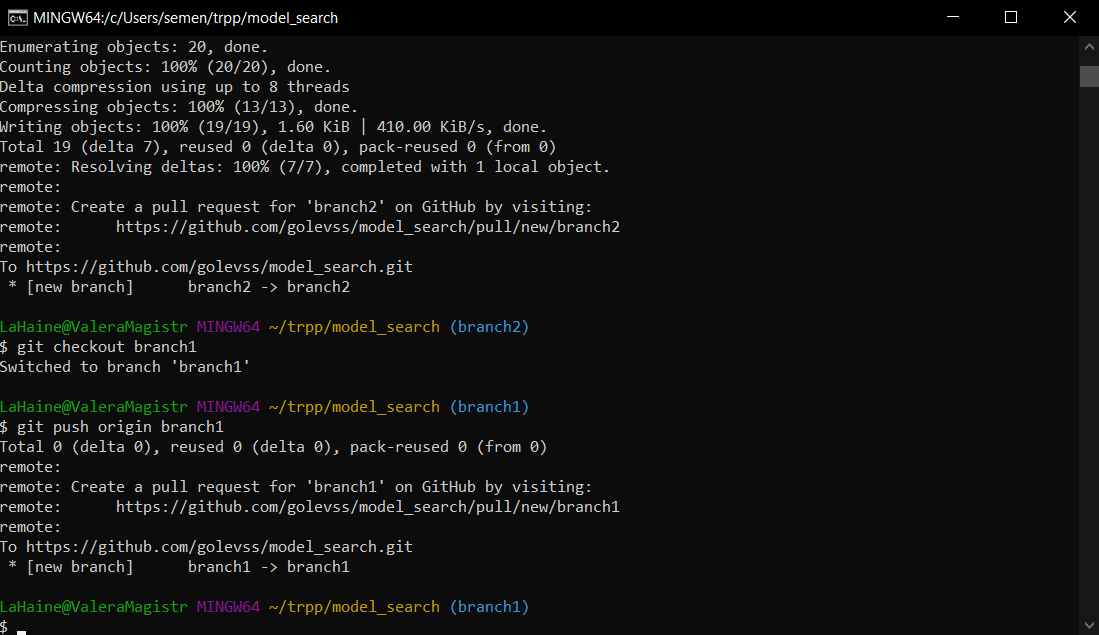


Рисунок 34 – Выполнение задания 6 третьей части

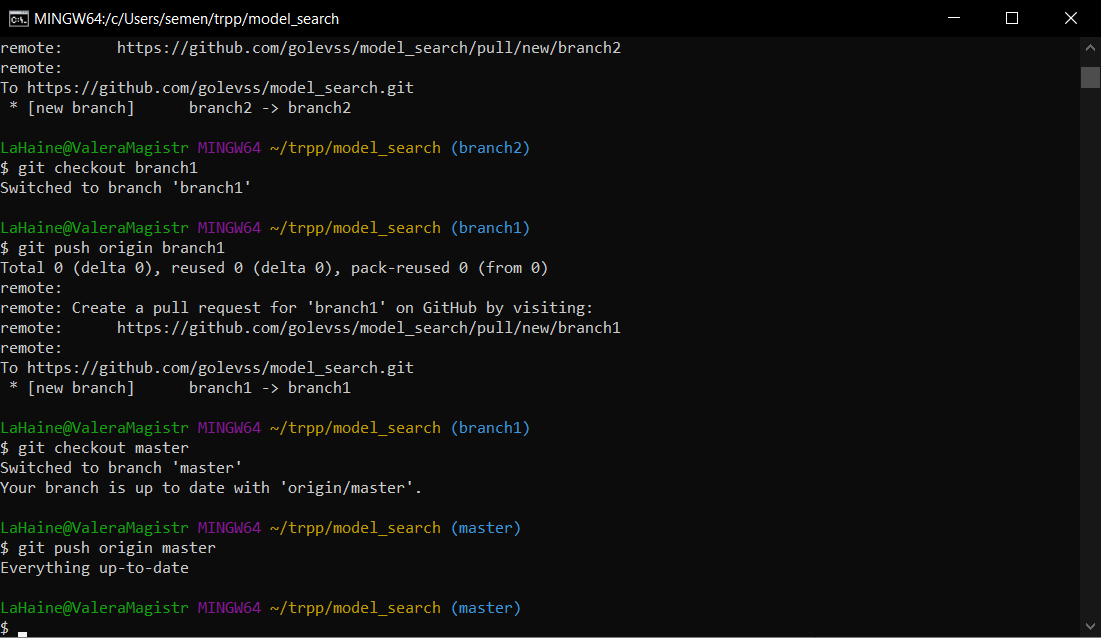


Рисунок 35 – Выполнение задания 6 третьей части

Повторное клонирование репозитория.



