Отчет по лабораторной работе №1

Использование git

Ильин Никита Евгеньевич

2022 Feb 10th

Содержание

[Цель работы 1](#_Toc95568015)

[Задание 1](#_Toc95568016)

[Теоретическое введение 2](#_Toc95568017)

[Выполнение лабораторной работы 3](#_Toc95568018)

[Выводы 28](#_Toc95568019)

# Цель работы

Цель данной работы — Изучение основ работы с системой контроля версий git.

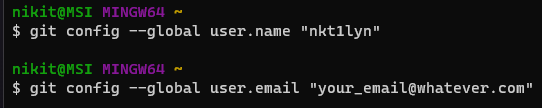
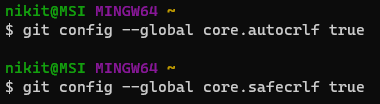
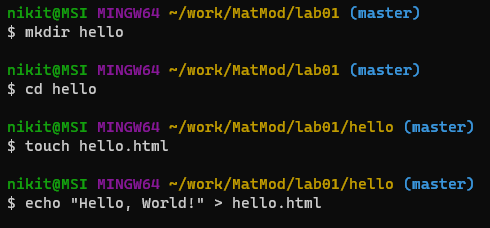
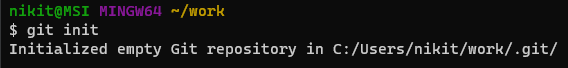
