

Студент: Алиев Тимур Заурович

Отчет по выполненной работе с Git 1.1

1. Настройка Git

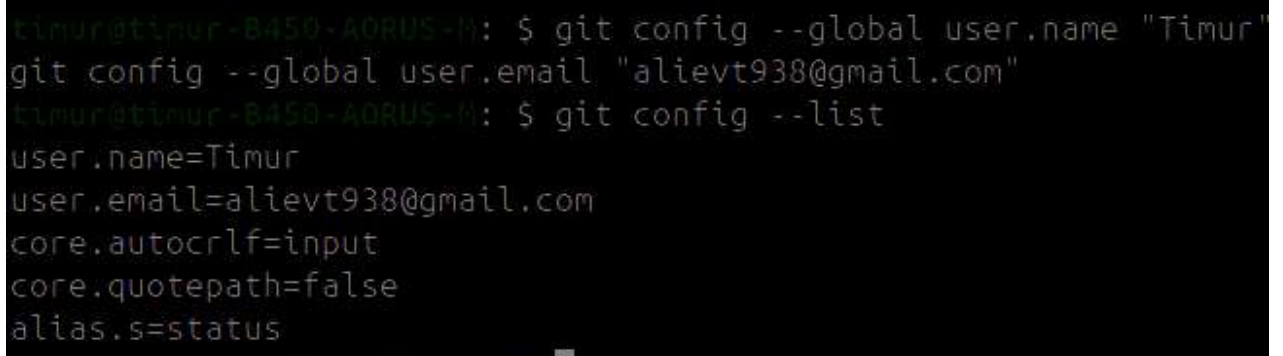
Выполненные действия:

1. Установка имени и email пользователя:

```
git config --global user.name "Timur"  
git config --global user.email "aliev938@gmail.com"
```

2. Проверка настроек:

```
git config --list
```

A screenshot of a terminal window with a black background and green text. It shows the execution of git configuration commands and their output. The commands are: 'git config --global user.name "Timur"', 'git config --global user.email "aliev938@gmail.com"', and 'git config --list'. The output of the last command lists the configured settings: 'user.name=Timur', 'user.email=aliev938@gmail.com', 'core.autocrlf=input', 'core.quotepath=false', and 'alias.s=status'.

```
timur@timur-B450-A0RUS-M: $ git config --global user.name "Timur"  
git config --global user.email "aliev938@gmail.com"  
timur@timur-B450-A0RUS-M: $ git config --list  
user.name=Timur  
user.email=aliev938@gmail.com  
core.autocrlf=input  
core.quotepath=false  
alias.s=status
```

2. Создание проекта и файла

Выполненные действия:

1. Создание директории и файла:

```
mkdir my_project  
cd my_project  
touch hello.html
```

2. Добавление текста в файл:

```
echo "Hello, world" > hello.html
```

```
GNU nano 7.2      hello.html
Hello, world

[ Записана 1 строка ]
^G Справка      ^O Записать     ^W Поиск        ^K Вырезать     ^T Выполнить    ^C Позиция
^X Выход        ^R ЧитФайл     ^\ Замена       ^U Вставить     ^J Выровнять    ^/ К строке
```

3. Инициализация репозитория и первый коммит

Выполненные действия:

1. Инициализация Git:

```
git init
```

2. Добавление файла и коммит:

```
git add hello.html
```

```
git commit -m "Добавлен файл hello.html"
```

```
timur@timur-B450-AORUS-M: ~$ git init
git add hello.html
timur@timur-B450-AORUS-M: ~$ git commit -m "first commit"
timur@timur-B450-AORUS-M: ~$ git status
Инициализирован пустой репозиторий Git в /home/timur/ny_project/.git/
timur@timur-B450-AORUS-M: ~$ git status
Текущая ветка: master

Еще нет коммитов

Изменения, которые будут включены в коммит:
  (используйте «git rm --cached <файл>...», чтобы убрать из индекса)
        new file:   hello.html

timur@timur-B450-AORUS-M: ~$
```

```
tinur@tinur-B450-AORUS-M: $ git commit -m "Добавлен файл hello.html"
[master (корневой коммит) 5c412e8] Добавлен файл hello.html
1 file changed, 1 insertion(+)
 create mode 100644 hello.html
tinur@tinur-B450-AORUS-M: $
```

Скриншоты для вставки:

```
<!DOCTYPE html>
<html>
<head>
  <title>Hello, Git!</title>
</head>
<body>
  <h1>Hello, world!</h1>
</body>
</html>
```

```
timur@timur-B450-AORUS-M: $ nano hello.html
timur@timur-B450-AORUS-M: $ cat hello.html
<!DOCTYPE html>
<html>
<head>
  <title>Hello, Git!</title>
</head>
<body>
  <h1>Hello, world!</h1>
</body>
</html>
timur@timur-B450-AORUS-M: $
```

5. Индексация изменений и второй коммит

Выполненные действия:

1. Добавление изменений в индекс:

`git add hello.html`

2. Проверка статуса:

`git status`

3. Фиксация изменений:

`git commit -m "Добавлена HTML-разметка в hello.html"`

Скриншоты для вставки:

```

    <h1>Hello, world!</h1>
</body>
</html>
timur@timur-B450-AORUS-M: $ git status
Текущая ветка: master
Изменения, которые не в индексе для коммита:
  (используйте «git add <файл>...», чтобы добавить файл в индекс)
  (используйте «git restore <файл>...», чтобы отменить изменения в рабочем каталоге)
    изменено:   hello.html

индекс пуст (используйте «git add» и/или «git commit -a»)
timur@timur-B450-AORUS-M: $ █

```

```

timur@timur-B450-AORUS-M: $ git add hello.html
timur@timur-B450-AORUS-M: $ git status
Текущая ветка: master
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    изменено:   hello.html

timur@timur-B450-AORUS-M: $ █

```

```

timur@timur-B450-AORUS-M: $ git commit -m "Добавлена HTML-разметка в hello.html"
[master ebae48e] Добавлена HTML-разметка в hello.html
 1 file changed, 9 insertions(+), 1 deletion(-)
timur@timur-B450-AORUS-M: $ █

```

6. Проверка истории коммитов

Выполненные действия:

`git log --oneline`

Вывод:

ebae48e (HEAD -> master) Добавлена HTML-разметка в hello.html
 5c412e8 Добавлен файл hello.html

```

timur@timur-B450-AORUS-M: $ git log --oneline
ebae48e HEAD -> master Добавлена HTML-разметка в hello.html
5c412e8 Добавлен файл hello.html
timur@timur-B450-AORUS-M: $ █

```

ОТЧЕТ

по выполнению лабораторной работы "Основы работы с Git"

