

**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технологический университет «СТАНКИН»**

**(ФГБОУ ВО «МГТУ «СТАНКИН»)**

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | Институт цифровых интеллектуальных систем | Кафедра  компьютерных систем управления | |

Дисциплина «Основы системного программного обеспечения»

**Отчет по лабораторной работе №1**

**Работа с системами контроля версий на примере Git Hub**

|  |  |  |
| --- | --- | --- |
| **Выполнила**  **студентка гр. АДБ-21-06:** | \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_  *(дата) (подпись)* | **Гулаков М.Р.** |
| **Проверил**  **к.т.н., доцент** | \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_  *(дата) (подпись)* | **Ковалев И.А.** |

**Москва 2024 г.**

Оглавление

[Цель работы 3](#_Toc167658783)

[Задачи работы 3](#_Toc167658784)

[Выполнение заданий 3](#_Toc167658785)

[Вывод 14](#_Toc167658786)

# Цель работы: получить навыки работы с системой контроля версий на примере Git Hub.

# Задачи работы

1. Изучить теоретический материал к лабораторной работе.
2. Проанализировать ход выполнения лабораторной работы.
3. Выполнить лабораторную работу в соответствии с методическими указаниями.

# Выполнение заданий

Создавать аккаунт на GtiHub не будем, уже есть.

Создадим новый репозиторий на GitHub для лабораторной №1 с названием «Lab1»:

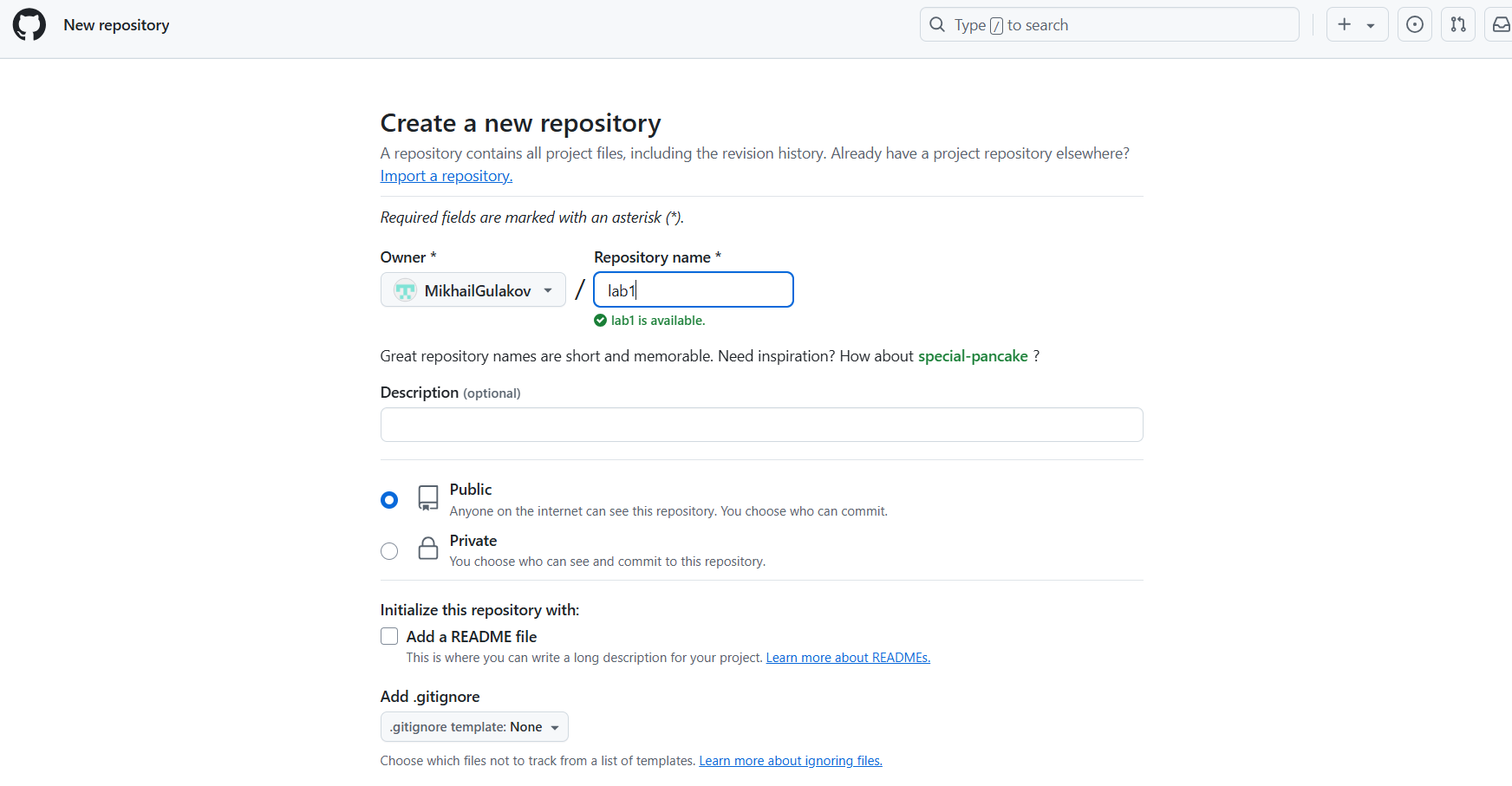


Рис. 1. Создание нового репозитория

Пропустим пункт «удаление предыдущей записи git на ПК», так как выполняем работу с личного ПК.