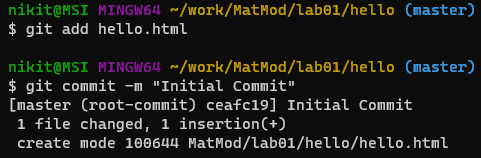
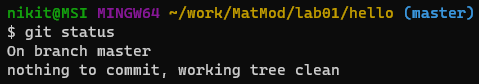
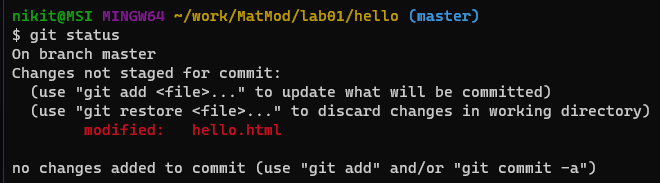
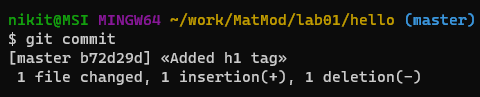
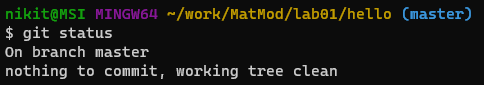
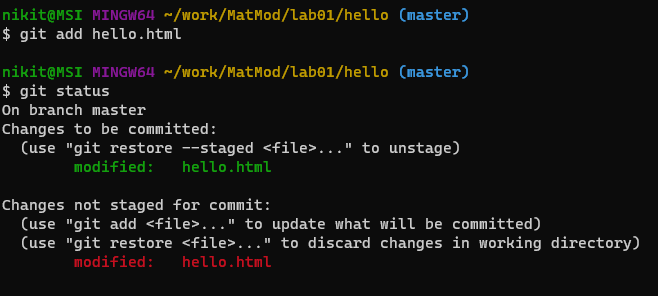
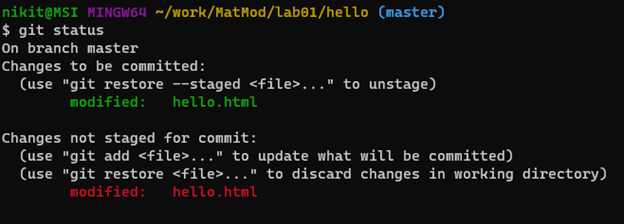
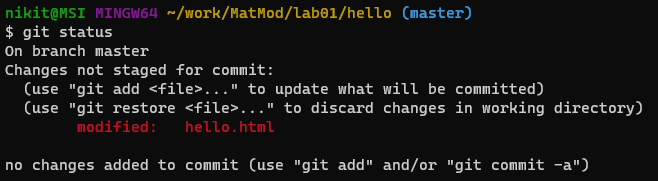
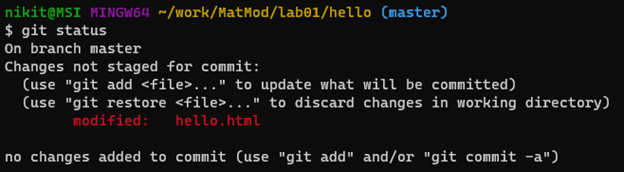
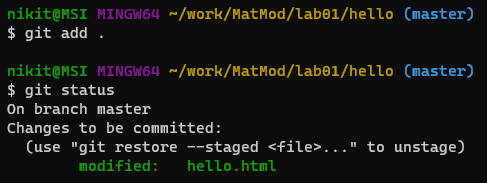
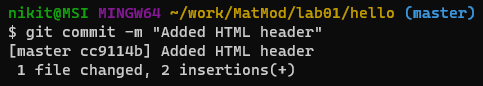
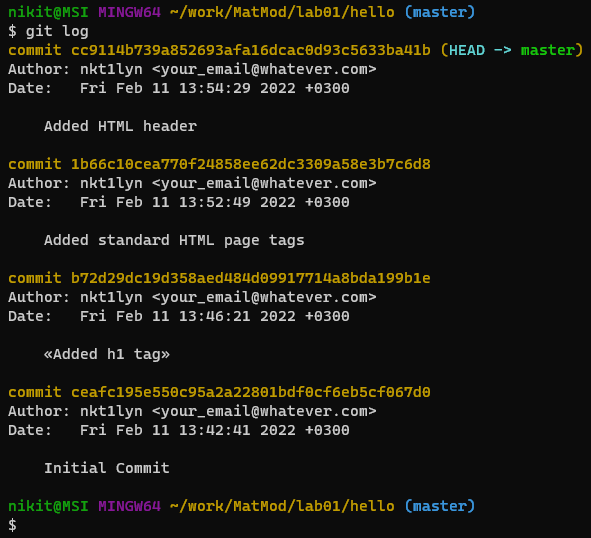
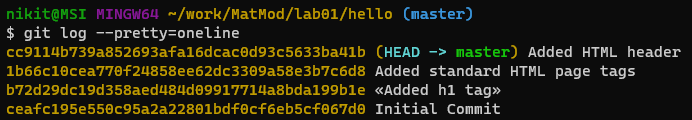