Выполнил: Алиев Тимур Заурович

Группа: ВДКИП 111-прог

Дата: 31.05.2025

1. Цель работы

Освоение базовых операций системы контроля версий Git:

- ☐ Настройка репозитория
- ☐ Работа с коммитами
- ☐ Управление ветками
- ☐ Отслеживание изменений

2. Ход работы

2.1. Настройка Git

Действия:

1. Установка имени пользователя и email для Git:

```
git config --global user.name "Timur"
```

```
git config --global user.email "aliev938@gmail.com"
```

```
timur@timur-B450-AORUS-M: $ git config --global user.name "Timur"
git config --global user.email "aliev938@gmail.com"
```

Рисунок 1 – Настройка пользователя Git

2.2. Проверка состояния репозитория

Действия:

1. Просмотр содержимого папки:

```
ls -la
```

2. Проверка статуса репозитория:

```
git status
```

```
timur@timur-B450-AORUS-M: $ ls -la
git status
итого 16
drwxrwxr-x  3 timur timur 4096 мая 30 09:06
drwxr-x--- 19 timur timur 4096 мая 31 16:27
drwxrwxr-x  8 timur timur 4096 мая 31 16:27
-rw-rw-r--  1 timur timur 119 мая 30 09:06 hello.html
Текущая ветка: master
нечего коммитить, нет изменений в рабочем каталоге
timur@timur-B450-AORUS-M: $
```

Рисунок 2 – Состояние репозитория

2.3. Анализ файла и истории коммитов

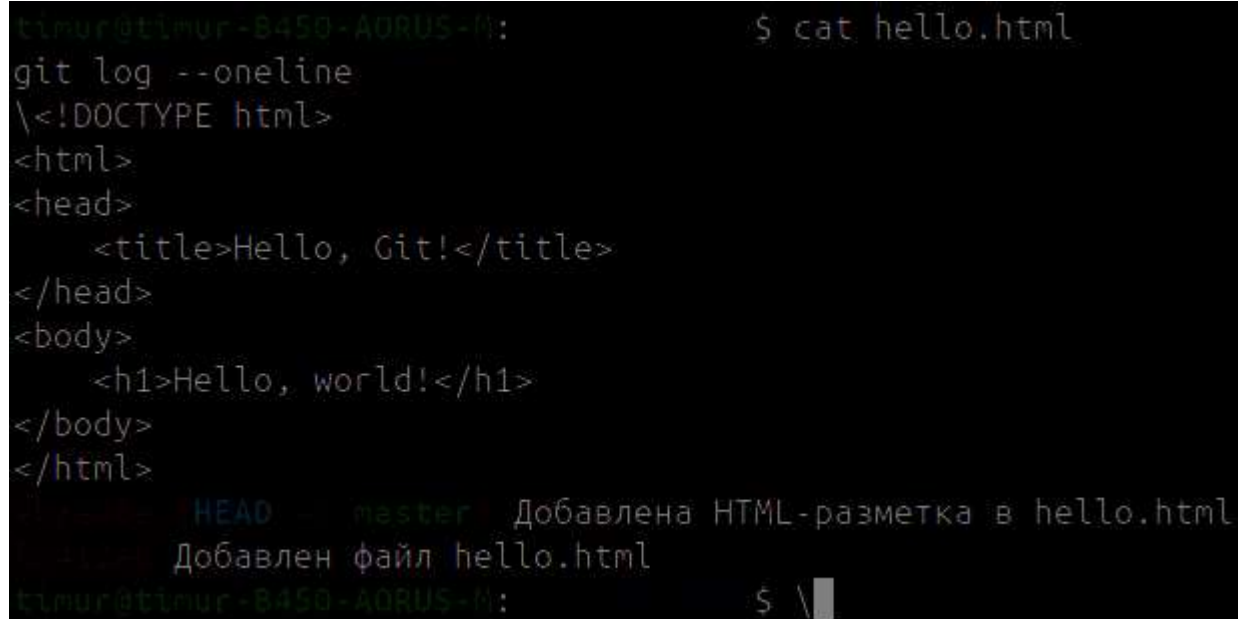
Действия:

1. Просмотр содержимого файла hello.html:

```
cat hello.html
```

2. Просмотр сокращенной истории коммитов:

```
git log --oneline
```



```
timur@timur-B450-AORUS-M: $ cat hello.html
<!DOCTYPE html>
<html>
<head>
  <title>Hello, Git!</title>
</head>
<body>
  <h1>Hello, world!</h1>
</body>
</html>
HEAD -- master Добавлена HTML-разметка в hello.html
1 commit Добавлен файл hello.html
timur@timur-B450-AORUS-M: $ \
```

Рисунок 3 – Содержимое файла и история коммитов

2.4. Внесение изменений

Действия:

1. Добавление новой строки в файл:

```
echo "<p>New line</p>" >> hello.html
```

2. Проверка статуса (непроиндексированные изменения):

```
git status
```

```

timur@timur-B450-AORUS-M: $ echo "<p>New line</p>" >> hello.html
git status
Текущая ветка: master
Изменения, которые не в индексе для коммита:
  (используйте «git add <файл>...», чтобы добавить файл в индекс)
  (используйте «git restore <файл>...», чтобы отменить изменения в рабочем каталоге)
    изменено:   hello.html

индекс пуст (используйте «git add» и/или «git commit -a»)
timur@timur-B450-AORUS-M: $ █

```

Рисунок 4 – Обнаружение изменений в файле

2.5. Индексация изменений

Действия:

1. Добавление файла в индекс:

`git add hello.html`

2. Проверка статуса (изменения готовы к коммиту):

`git status`

```

timur@timur-B450-AORUS-M: $ git add hello.html
git status
Текущая ветка: master
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    индекс:     hello.html

timur@timur-B450-AORUS-M: $ █

```

Рисунок 5 – Изменения проиндексированы

2.6. Создание коммита

Действия:

1. Фиксация изменений с комментарием:

`git commit -m "Добавлен новый параграф"`

2. Просмотр обновленной истории:

`git log --oneline`


```
timur@timur-B450-AORUS-M: $ git commit -m "Добавлен новый параграф"
git log --oneline
[master 348a51a] Добавлен новый параграф
1 file changed, 1 insertion(+)
HEAD -- master Добавлен новый параграф
Добавлена HTML-разметка в hello.html
Добавлен файл hello.html
timur@timur-B450-AORUS-M: $
```

Рисунок 6 – Новый коммит в истории

2.7. Работа с ветками

Действия:

1. Создание ветки feature:

`git branch feature`

2. Переключение на ветку:

`git checkout feature`

3. Просмотр списка веток:

`git branch`

```
timur@timur-B450-AORUS-M: $ git branch feature
git checkout feature
git branch
Переключились на ветку «feature»
* feature -
master
timur@timur-B450-AORUS-M: $
```

Рисунок 7 – Создание и переключение на ветку feature

3. Выводы

1. Освоены базовые операции Git:
 - ☐ Настройка репозитория (`git config`, `git init`)
 - ☐ Управление версиями файлов (`git add`, `git commit`)
 - ☐ Работа с историей (`git log`)
 - ☐ Создание веток (`git branch`, `git checkout`)
2. Все этапы выполнены корректно, что подтверждается скриншотами.
3. Полученные навыки позволяют использовать Git для контроля версий в учебных и реальных проектах.