Рисунок 36 – Выполнение задания 8 третьей части

Три новых коммита.

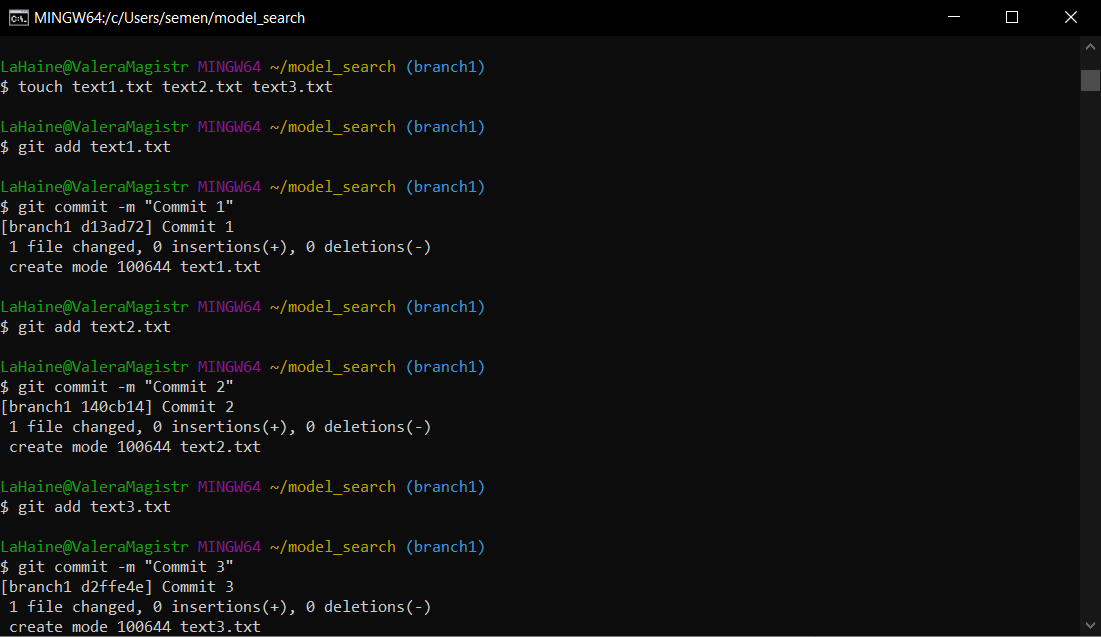


Рисунок 37 – Выполнение задания 9 третьей части

Выгрузка всех трёх изменений в новый репозиторий.