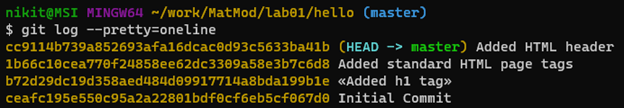
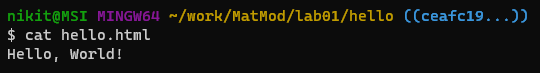
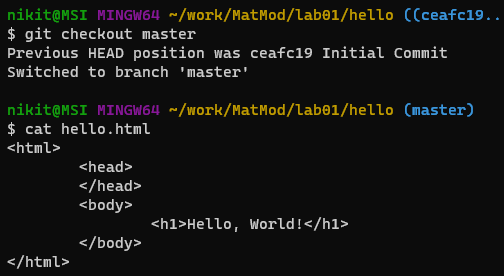
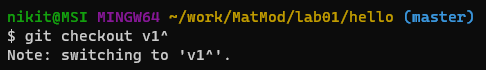
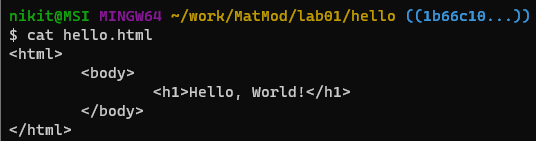
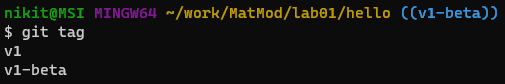
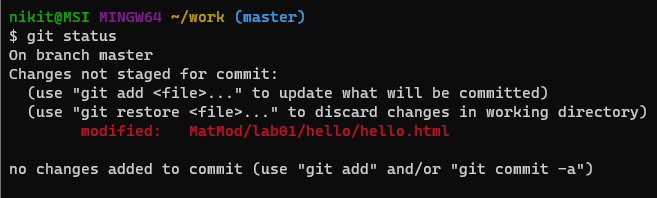
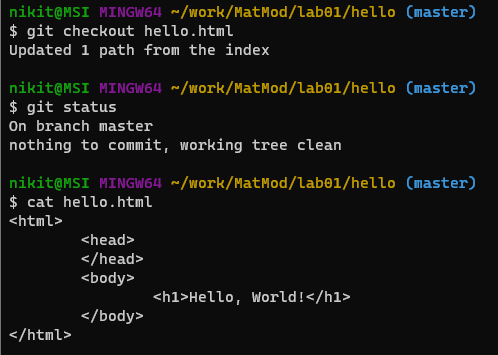
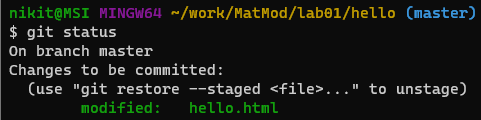
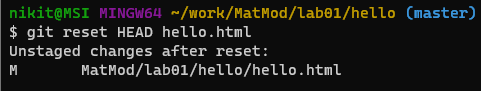
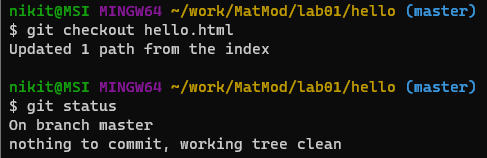
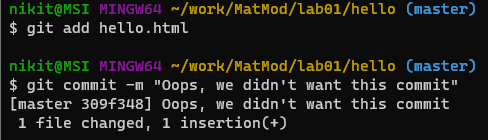
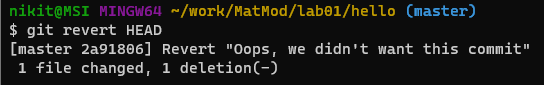
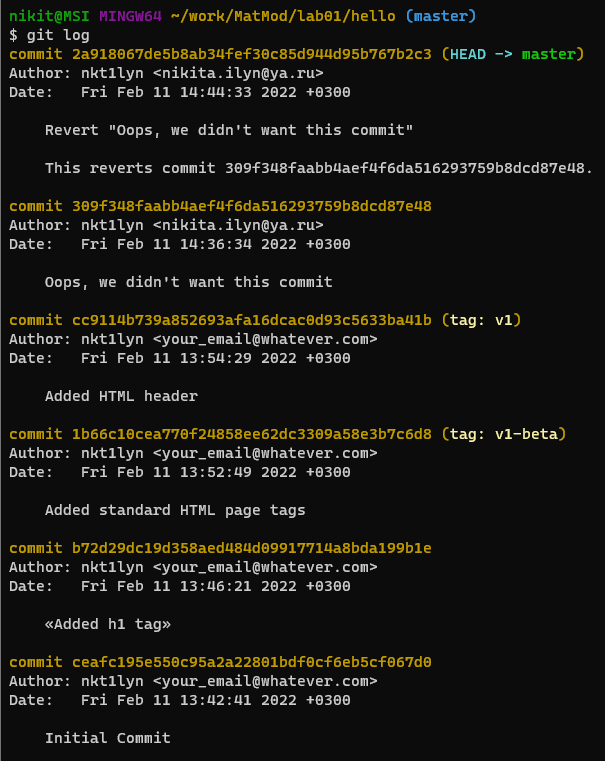