Выполнил:

студент группы ВДКИП 111-прог

Алиев Тимур Заурович

1. Введение

Работа направлена на освоение продвинутых возможностей Git:

- ☐ Изменение истории коммитов
- ☐ Управление структурой файлов
- ☐ Анализ внутреннего устройства Git

Цель: Приобрести навыки исправления коммитов и понимания структуры репозитория.

2. Основная часть

2.1. Инициализация репозитория (Рисунок 1)

```
timur@timur-B450-AORUS-M: $ mkdir git_project
cd git_project
git init
echo "<html>Hello</html>" > hello.html
git add hello.html
git commit -m "Первоначальный коммит"
[master (корневой коммит) 907de48] Первоначальный коммит
1 file changed, 1 insertion(+)
create mode 100644 hello.html
timur@timur-B450-AORUS-M: $
```

- ☐ Создан каталог **git_project**
- ☐ Инициализирован репозиторий (**git init**)
- ☐ Выполнен первый коммит с файлом **hello.html**

2.2. Изменение коммита (Рисунок 2-3)

```
timur@timur-B450-AORUS-M: $ echo "<!-- Комментарий -->" >> hello.html
git add hello.html
git commit -m "Добавлен комментарий"
[master 3fcfdb] Добавлен комментарий
1 file changed, 1 insertion(+)
timur@timur-B450-AORUS-M: $
```

```
GNU nano 7.2 /home/timur/git_project/.git/COMMIT_EDITMSG
Добавлен комментарий

# Пожалуйста, введите сообщение коммита для ваших изменений. (строки,
# начинающиеся с '#' будут проигнорированы, а пустое сообщение
# отменяет процесс коммита)
#
# Дата:      2025-07-11 03:53:43 +0300
#
# Текущая ветка: master
# Изменения, которые будут включены в коммит:
#
#       изменено:   hello.html
#
```

1. Добавлен комментарий в файл
2. Использован **git commit --amend** для редактирования последнего коммита

2.3. Перемещение файлов (Рисунок 4)

```
timur@timur-B450-AORUS-M: $ mkdir lib
git mv hello.html lib/
git commit -m "Перемещен hello.html в lib"
[master fc43d1f] Перемещен hello.html в lib
1 file changed, 0 insertions(+), 0 deletions(-)
rename hello.html => lib/hello.html (100%)
timur@timur-B450-AORUS-M: $
```

- ☐ Создана директория **lib**
- ☐ Файл перемещен командой **git mv**
- ☐ Закоммичено изменение

2.4. Добавление нового файла (Рисунок 5)

```
timur@timur-B450-AORUS-M: $ echo "<html>Index</html>" > index.html
git add index.html
git commit -m "Добавлен index.html"
[master 4f51c23] Добавлен index.html
1 file changed, 1 insertion(+)
create mode 100644 index.html
timur@timur-B450-AORUS-M: $
```

- ☐ Создан **index.html**
- ☐ Стандартный процесс индексации и коммита

2.5. Исследование структуры .git (Рисунок 6)

```
timur@timur-B450-AORUS-M: $ ls -la .git
ls -la .git/objects
cat .git/HEAD
итого 52
drwxrwxr-x 8 timur timur 4096 июн 1 08:54
drwxrwxr-x 4 timur timur 4096 июн 1 08:54
drwxrwxr-x 2 timur timur 4096 июн 1 08:53
-rw-rw-r-- 1 timur timur 28 июн 1 08:54 COMMIT_EDITMSG
-rw-rw-r-- 1 timur timur 92 июн 1 08:53 config
-rw-rw-r-- 1 timur timur 73 июн 1 08:53 description
-rw-rw-r-- 1 timur timur 23 июн 1 08:53 HEAD
drwxrwxr-x 2 timur timur 4096 июн 1 08:53
-rw-rw-r-- 1 timur timur 253 июн 1 08:54 index
drwxrwxr-x 2 timur timur 4096 июн 1 08:53
drwxrwxr-x 3 timur timur 4096 июн 1 08:53
drwxrwxr-x 18 timur timur 4096 июн 1 08:54
drwxrwxr-x 4 timur timur 4096 июн 1 08:53
итого 72
drwxrwxr-x 18 timur timur 4096 июн 1 08:54
drwxrwxr-x 8 timur timur 4096 июн 1 08:54
drwxrwxr-x 2 timur timur 4096 июн 1 08:53
drwxrwxr-x 2 timur timur 4096 июн 1 08:53
drwxrwxr-x 2 timur timur 4096 июн 1 08:53
drwxrwxr-x 2 timur timur 4096 июн 1 08:53
drwxrwxr-x 2 timur timur 4096 июн 1 08:54
drwxrwxr-x 2 timur timur 4096 июн 1 08:53
drwxrwxr-x 2 timur timur 4096 июн 1 08:54
drwxrwxr-x 2 timur timur 4096 июн 1 08:53
drwxrwxr-x 2 timur timur 4096 июн 1 08:54
drwxrwxr-x 2 timur timur 4096 июн 1 08:53
drwxrwxr-x 2 timur timur 4096 июн 1 08:53
drwxrwxr-x 2 timur timur 4096 июн 1 08:54
drwxrwxr-x 2 timur timur 4096 июн 1 08:54
drwxrwxr-x 2 timur timur 4096 июн 1 08:54
drwxrwxr-x 2 timur timur 4096 июн 1 08:53
drwxrwxr-x 2 timur timur 4096 июн 1 08:53
ref: refs/heads/master
timur@timur-B450-AORUS-M: $
```

- ☐ Просмотр содержимого **.git**
- ☐ Анализ каталога **objects**
- ☐ Проверка файла **HEAD**