Создаём папку на диске со своей фамилией (локальный репозиторий):

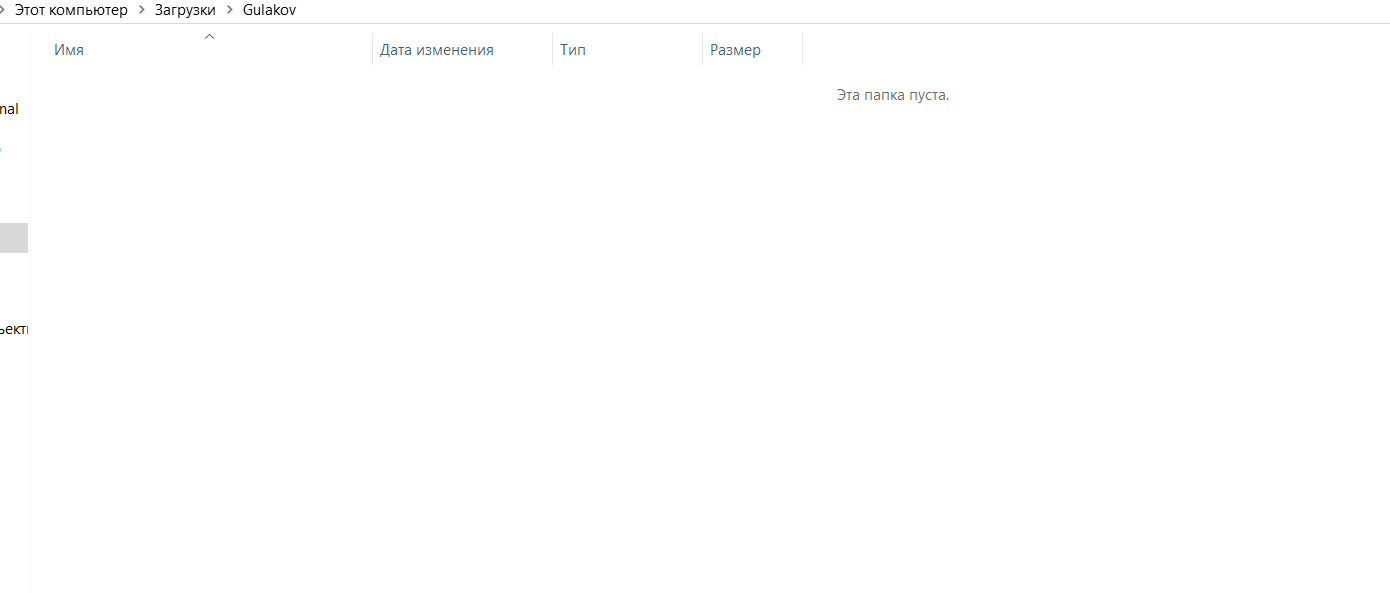


Рис. 2. Создание папки

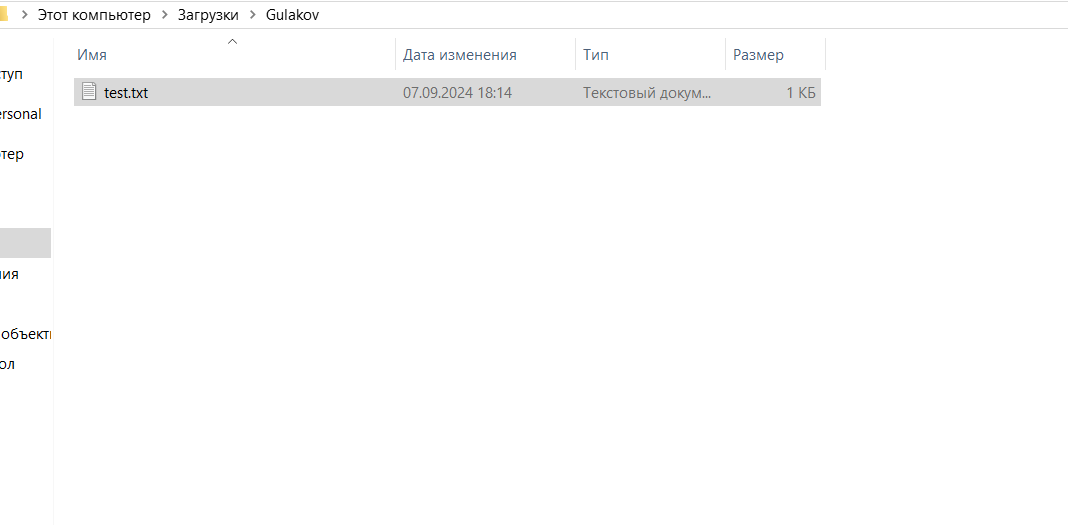


Рис. 3. Создание файла Test.txt

После того, как создали локальный репозиторий, переходим в командную строку для работы с этим репозиторием, настроим имя и электронную почту, а также проинициализируем эту папку как git репозиторий:

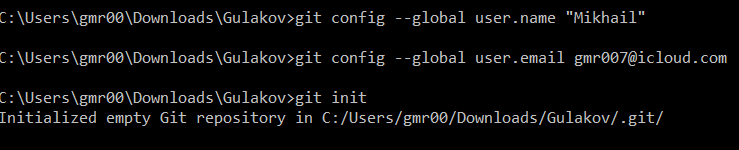


Рис. 4. Инициализация репозитория

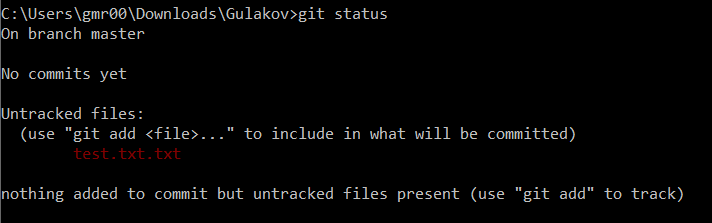


Рис. 5. Результат команды git status

Отображается сообщение, что есть новый файл, но он не отслеживается.

Это значит, что файл новый и система еще не знает, нужно ли следить за изменениями в файле или его можно просто игнорировать. Для того, чтобы начать отслеживать новый файл, нужно его специальным образом объявить.

Добавим его командой git add .:

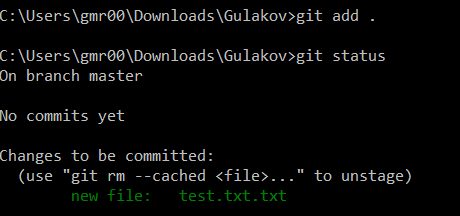


Рис. 6. Добавление файла для коммита

Теперь закомитим наши файлы. Коммит представляет собой состояние репозитория в определенный момент времени как некий опечаток во времени.

Коммит отмечается hash суммой и к которому мы можем в любой момент времени вернуться.

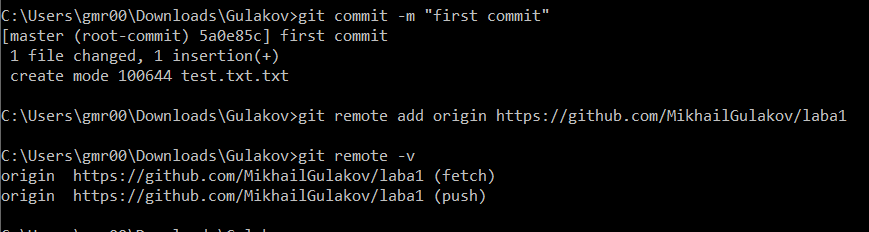


Рис. 7. Отправка коминат на сервер

Запросим изменения с сервера командой git pull origin master:

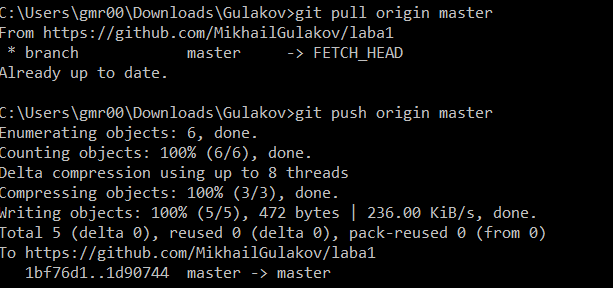


Рис. 8. Запрос изменения с сервера

Теперь перешлём локальный коммит на сервер командой git push origin master:

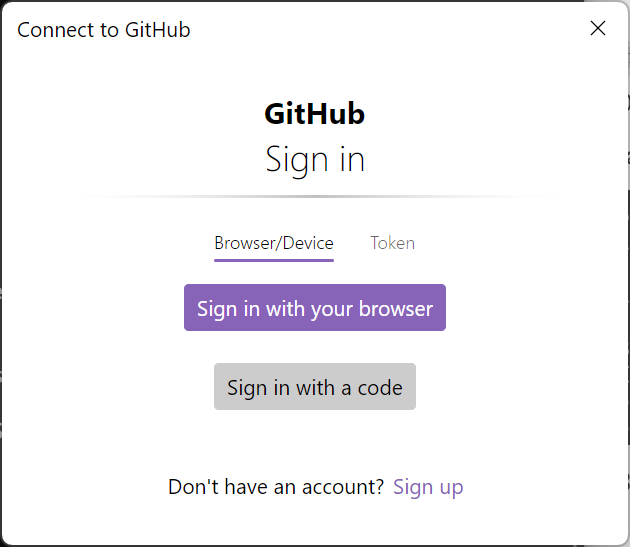


Рис. 9. Окно для авторизации на GitHub

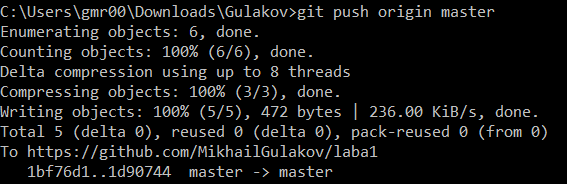


Рис. 10. Отправка изменений на сервер

Посмотрим наши изменения с использованием команды git log:

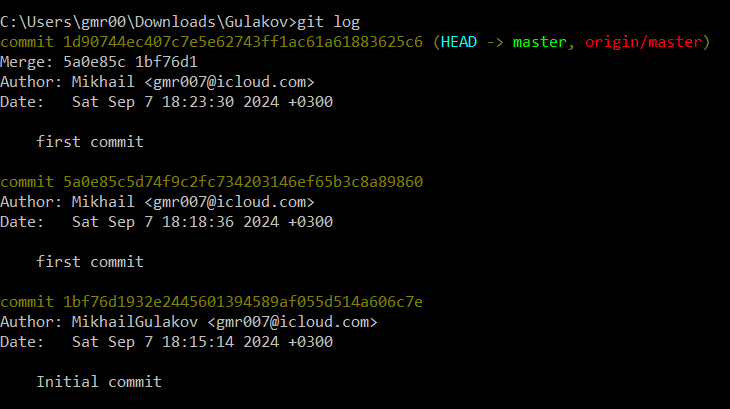


Рис. 11. Просмотр изменений

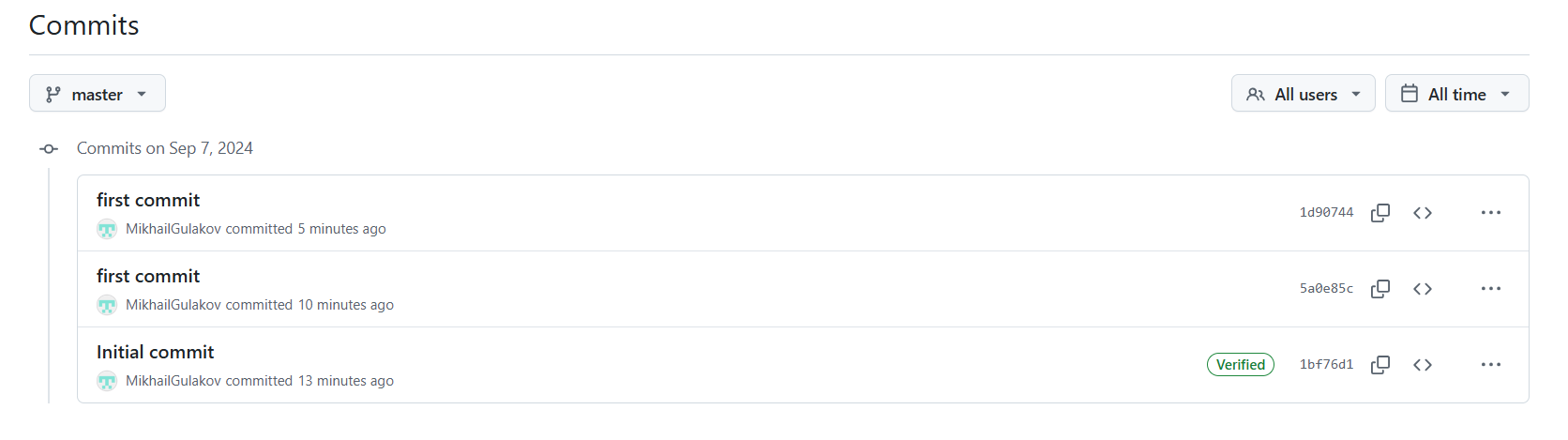


Рис. 12. Просмотр изменений на GitHub

Создание новой ветки

Во время разработки новой функциональности считается хорошей практикой работать с копией оригинального проекта, которую называют веткой. Ветви имеют свою собственную историю и изолированные друг от друга изменения до тех пор, пока мы не решим слить изменения вместе.

Создадим новую ветку командой git branch и переключимся на неё git checkout second:

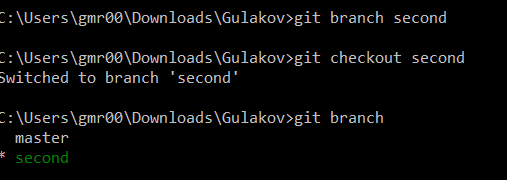


Рис. 13. Работа с ветвями

Создаем новый файл Test\_2.txt с фамилией, добавим в область подготовленных файлов, закомитим и отправим на сервер