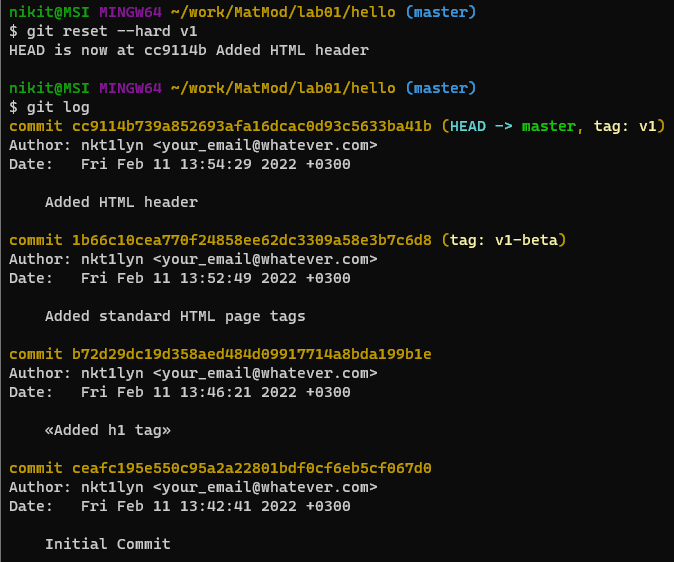
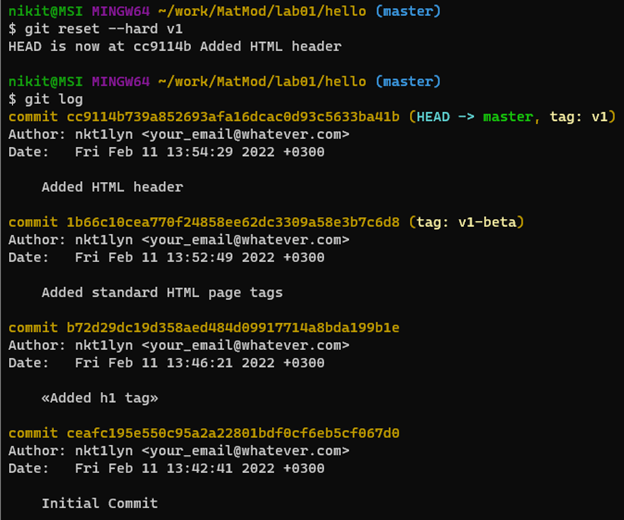
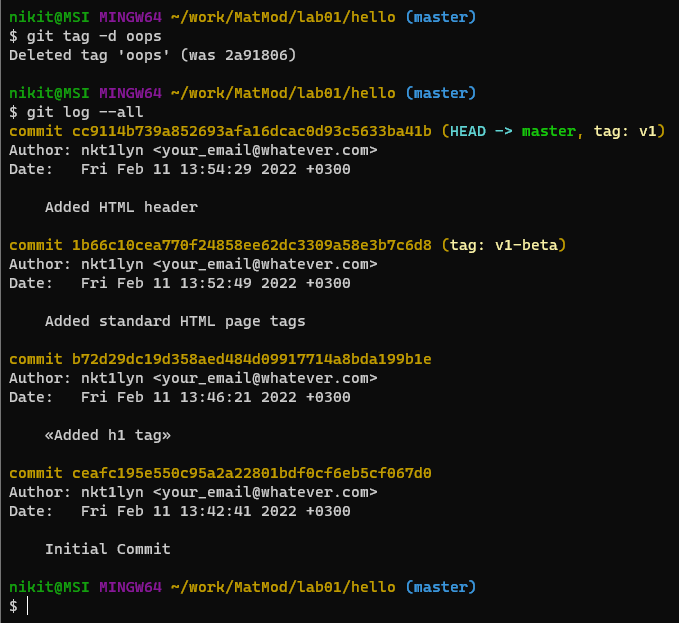
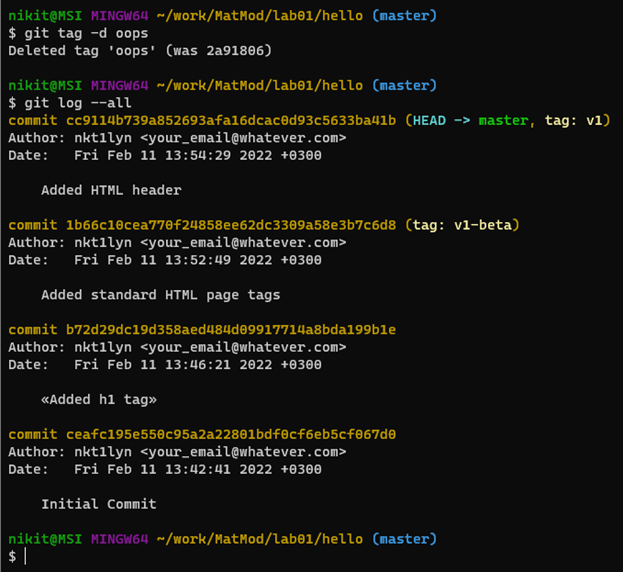
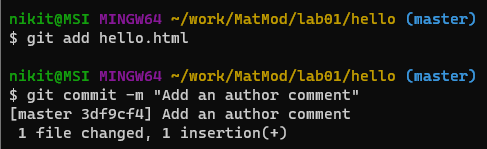
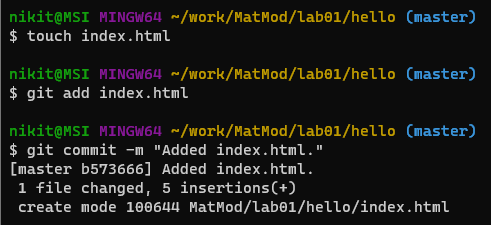
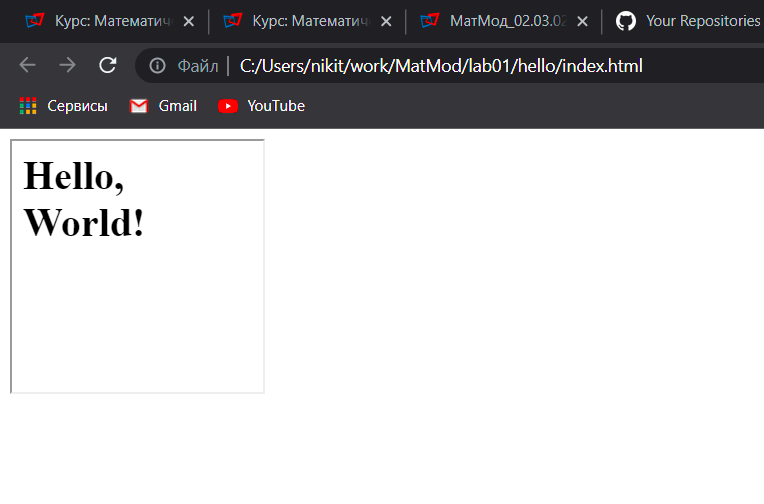
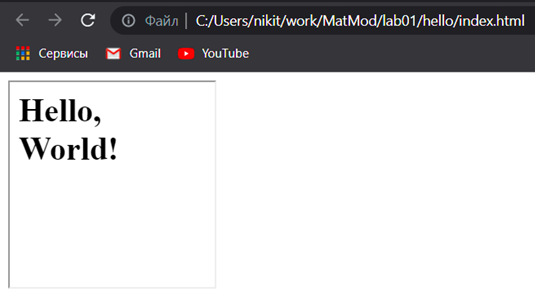
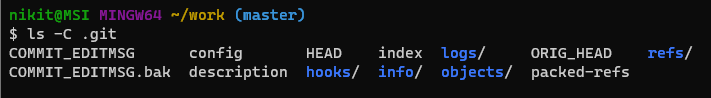
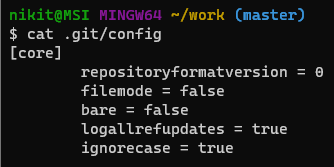
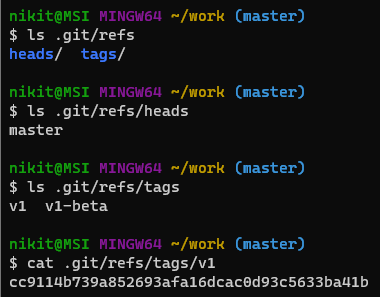
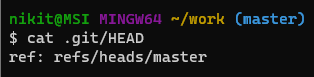
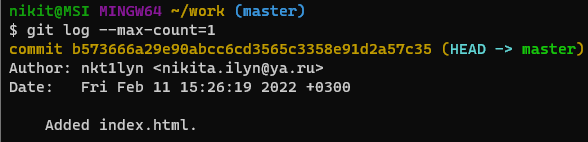