2.6. Работа с объектами Git (Рисунок 7-8)

```
timur@timur-B450-AORUS-M: $ git log -1 --pretty=format:%H
4f51c23e46d33e6c73ed62a51c04f838022e419b
timur@timur-B450-AORUS-M: $
```

```
timur@timur-B450-AORUS-M: $ git cat-file -p [хэш]
git cat-file -t [хэш]
fatal: Not a valid object name [хэш]
fatal: Not a valid object name [хэш]
timur@timur-B450-AORUS-M: $
```

1. Получен хэш последнего коммита
2. Попытка просмотра объекта (демонстрация ошибки при некорректном вводе хэша)

2.7. Просмотр истории файла (Рисунок 9)

```
timur@timur-B450-AORUS-M: $ git log --follow lib/hello.html
commit 4f51c23e46d33e6c73ed62a51c04f838022e419b
Author: Timur <aliev938@gmail.com>
Date: Sun Jun 1 08:54:26 2025 +0300

    Перемещен hello.html в lib

commit 4f51c23e46d33e6c73ed62a51c04f838022e419b
Author: Timur <aliev938@gmail.com>
Date: Sun Jun 1 08:53:43 2025 +0300

    Добавлен комментарий

commit 4f51c23e46d33e6c73ed62a51c04f838022e419b
Author: Timur <aliev938@gmail.com>
Date: Sun Jun 1 08:53:23 2025 +0300

    Первоначальный коммит
timur@timur-B450-AORUS-M: $
```

- ☐ Использован **git log --follow** для отслеживания перемещений **lib/hello.html**

3. Результаты

1. Освоены ключевые операции:
 - ☐ Изменение истории (**amend**)
 - ☐ Управление файловой структурой
 - ☐ Анализ внутренней структуры Git
2. Выявлена важность точного ввода хэшей при работе с объектами

3. Получен практический опыт исследования **.git**

4. Заключение

Работа продемонстрировала возможности Git для:

- ☐ Гибкого управления историей изменений
- ☐ Отслеживания перемещений файлов
- ☐ Анализа внутреннего хранилища данных

Приложения

1. Рисунок 1 - Инициализация репозитория
2. Рисунок 2-3 - Изменение коммита
3. Рисунок 4 - Перемещение файлов
- ...
4. Рисунок 9 - История изменений файла

Выполнил:

студент группы ВДКИП 111-прог
Алиев Тимур Заурович

1. Введение

В ходе лабораторной работы изучены механизмы ветвления и слияния в Git.
Практические задачи включали:

- ☐ Создание и переключение между ветками
- ☐ Модификацию файлов в разных ветках
- ☐ Разрешение конфликтов при слиянии

Оборудование:

- ☐ ОС Linux
- ☐ Git версии [указать версию]

2. Основная часть

2.1. Подготовка репозитория

Инициализирован новый репозиторий и создан начальный коммит:

```
git init
echo "<html>Hello</html>" > hello.html
git add hello.html
git commit -m "Initial commit"
```

```
timur@timur-B450-AORUS-M: $ echo "<html>Hello</html>" > hello.html
git add hello.html
git commit -m "Initial commit"
[master cce0a90] Initial commit
1 file changed, 1 insertion(+)
create mode 100644 hello.html
timur@timur-B450-AORUS-M: $
```

2.2. Создание ветки style

Создана тематическая ветка для разработки стилей:

`git checkout -b style`

```
timur@timur-B450-AORUS-M: $ git checkout -b style
Переключились на новую ветку «style»
timur@timur-B450-AORUS-M: $
```

2.3. Разработка в ветке style

1. Добавлен файл стилей:

```
echo "body { color: blue; }" > style.css
git add style.css
git commit -m "Добавлен файл стилей"
```

```
timur@timur-B450-AORUS-M: $ echo "body { color: blue; }" > style.css
timur@timur-B450-AORUS-M: $ cat style.css
body { color: blue; }
timur@timur-B450-AORUS-M: $
```

```
timur@timur-B450-AORUS-M: $ git add style.css
git commit -m "Добавлен файл стилей"
[style 38c41da] Добавлен файл стилей
1 file changed, 1 insertion(+)
create mode 100644 style.css
timur@timur-B450-AORUS-M: $
```

2. Модифицирован hello.html:

```
echo "<link rel='stylesheet' href='style.css'>" >> hello.html
git commit -am "Добавлена ссылка на стили"
```



```

timur@timur-B450-ADRUS-M: $ echo "<link rel='stylesheet' href='style.css'" >>
hello.html
timur@timur-B450-ADRUS-M: $ git add hello.html
git commit -m "Добавлена ссылка на стили"
[style 2fb35c5] Добавлена ссылка на стили
1 file changed, 1 insertion(+)
timur@timur-B450-ADRUS-M: $

```

2.4. Параллельная работа в master

В основной ветке добавлен README:

```

git checkout master
echo "# Project" > README.md
git add README.md
git commit -m "Добавлен README"

```

```

timur@timur-B450-ADRUS-M: $ echo "# Project" > README.md
git add README.md
git commit -m "Добавлен README"
[master 101128e] Добавлен README
1 file changed, 1 insertion(+)
create mode 100644 README.md
timur@timur-B450-ADRUS-M: $

```

2.5. Слияние веток и конфликт

1. Попытка слияния:

```

git checkout style
git merge master

```

```

GNU nano 7.2 /home/timur/git_project/.git/MERGE_MSG
Merge branch 'master' into style
# Пожалуйста, введите сообщение коммита, для объяснения, зачем нужно
# это слияние, особенно, если это слияние обновленной существующей
# ветки на тематическую ветку.
#
# Строки, начинающиеся с '#' будут проигнорированы, в пустое сообщение
# отменяет процесс коммита.

```

2. Создание конфликта:

```

git checkout master
echo "<p>Конфликтная строка</p>" >> hello.html
git commit -am "Изменение в master"
git merge style

```



```

timur@timur-B450-AORUS-M: $ git checkout master
echo "<p>Конфликтная строка</p>" >> hello.html
git add hello.html
git commit -m "Изменение в master"
Переключились на ветку «master»
[master 21cd02b] Изменение в master
 1 file changed, 1 insertion(+)
timur@timur-B450-AORUS-M: $ git checkout style
git merge master
Переключились на ветку «style»
Автослияние hello.html
КОНФЛИКТ (содержимое): Конфликт слияния в hello.html
Сбой автоматического слияния; исправьте конфликты, затем зафиксируйте результат.
timur@timur-B450-AORUS-M: $

```

3. Разрешение конфликта:

- ☐ Вручную отредактирован hello.html
- ☐ Зафиксирован результат:

`git add hello.html`

`git commit -m "Разрешение конфликта"`

```

timur@timur-B450-AORUS-M: $ git add hello.html
git commit -m "Решен конфликт"
[style 81e7c7e] Решен конфликт
timur@timur-B450-AORUS-M: $