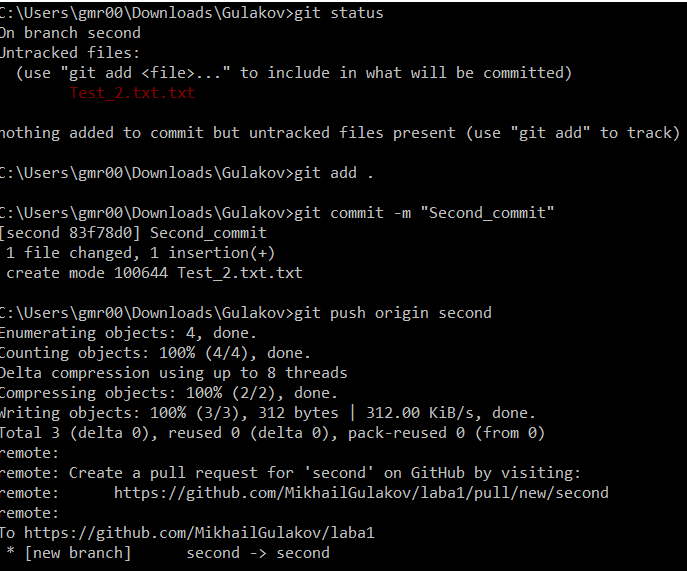


Рис. 14. Отправка нового коммита на сервер

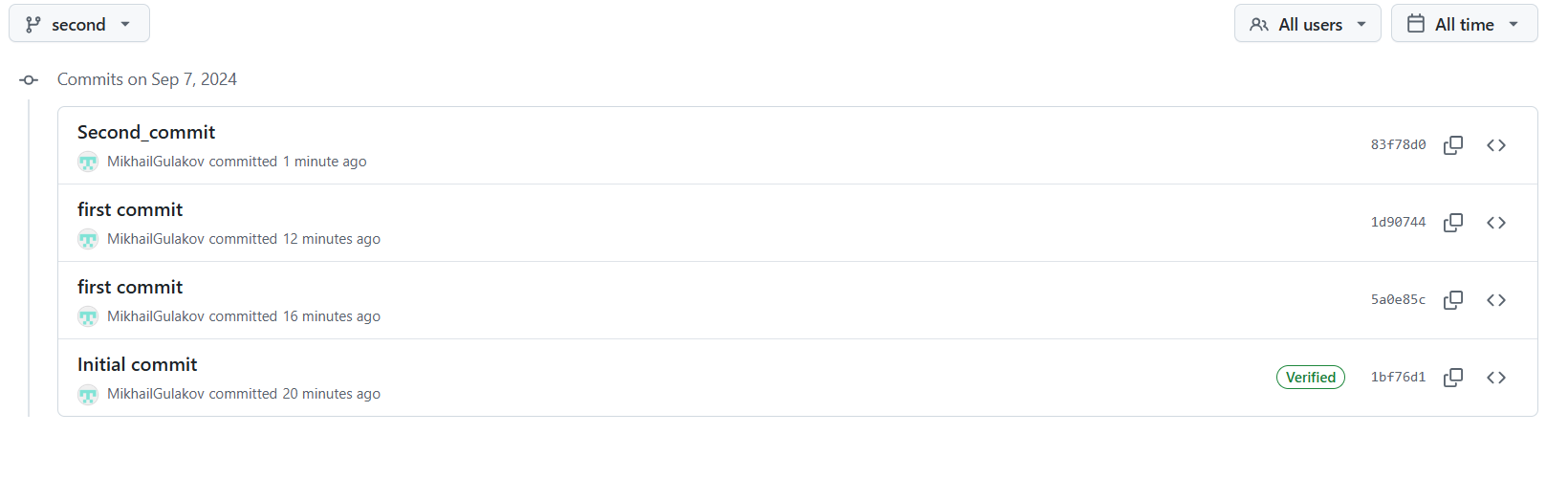


Рис. 15. Изменения на GitHub

Сделаем слияние веток

Для начала переключимся на ветку master, воспользуемся командой merge для объединения веток и удалим ветку second.

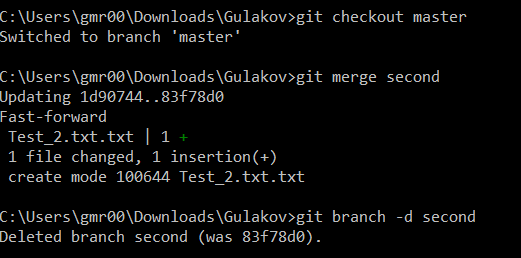


Рис. 16. Выполнение объединения

Немного усложним задание и предположим, что есть одинаковый файл на двух разных ветках.

Создадим ветку newdev, переключимся на неё:

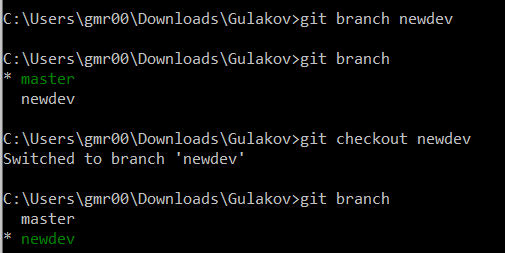


Рис. 17. Создание ветки newdev

Добавим в наш исходный файл отчество и закоммитим изменения:

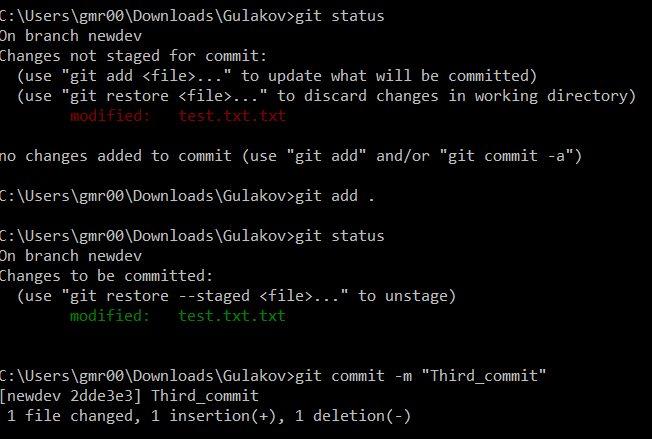


Рис. 18. Изменение исходного файла Test.txt

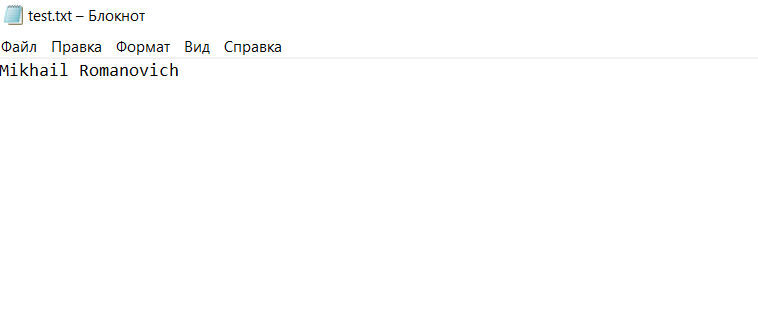


Рис. 19. Содержимое файла Test.txt

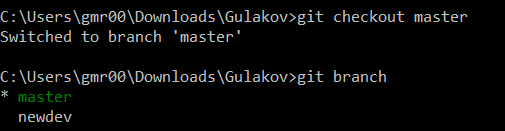


Рис. 20. Переключение на ветку master

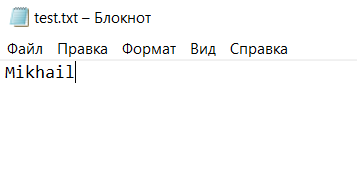


Рис. 21. Содержимое файла Test.txt

Объединим ветки master и newdev:

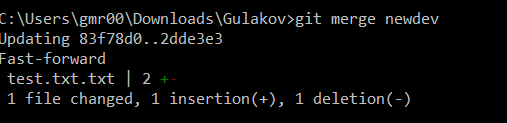


Рис. 22. Слияние веток

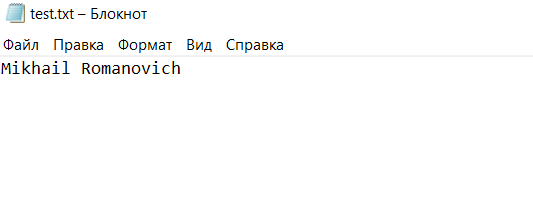
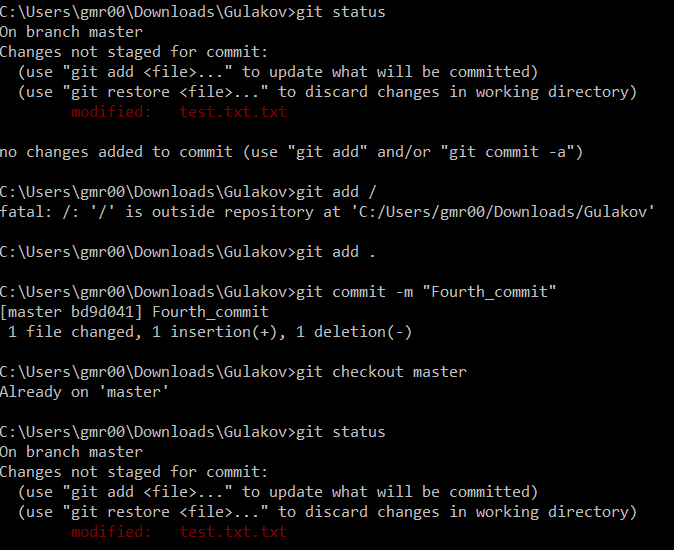


Рис. 23. Проверка изменений в файле после слияния веток

Сделаем теперь изменения в файле в двух ветках и попробуем объединить их после этого:



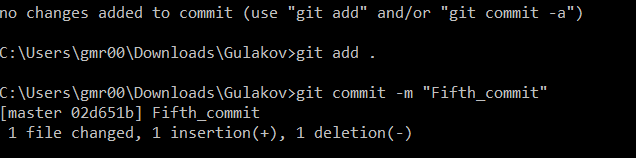


Рис. 24. Результат изменения в файле на каждой ветке и их слияния

Разрешим данный конфликт

Наберем команду git diff

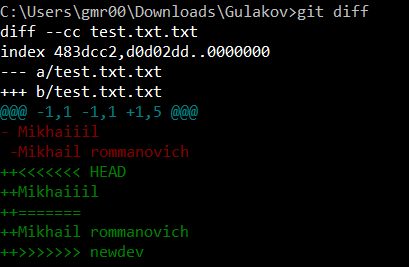


Рис. 25. Выполнение git diff

Исправим конфликт в нашем файле и отправим изменения на сервер:

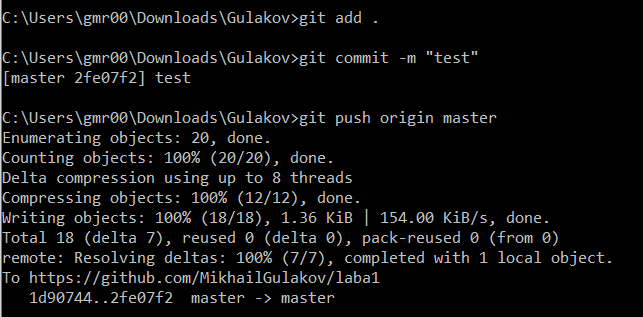


Рис. 26. Сохранение изменений и отправка на сервер

В конце удалим созданные нами ветки newdev и second

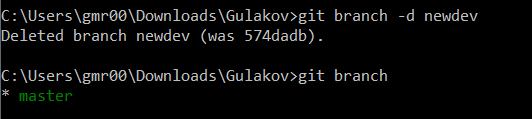
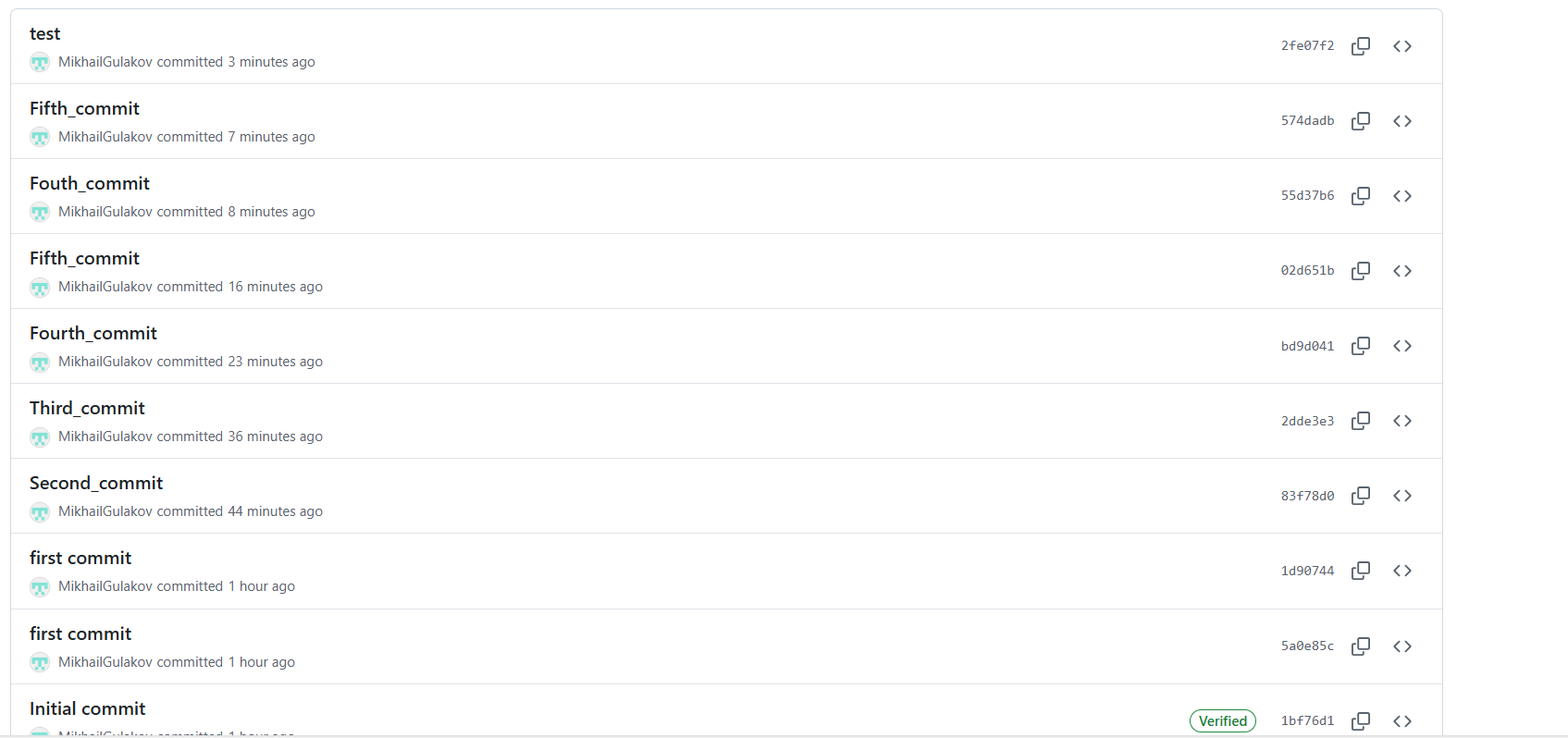
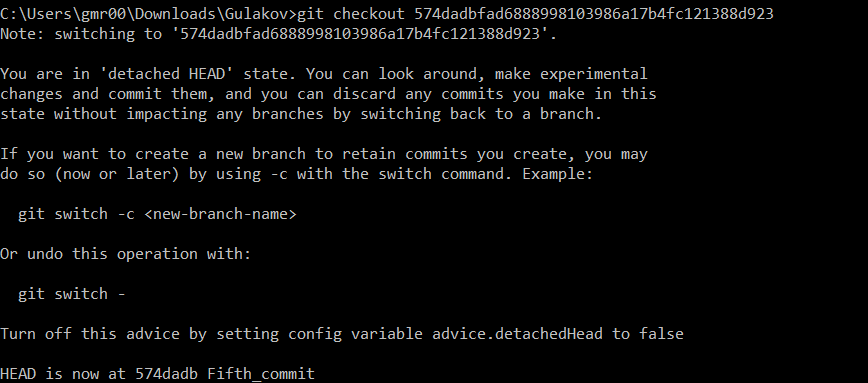
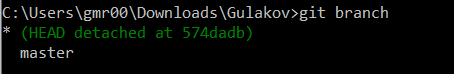
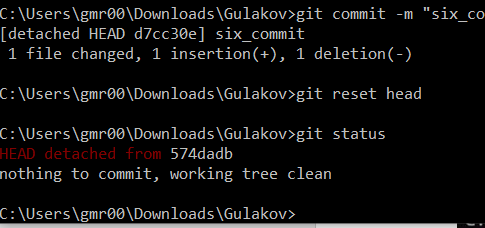