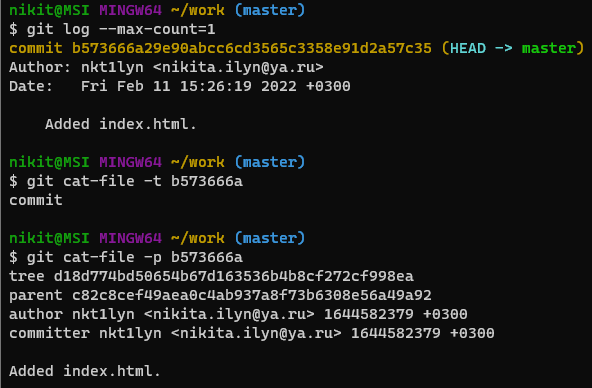
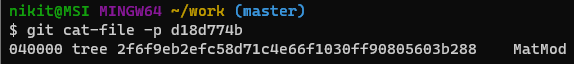
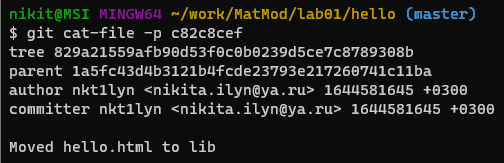
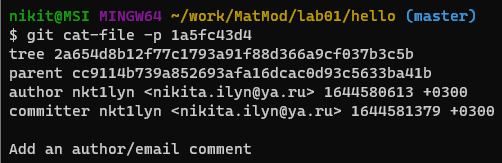
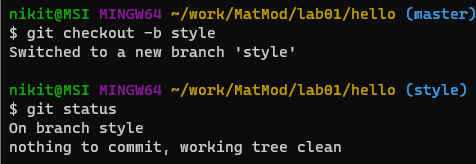
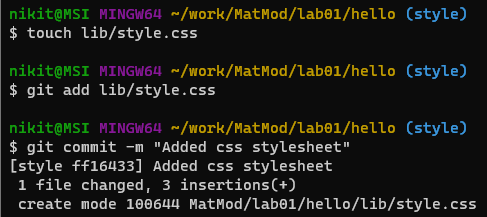
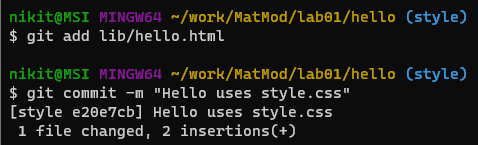
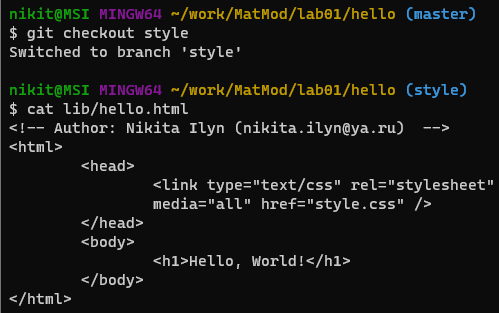
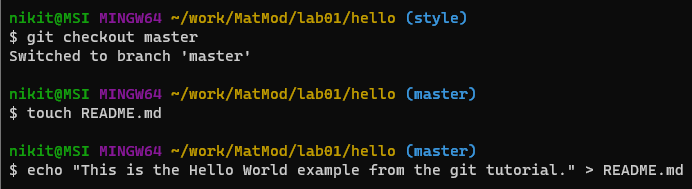
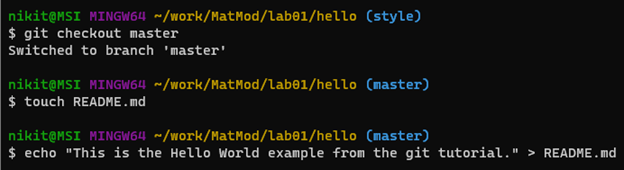
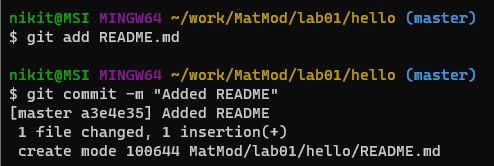
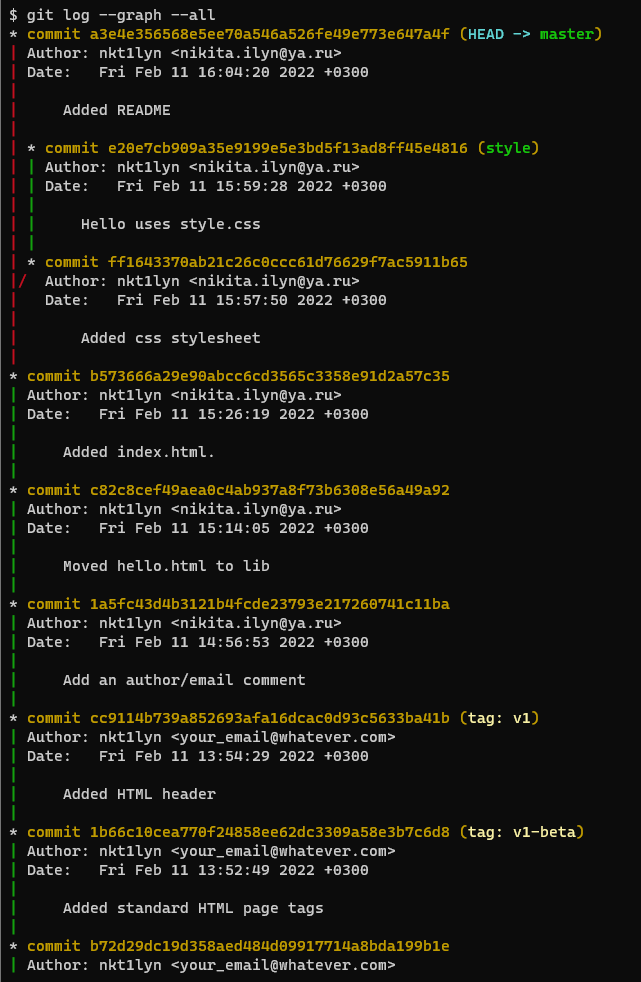
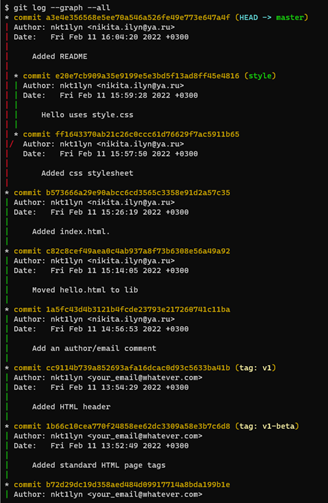