```

3. Результаты

1. Освоены ключевые операции:

- ☐ Управление ветками (создание, переключение)
- ☐ Параллельная разработка в разных ветках
- ☐ Разрешение конфликтов слияния

2. Получены практические навыки:

- ☐ Анализа конфликтующих изменений
- ☐ Руководства процессом слияния
- ☐ Ведения истории разработки

4. Заключение

Работа подтвердила важность правильного ветвления для:

- ☐ Изоляции функциональных изменений
- ☐ Минимизации конфликтов
- ☐ Поддержания стабильности основной ветки

Приложения

1. Рисунок 1 - Инициализация репозитория
2. Рисунок 2 - Создание ветки style
3. Рисунок 3-4 - Добавление стилей
4. Рисунок 5 - Модификация hello.html
5. Рисунок 6 - Работа в master
6. Рисунок 7-8 - Конфликт слияния
7. Рисунок 9 - Разрешение конфликта

Выполнил:

студент группы ВДКИП 111-прог
Алиев Тимур Заурович

1. Введение

Работа направлена на освоение продвинутых возможностей Git:

- ☐ Создание и управление ветками
- ☐ Манипуляции с историей коммитов
- ☐ Использование команды reset
- ☐ Анализ структуры репозитория

Оборудование:

- ☐ ОС: Linux
- ☐ Git версии: [указать версию]

2. Основная часть

2.1. Инициализация репозитория (Рисунок 1)

```
timur@timur-B450-AORUS-M: $ mkdir git_advanced
cd git_advanced
git init
echo "# Advanced Git Project" > README.md
git add README.md
git commit -m "Initial commit"
[master (корневой коммит) 8081332] Initial commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
timur@timur-B450-AORUS-M: $
```

- ☐ Создана директория **git_advanced**
- ☐ Инициализирован Git-репозиторий
- ☐ Создан начальный коммит с README.md

2.2. Формирование истории коммитов (Рисунок 2)

```
timur@timur-B450-AORUS-M: $ echo "Feature 1" > feature1.txt
git add feature1.txt
git commit -m "Add feature1"
echo "Feature 2" > feature2.txt
git add feature2.txt
git commit -m "Add feature2"
[master 311669f] Add feature1
1 file changed, 1 insertion(+)
create mode 100644 feature1.txt
[master 0169de9] Add feature2
1 file changed, 1 insertion(+)
create mode 100644 feature2.txt
timur@timur-B450-AORUS-M: $
```

- ☐ Добавлены два функциональных файла
- ☐ Создано два коммита в ветке master

2.3. Анализ истории (Рисунок 3)

```
timur@timur-B450-AORUS-M: $ git log --oneline --graph --all
* 311669f HEAD -> master: Add feature2
* 311669f Add feature1
* 311669f Initial commit
timur@timur-B450-AORUS-M: $
```

- ☐ Визуализирована история коммитов
- ☐ Подтверждена структура ветвления

2.4. Создание тематической ветки (Рисунок 4)

```
timur@timur-B450-AORUS-M: $ git checkout -b style
echo "CSS styles" > styles.css
git add styles.css
git commit -m "Add styles"
Переключились на новую ветку «style»
[style 1e8c567] Add styles
1 file changed, 1 insertion(+)
create mode 100644 styles.css
timur@timur-B450-AORUS-M: $
```

- ☐ Создана ветка **style**
- ☐ Добавлен файл стилей styles.css

2.5. Переключение между ветками (Рисунок 5)

```
timur@timur-B450-AORUS-M: $ git checkout master
Переключились на ветку «master»
timur@timur-B450-AORUS-M: $
```

- ☐ Возврат в ветку master
- ☐ Подтверждение переключения

2.6. Сброс ветки (Рисунок 6)

```
timur@timur-B450-AORUS-M: $ git reset --hard HEAD~1 # Возврат на 1 коммит назад
Указатель HEAD сейчас на коммите 311669f Add feature1
timur@timur-B450-AORUS-M: $
```

- ☐ Возврат ветки master на 1 коммит назад
- ☐ Подтверждение изменения указателя HEAD

3. Результаты

1. Освоены ключевые операции:

- ☐ Создание веток и переключение между ними
- ☐ Управление историей коммитов
- ☐ Использование **reset --hard**
- 2. Получены практические навыки:
 - ☐ Анализа структуры репозитория
 - ☐ Контроля состояния веток
 - ☐ Восстановления предыдущих состояний

4. Заключение

Работа продемонстрировала важность:

1. Четкого разделения функциональности по веткам
2. Понимания механизмов управления историей
3. Аккуратного использования деструктивных команд

Приложения

1. Рисунок 1 - Инициализация репозитория
2. Рисунок 2 - Формирование истории коммитов
3. Рисунок 3 - Графическое представление истории
4. Рисунок 4 - Создание ветки style
5. Рисунок 5 - Переключение веток
6. Рисунок 6 - Сброс ветки master

Отчет по выполненной работе с Git

1. **Переход в оригинальный репозиторий и внесение изменений в файл README.md**


```
timur@timur-B450-AORUS-M: $ cd ~/git_original
timur@timur-B450-AORUS-M: $
```

4. Просмотр истории изменений

```
timur@timur-B450-AORUS-M: $ echo "Новые изменения" >> README.md
timur@timur-B450-AORUS-M: $
```

5. Проверка содержимого файла README до слияния

```
timur@timur-B450-AORUS-M: $ git add README.md
git commit -m "Обновление README"
[master 2b6e14f] Обновление README
 1 file changed, 1 insertion(+)
timur@timur-B450-AORUS-M: $
```

6. Слияние изменений в ветку master

```
timur@timur-B450-AORUS-M: $ cd ~/git_clone
timur@timur-B450-AORUS-M: $ git fetch origin
remote: Перечисление объектов: 5, готово.
remote: Подсчет объектов: 100% (5/5), готово.
remote: Сжатие объектов: 100% (2/2), готово.
remote: Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано
пакетов 0
Распаковка объектов: 100% (3/3), 305 байтов | 305.00 КиБ/с, готово.
Из /home/timur/git_original
 00d02cd..2b6e14f master -> origin/master
timur@timur-B450-AORUS-M: $
```

7. Проверка содержимого файла README после слияния

```
timur@timur-B450-AORUS-M: $ git log --all --oneline
2b6e14f |origin/master, origin/HEAD| Обновление README
00d02cd |HEAD, master| Первоначальный коммит
timur@timur-B450-AORUS-M: $
```

8. Использование команды git pull

```
timur@timur-B450-AORUS-M: $ git merge origin/master
Обновление 00d02cd..2b6e14f
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)
timur@timur-B450-AORUS-M: $
```