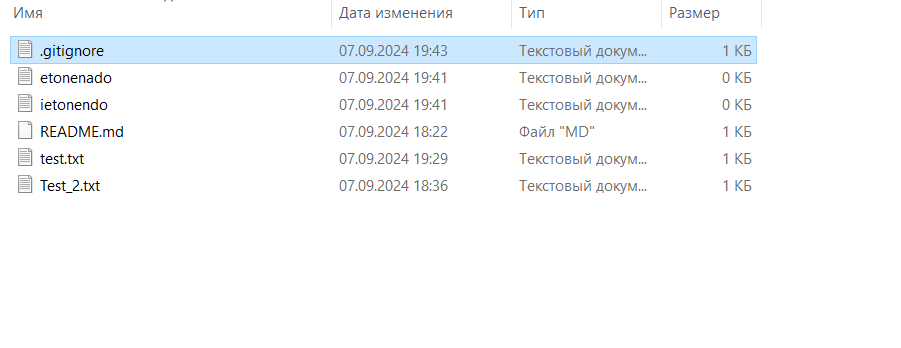
Рис. 27. Просмотр доступных веток

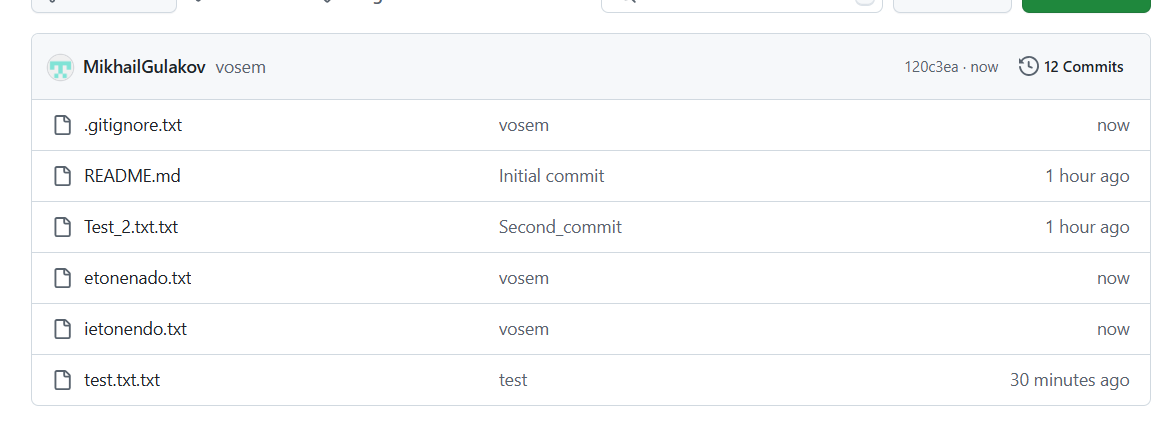












# Вывод: в ходе выполнения лабораторной работы были получены навыки по работе с системой контроля версий на примете Git Hub: фиксация изменений файлов; создание и отправка коммитов на сервер; создание новых веток и их слияние; разрешение конфликтов после слияния двух веток, где изменялись одинаковые файлы разными разработчиками.

**Список использованной литературы:**

1. «Большая книга CSS3», Дэвид Макфарланд. Изд.: Питер, 2016.

2. «JavaScript», Дэвид Макфарланд. Изд.: Символ-Плюс, 2013.

3. «HTML, XHTML and CSS», Andy Harris. Изд.: Машиностроение, 2012.