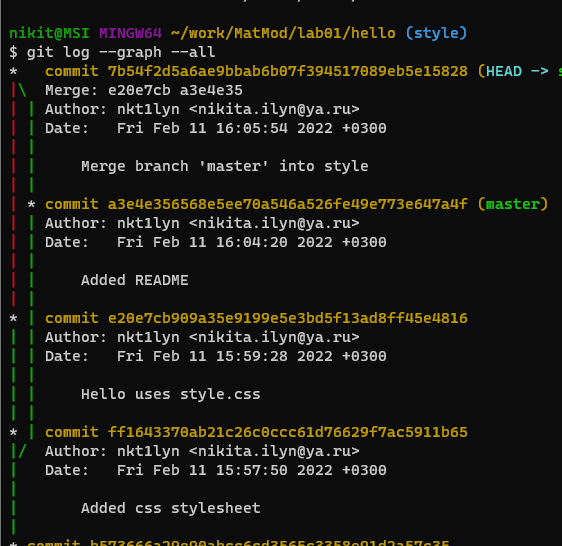
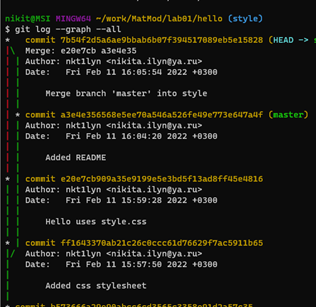
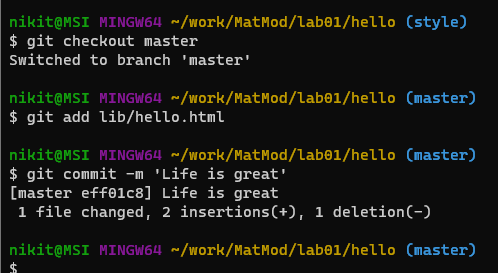
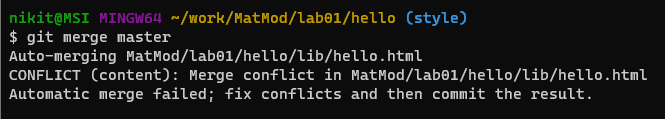
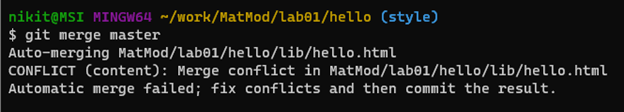
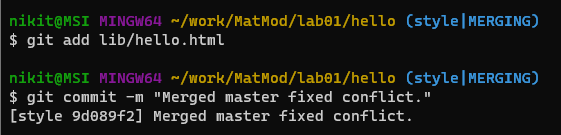
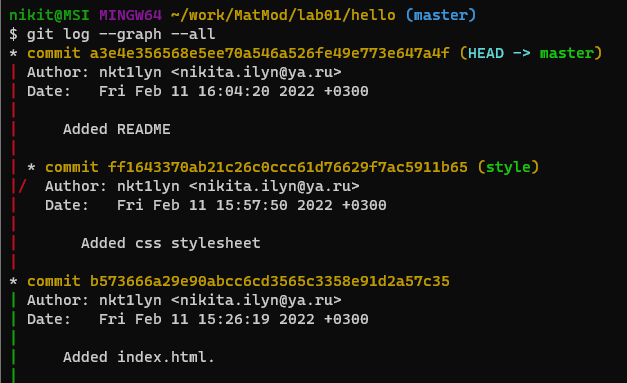
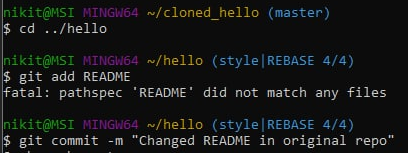
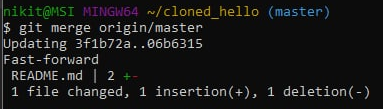
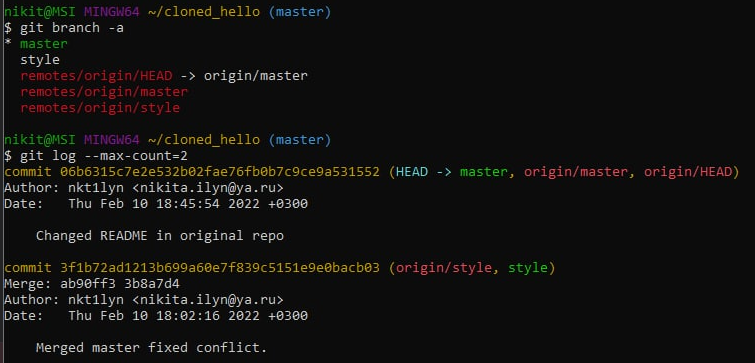
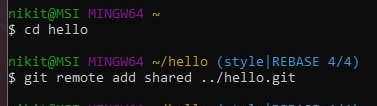
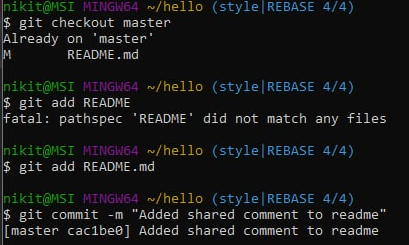
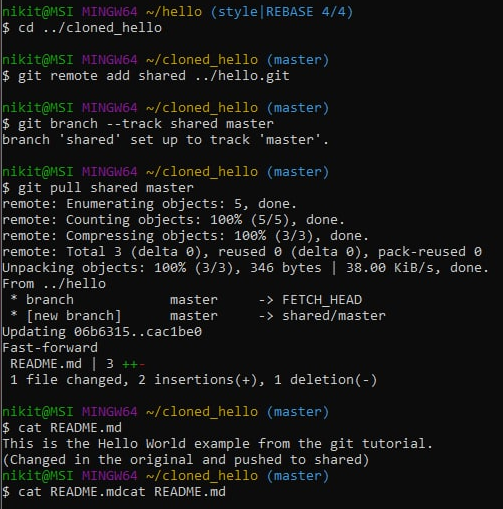
# Задание