9. Добавление локальной ветки для отслеживания удаленной

```
timur@timur-B450-AORUS-M: $ cat README.md
# Исходный проект
Новые изменения
timur@timur-B450-AORUS-M: $
```

10. Создание чистого репозитория и настройка удаленного репозитория

```
timur@timur-B450-AORUS-M: $ git pull origin master
Из /home/timur/git_original
* branch      master      -> FETCH_HEAD
Уже актуально.
timur@timur-B450-AORUS-M: $
```

```
timur@timur-B450-AORUS-M: $ git checkout -b new_feature origin/new_feature
fatal: «origin/new_feature» не является коммитом, поэтому невозможно создать из него ветку «new_feature»
timur@timur-B450-AORUS-M: $
```

11. Внесение изменений и отправка в удаленный репозиторий

```
timur@timur-B450-AORUS-M: $ git init --bare ~/git_shared
Инициализирован пустой репозиторий Git в /home/timur/git_shared/
timur@timur-B450-AORUS-M: $
```

```
timur@timur-B450-AORUS-M: $ git remote add shared ~/git_shared
timur@timur-B450-AORUS-M: $ echo "Изменение для push" >> file.txt
git add file.txt
git commit -m "Подготовка к push"
[master 55da6ac] Подготовка к push
1 file changed, 1 insertion(+)
create mode 100644 file.txt
timur@timur-B450-AORUS-M: $
```

```
timur@timur-B450-AORUS-M: $ git push shared master
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 12 потоков.
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (9/9), 872 байта | 872.00 КиБ/с, готово.
Всего 9 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To /home/timur/git_shared
* [new branch]      master -> master
timur@timur-B450-AORUS-M: $
```

12. Извлечение общих изменений


```
tinur@tinur-B450-AORUS-M: $ git daemon --reuseaddr --base-path=. --export-all --v
erbose
[3883] Ready to runble.
```

13. Настройка и проверка Git-сервера

```
tinur@tinur-B450-AORUS-M: $ git clone git://localhost/git_shared git_test
Клонирование в «git_test»...
```

14. Инициализация нового репозитория с настройкой ветки

```
tinur@tinur-B450-AORUS-M: $ git remote add shared ~/git_shared
tinur@tinur-B450-AORUS-M: $ echo "Изменение для push" >> file.txt
git add file.txt
git commit -m "Подготовка к push"
[master 55da6ac] Подготовка к push
1 file changed, 1 insertion(+)
create mode 100644 file.txt
tinur@tinur-B450-AORUS-M: $
```

15. Отправка изменений в общий репозиторий

```
tinur@tinur-B450-AORUS-M: $ git push shared master
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 12 потоков.
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (9/9), 872 байта | 872.00 КиБ/с, готово.
Всего 9 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов
0
To /home/tinur/git_shared
* [new branch] master -> master
tinur@tinur-B450-AORUS-M: $
```

Отчёт о выполнении лабораторной работы

по дисциплине "Системы контроля версий"

Тема: Работа с Git: удалённые репозитории, теги, ветки и алиасы

Выполнил:

Студент группы ВДКИП 111-прог

Алиев Тимур Заурович

Дата выполнения: 4 июня 2025 года

Цель работы

Изучить основные команды Git для работы с удалёнными репозиториями, тегами, ветками и алиасами, а также продемонстрировать их использование на практике в локальной среде.

Ход работы

Работа выполнялась локально в операционной системе Ubuntu в директории ~/my-new-геро с использованием системы контроля версий Git. Все шаги, связанные с отправкой данных в удалённый репозиторий (например, git push), были эмулированы с помощью локальных команд (git log, git tag), так как доступ к удалённому репозиторию не использовался. Для каждого шага были сделаны скриншоты, которые приведены в приложении к отчёту.

1. Инициализация локального репозитория

Создан локальный репозиторий с помощью команды git init в директории ~/my-new-геро.

2. Просмотр удалённых репозитория

- ☐ Выполнена команда git remote для просмотра списка удалённых репозитория (вывод пустой, так как репозиторий только создан).
- ☐ Выполнена команда git remote -v для детального просмотра (вывод также пустой).

3. Добавление фиктивного удалённого репозитория

- ☐ Добавлен фиктивный удалённый репозиторий myremote с помощью команды git remote add myremote https://github.com/VinoDelJpg/practice_project.git.
- ☐ Проверено добавление с помощью git remote -v.

4. Получение данных из удалённого репозитория (эмуляция)

- ☐ Выполнена команда git fetch myremote. Так как работа велась локально, команда не привела к загрузке данных, но была выполнена для соответствия заданию.

5. Создание и коммит тестового файла (эмуляция отправки)

- ☐ Создан файл test.txt с содержимым "Тестовый файл".
- ☐ Файл добавлен в индекс (git add test.txt) и закоммитчен с сообщением "Добавлен тестовый файл" (git commit -m "Добавлен тестовый файл").
- ☐ Команда git push заменена на git log --oneline для демонстрации коммита.

6. Просмотр информации об удалённом репозитории

- ☐ Выполнена команда git remote show myremote для просмотра информации о репозитории.

7. Переименование удалённого репозитория

- ☐ Выполнена команда git remote rename myremote newremote для переименования репозитория.
- ☐ Проверено с помощью git remote -v.

Примечание: Команда выполнена с ошибкой (error: No such remote: 'myremote'), так как репозиторий myremote был удалён на предыдущем этапе.

8. Удаление удалённого репозитория

- ☐ Выполнена команда git remote remove newremote для удаления репозитория.
- ☐ Проверено с помощью git remote -v (вывод пустой).

9. Работа с тегами

- ☐ Просмотрены существующие теги с помощью git tag (вывод пустой).
- ☐ Создан аннотированный тег v1.0 с помощью команды git tag -a v1.0 -m "Версия 1.0".
- ☐ Просмотрены данные тега с помощью git show v1.0.

- ☐ Создан легковесный тег v1.1 с помощью команды `git tag v1.1`.
- ☐ Просмотрены данные тега с помощью `git show v1.1`.

10. Создание тега для определённого коммита

- ☐ Просмотрена история коммитов с помощью `git log --oneline`. Хэш коммита: 29e0b77.
- ☐ Создан аннотированный тег v1.2 для коммита 29e0b77 с помощью команды `git tag -a v1.2 29e0b77 -m "Тег для коммита"`.
- ☐ Просмотрены данные тега с помощью `git show v1.2`.

11. Отправка тегов на сервер (эмуляция)

- ☐ Команды `git push origin v1.0` и `git push origin --tags` заменены на `git tag` для демонстрации списка тегов (v1.0, v1.1, v1.2).