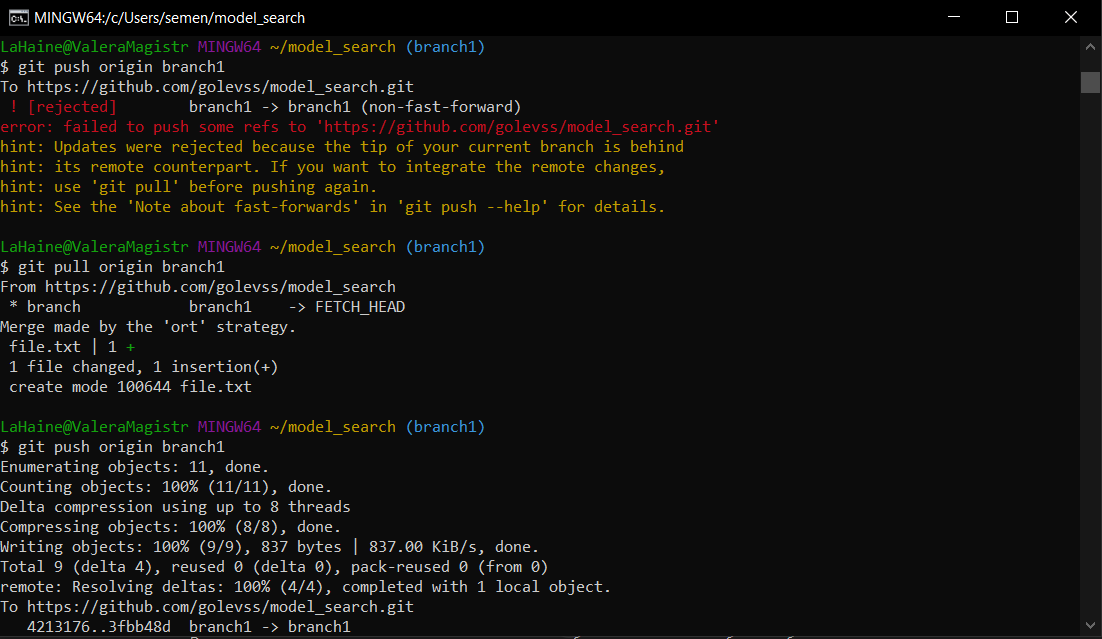


Рисунок 38 – Выполнение задания 10 третьей части

Возвращение в прошлый репозиторий, выгрузка изменений.

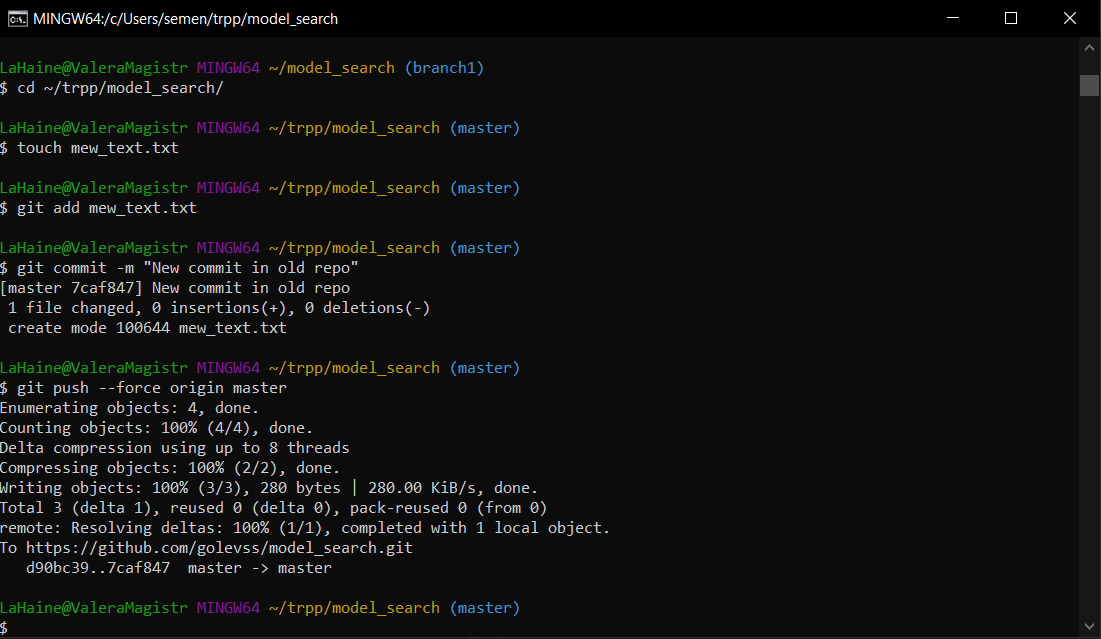


Рисунок 39 – Выполнение задания 11 третьей части

Получение этих изменений в новом репозитории.