| Номер задачи | Описание задачи |
| --- | --- |
| 1.1 | Подготовка к работе с git. |
| 1.2 | Создание проекта. |
| 1.3 | Внесение изменений в файлы. |
| 1.4 | Индексация изменений. |
| 1.5 | Отмена локльных изменений (до индексации). |
| 1.6 | Отмена проиндексированных изменений. |
| 1.7 | Отмена коммитов. |
| 1.8 | Удаление коммитов из ветки. |
| 1.9 | Удаление тега oops. |
| 1.10 | Внесение изменений в коммиты. |
| 1.11 | Перемещение файлов. |
| 1.12 | Второй способо перемещения файлов. |
| 1.13 | Подробнее о структуре. |
| 1.14 | Git внутри: Каталог .git. |
| 1.15 | Работа непосредственно с объектами git. |
| 1.16 | Создание ветки. |
| 1.17 | Навигация по веткам. |
| 1.18 | Изменения в ветке master. |
| 1.19 | Сделайте коммит изменений README.md в ветку master. |
| 1.20 | Слияние. |
| 1.21 | Создание конфликта. |
| 1.22 | Разрешение конфликтов. |
| 1.23 | Сброс ветки style. |
| 1.24 | Сброс ветки master. |
| 1.25 | Перебазирование. |
| 1.26 | Слияние в ветку master. |
| 1.27 | Клонирование репозиториев. |
| 1.28 | Просмотр клонированного репозитория. |
| 1.29 | Что такое origin? |
| 1.30 | Удаленные ветки. |
| 1.31 | Изменение оригинального репозитория. |
| 1.32 | Слияние извлеченных изменений. |
| 1.33 | Добавление ветки наблюдения. |
| 1.34 | Чистые репозитории. |
| 1.35 | Создайте чистый репозиторий. |
| 1.36 | Добавление удаленного репозитория. |
| 1.37 | Отправка изменений. |
| 1.38 | Извлечение общих изменений. |