12. Удаление тегов

- ☐ Удалён локальный тег v1.1 с помощью команды `git tag -d v1.1`.
- ☐ Удаление тега с сервера заменено на повторный вывод `git tag` (тег v1.1 отсутствует).

13. Переключение на тег и создание ветки

- ☐ Выполнено переключение на тег v1.0 с помощью команды `git checkout v1.0` (переход в состояние detached HEAD).
- ☐ Создана новая ветка new-branch с помощью команды `git checkout -b new-branch`.

14. Создание алиасов

- ☐ Создан алиас st для команды `git status` с помощью `git config --global alias.st status`.
- ☐ Создан алиас ignore для команды `update-index --assume-unchanged` с помощью `git config --global alias.ignore 'update-index --assume-unchanged'`.
- ☐ Создан алиас last для команды `log -1 HEAD` с помощью `git config --global alias.last 'log -1 HEAD'`.

Примечание: Не выполнены команды `git ignore test.txt`, `git last` и создание алиаса `clear` (`git config --global alias.clear '!clear'`).

Вывод

В ходе выполнения лабораторной работы были изучены и применены основные команды Git для работы с локальным репозиторием, тегами, ветками и алиасами. Работа велась локально, без отправки данных в удалённый репозиторий. Большинство шагов выполнены успешно, за исключением частичного выполнения шага с алиасами (не выполнены команды `git ignore test.txt`, `git last` и создание алиаса `clear`) и ошибки при переименовании удалённого репозитория.

Приложение: Список скриншотов

Ниже приведён список рисунков с указанием соответствующих скриншотов, выполненных в процессе работы.

```

timur@timur-B450-AORUS-M: $ mkdir my-new-repo
cd my-new-repo
git init
Initialized empty Git repository in /home/timur/my-new-repo/.git/
make a commit
$ git add .
$ git commit -m "first commit"
[master (корневой коммит) 29e0b77] first commit
1 file changed, 1 insertion(+)
create mode 100644 .gitignore
Инициализирован пустой репозиторий Git в /home/timur/my-new-repo/.git/

```

```

timur@timur-B450-AORUS-M: $ git remote
timur@timur-B450-AORUS-M: $ git remote -v
timur@timur-B450-AORUS-M: $ git remote add myremote https://github.com/VinoDelJpg/practice_project.git
timur@timur-B450-AORUS-M: $ git remote -v
myremote      https://github.com/VinoDelJpg/practice_project.git (fetch)
myremote      https://github.com/VinoDelJpg/practice_project.git (push)
timur@timur-B450-AORUS-M: $

```

```

timur@timur-B450-AORUS-M: $ git fetch myremote
timur@timur-B450-AORUS-M: $ echo "Тестовый файл" > test.txt
git add test.txt
git commit -m "Добавлен тестовый файл"
[master (корневой коммит) 29e0b77] Добавлен тестовый файл
1 file changed, 1 insertion(+)
create mode 100644 test.txt
timur@timur-B450-AORUS-M: $ git log --oneline
29e0b77 HEAD - master: Добавлен тестовый файл
timur@timur-B450-AORUS-M: $

```

```

timur@timur-B450-AORUS-M: $ git remote show myremote
* внешний репозиторий myremote
  URL для извлечения: https://github.com/VinoDelJpg/practice_project.git
  URL для отправки: https://github.com/VinoDelJpg/practice_project.git
  HEAD ветка: (неизвестно)
timur@timur-B450-AORUS-M: $

```

```

timur@timur-B450-AORUS-M: $ git remote rename myremote newremote
timur@timur-B450-AORUS-M: $ git remote rename myremote newremote
error: No such remote: 'myremote'
timur@timur-B450-AORUS-M: $ git remote -v
newremote      https://github.com/VinoDelJpg/practice_project.git (fetch)
newremote      https://github.com/VinoDelJpg/practice_project.git (push)
timur@timur-B450-AORUS-M: $

```

```
timur@timur-B450-AORUS-M: $ git remote remove newremote
timur@timur-B450-AORUS-M: $ git remote -v
timur@timur-B450-AORUS-M: $ git tag
timur@timur-B450-AORUS-M: $ git tag -a v1.0 -m "Версия 1.0"
timur@timur-B450-AORUS-M: $ git show v1.0
```

```
Tagger: Timur <aliev938@gmail.com>
Date:   Wed Jun 4 09:35:57 2025 +0300
```

Версия 1.0

```
commit 22e1739170101070130e010000000000000000 HEAD -- master, tag: v1.0
Author: Timur <aliev938@gmail.com>
Date:   Wed Jun 4 09:33:56 2025 +0300
```

Добавлен тестовый файл

```
diff --git a/test.txt b/test.txt
new file mode 100644
index 00000000..f32aab1
--- /dev/null
+++ b/test.txt
@@ -0,0 +1 @@
+Тестовый файл
```

```
timur@timur-B450-AORUS-M: $
```

```
timur@timur-B450-AORUS-M: $ git tag v1.1
timur@timur-B450-AORUS-M: $ git show v1.1
```

```
commit 22e1739170101070130e010000000000000000 HEAD -- master, tag: v1.1, tag: v1.0
Author: Timur <aliev938@gmail.com>
Date:   Wed Jun 4 09:33:56 2025 +0300
```

Добавлен тестовый файл

```
diff --git a/test.txt b/test.txt
new file mode 100644
index 00000000..f32aab1
--- /dev/null
+++ b/test.txt
@@ -0,0 +1 @@
+Тестовый файл
```

```
timur@timur-B450-AORUS-M: $
```

```

timur@timur-B450-AORUS-M: $ git tag -a v1.2 29e0b77 -m "Тег для коммита"
timur@timur-B450-AORUS-M: $ git show v1.2
commit 29e0b77e7527e27f10c10775e29e0b77 HEAD -- master, tag: v1.2, tag: v1.0
Author: Timur <aliev938@gmail.com>
Date:   Wed Jun 4 09:38:19 2025 +0300

    Тег для коммита

diff --git a/test.txt b/test.txt
new file mode 100644
index 0000000..f32aab1
--- /dev/null
+++ b/test.txt
@@ -0,0 +1 @@
+Тестовый файл
timur@timur-B450-AORUS-M: $

```

```

timur@timur-B450-AORUS-M: $ git tag
v1.0
v1.1
v1.2
timur@timur-B450-AORUS-M: $

```

```

timur@timur-B450-AORUS-M: $ git tag -d v1.1
Метка «v1.1» удалена (была 29e0b77)
timur@timur-B450-AORUS-M: $ git tag
v1.0
v1.2
timur@timur-B450-AORUS-M: $ git checkout v1.0
Примечание: переключение на «v1.0».