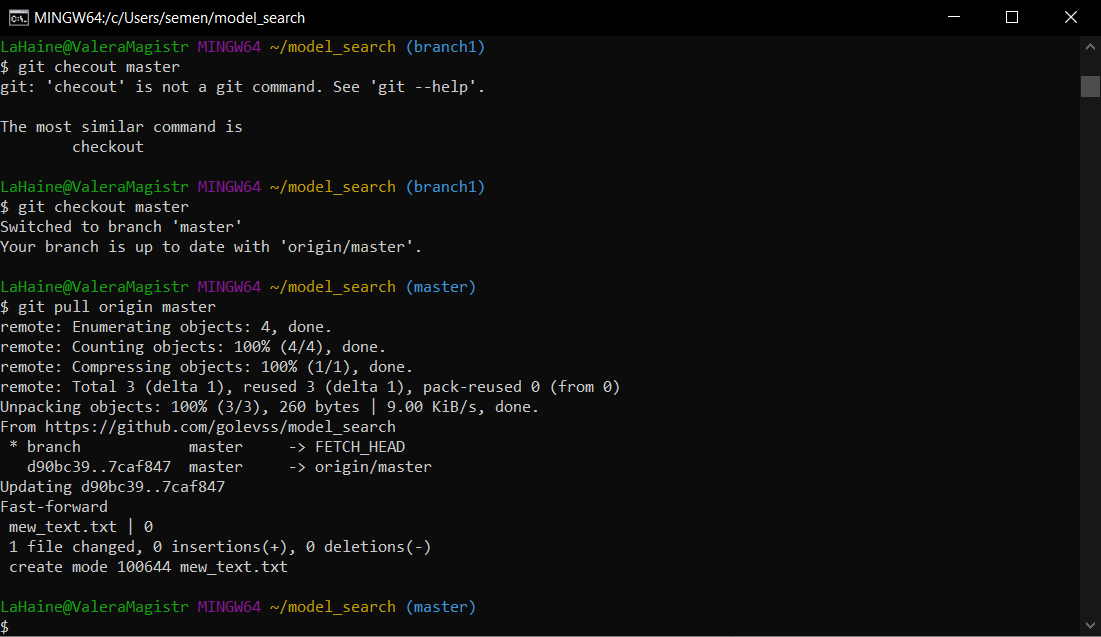


Рисунок 40 – Выполнение задания 12 третьей части

# КОНТРОЛЬНЫЕ ВОПРОСЫ

**4. Что такое коммит?**

Коммит — это операция в системе контроля версий (например, Git), которая сохраняет изменения в репозитории, фиксируя текущую версию файлов.

**7. Что такое слияние двух веток?**

Слияние двух веток (merge) — это процесс, при котором изменения из одной ветки (например, branch1) интегрируются в другую ветку (например, master или branch2). В Git это операция, которая объединяет истории двух веток, создавая новый коммит слияния.

**10. Для чего нужен .gitignore?**

Для указания git какие файлы не должны индексироваться.

**13. Что делает команда git log?**

Отображает историю коммитов, включаю информацию о этих коммитах. Также имеет множество видов отображения для корректного понимания истории коммитов.

**25. Как отменить действие команды "git add" на файл?**

Команда *git reset <название файла>* удаляет индексацию указанных файлов (то есть, отменяет добавление файла в коммит), но при этом не затрагивает сами изменения в рабочем каталоге.

**19. Чем отличаются команды "git push" и "git pull"?**

Команда *git push* отправляет изменения в удаленный репозиторий, а команда *git pull* принимает изменения из удалённого репозитория.

**15. Что делает команда git show?**

Команда git show <хэш коммита> используется для отображения подробной информации о конкретном коммите или объекте в Git.

**16. Что делает команда git stash?**

Команда git stash временно сохраняет все несохранённые изменения в рабочем каталоге, чтобы можно было переключиться на другую ветку или выполнить другую операцию, не теряя текущие наработки.

# ВЫВОД

В ходе данной работы была изучена система контроля версий. Изучены основные команды, работа с удалённым репозиторием, проблемы при слиянии веток, работа с несколькими ветками.