# Теоретическое введение

| Имя каталога | Описание каталога |
| --- | --- |
| / | Корневая директория, содержащая всю файловую |
| /bin | Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям |
| /etc | Общесистемные конфигурационные файлы и файлы конфигурации установленных программ |
| /home | Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя |
| /media | Точки монтирования для сменных носителей |
| /root | Домашняя директория пользователя root |
| /tmp | Временные файлы |
| /usr | Вторичная иерархия для данных пользователя |

# Выполнение лабораторной работы

1. Указываем имя и электронную почту для git  
   
2. Устанавливаем параметры установки окончаний строк  
   
3. Устанавливаем отображение unicode  
   рис. 3
4. Создаем страницу Hello World  
   
5. Создаем репозиторий  
   
6. Добавление файла в репозиторий  
   
7. Проверяем состояние репозитория  
   
8. Изменяем страницу Hello World  
   
9. Индексируем изменения  
   
10. Коммит изменений  
      
    
11. Добавляем стандартные теги страницы  
    рис. 12  
      
      
      
    
12. История изменений  
      
    
13. Получение старых версий  
      
    
14. Создание тегов версий  
      
      
      
    рис. 24
15. Переключение по имени тега  
      
    
16. Просмотр тегов с помощью команды tag  
    рис. 27
17. Переключаемся на ветку master
18. Изменяем файл hello.html
19. Проверяем состояние  
    рис. 28
20. Отменяем изменения в рабочем каталоге  
    рис. 29
21. Измененяем файл и индексируем изменения  
    
22. Проверьте состояние  
    
23. Выполняем сброс буферной зоны  
    
24. Переключаемся на версию коммита  
    рис. 33
25. Отменяем коммит  
    
26. Изменяем файл и делаем коммит  
    
27. Делаем коммит с новыми изменениями, отменяя предыдущие  
    
28. Проверяем лог  
    
29. Выполняем команду git reset
30. Проверяем историю  
    
31. Отмечаем ветку  
    
32. Делаем сброс коммитов к предшествующим коммиту Oops  
    рис. 40
33. Удаление тега Oops  
    
34. Изменяеем страницу и делаем коммит  
    
35. Изменяем предыдущий коммит  
    
36. Перемещаем файл hello.html в каталог lib  
    
37. Делаем коммит в новый каталог  
    
38. Добавляем файл Index.html  
     
39. Каталог .git  
    
40. Углубляемся в базу данных объектов  
    
41. Открываем Config файл  
    
42. Ветки и теги  
    
43. Открываем файл HEAD  
    
44. Поиск последнего коммита  
    
45. Вывод коммита с помощью SHA1 хэша  
    
46. Поиск дерева  
    
47. Вывод каталога lib  
    
48. Вывод файла hello.html  
    
49. Создаем ветку  
    
50. Добавляем файл стилей style.css  
    
51. Изменяем файл index.html  
    
52. Переключаемся на ветку master  
    
53. Возвращаемся к ветке style  
    
54. Создаем файл README в ветке master  
    
55. Делаем коммит изменений  
    
56. Просматриваем текущие ветки  
    
57. Выполняем слияние веток  
    
58. Создаем конфликт  
    
59. Выполняем слияние веток master и style, и решаем конфликт  
    
60. Делаем коммит решения конфликта  
    
61. Сброс ветки style. Проверяем ветку  
    
62. Сброс ветки master  
    
63. Перебазирование  
    
64. Слияние style в master  
    
65. Изменяем оригинальный репозиторий  
    
66. Проверяем README  
    рис. 93
67. Сливаем извлеченные изменения  
    
68. Добавляем локальную ветку, отслеживающую оригинальную  
    
69. Отправляем изменения  
     
70. Извлекаем общие изменения  
    

# Выводы

В ходе данной работы были получены навыки использования системы контроля версий git. А также работы с сервисом GitHub.