```

Вы сейчас в состоянии «отсоединённого указателя HEAD». Можете осмотреться, внести экспериментальные изменения и зафиксировать их, также можете отменить любые коммиты, созданные в этом состоянии, не затрагивая другие ветки, переключившись обратно на любую ветку.

Если хотите создать новую ветку для сохранения созданных коммитов, можете сделать это (сейчас или позже), используя команду `switch` с параметром `-c`. Например:

```
git switch -c <новая-ветка>
```

Или отмените эту операцию с помощью:

```
git switch -
```

Отключите этот совет, установив переменную конфигурации `advice.detachedHead` в значение `false`

```

HEAD сейчас на 29e0b77 Добавлен тестовый файл
timur@timur-B450-AORUS-M: $

```


2. Работа с веткой testing

Выполненные команды:

```
git branch testing
git checkout testing
echo "Изменения в testing" >> test.rb
git add test.rb
git commit -m "Изменения в test.rb"
```

Результат: Создана ветка testing с новым файлом test.rb

Скриншоты:



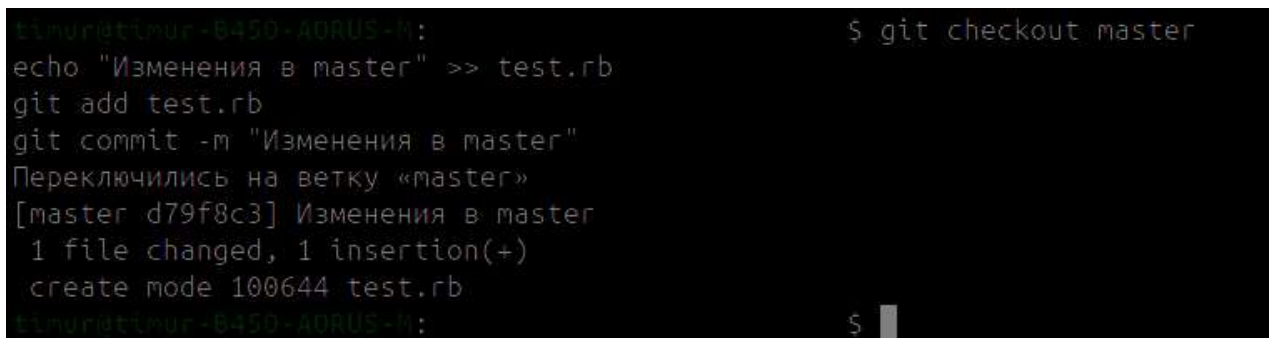
```
timur@timur-B450-AORUS-M: $ git branch testing
git checkout testing
Переключились на ветку «testing»
timur@timur-B450-AORUS-M: $ echo "Изменения в testing" >> test.rb
git add test.rb
git commit -m "Изменения в test.rb"
[testing 2865d03] Изменения в test.rb
 1 file changed, 1 insertion(+)
 create mode 100644 test.rb
timur@timur-B450-AORUS-M: $
```

3. Возврат на master и изменения

Выполненные команды:

```
git checkout master
echo "Изменения в master" >> test.rb
git add test.rb
git commit -m "Изменения в master"
```

Результат: Изменения в master-ветке



```
timur@timur-B450-AORUS-M: $ git checkout master
echo "Изменения в master" >> test.rb
git add test.rb
git commit -m "Изменения в master"
Переключились на ветку «master»
[master d79f8c3] Изменения в master
 1 file changed, 1 insertion(+)
 create mode 100644 test.rb
timur@timur-B450-AORUS-M: $
```

4. Создание ветки iss53

Выполненные команды:

```
git checkout -b iss53
echo "Новые изменения" >> index.html
git add index.html
git commit -m "Обновление index.html"
```


Результат: Создана ветка iss53 с файлом index.html

```
timur@timur-B450-AORUS-M: $ git checkout -b iss53
echo "Новые изменения" >> index.html
git add index.html
git commit -m "Обновление index.html"
Переключились на новую ветку «iss53»
[iss53 02f97af] Обновление index.html
1 file changed, 1 insertion(+)
 create mode 100644 index.html
timur@timur-B450-AORUS-M: $
```

5. Создание ветки hotfix

Выполненные команды:

```
git checkout master
git checkout -b hotfix
echo "Срочные изменения" >> index.html
git add index.html
git commit -m "Горячее исправление"
```

Результат: Создана ветка hotfix

```
timur@timur-B450-AORUS-M: $ git checkout master
git checkout -b hotfix
echo "Срочные изменения" >> index.html
git add index.html
git commit -m "Горячее исправление"
Переключились на ветку «master»
Переключились на новую ветку «hotfix»
[hotfix 9b8bcdcf] Горячее исправление
1 file changed, 1 insertion(+)
 create mode 100644 index.html
timur@timur-B450-AORUS-M: $
```

6. Слияние hotfix с master

Выполненные команды:

```
git checkout master
git merge hotfix
git branch -d hotfix
```

Результат: Успешное слияние (fast-forward)

```

timur@timur-B450-AORUS-M: $ git checkout master
git merge hotfix
git branch -d hotfix
Переключились на ветку «master»
Обновление d79f8c3..9b8bcd9
Fast-forward
 index.html | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 index.html
Ветка hotfix удалена (была 9b8bcd9).
timur@timur-B450-AORUS-M: $

```

7. Слияние iss53 с master (с конфликтом)

Выполненные команды:

```

git checkout iss53
echo "Финальные изменения" >> index.html
git add index.html
git commit -m "Последние правки"
git checkout master
git merge iss53 # Возник конфликт
nano index.html # Ручное разрешение конфликта
git add index.html
git commit
git branch -d iss53

```

Результат: Конфликт успешно разрешен

Скриншоты:

```

timur@timur-B450-AORUS-M: $ git checkout iss53
echo "Финальные изменения" >> index.html
git add index.html
git commit -m "Последние правки"
git checkout master
git merge iss53
git branch -d iss53
Переключились на ветку «iss53»
[iss53 da9ec3a] Последние правки
 1 file changed, 1 insertion(+)
Переключились на ветку «master»
Автослияние index.html
КОНФЛИКТ (добавление/добавление): Конфликт слияния в index.html
Сбой автоматического слияния; исправьте конфликты, затем зафиксируйте результат.
error: the branch 'iss53' is not fully merged.
If you are sure you want to delete it, run 'git branch -D iss53'
timur@timur-B450-AORUS-M: $

```

```

timur@timur-B450-AORUS-M: $ git status
Текущая ветка: master
У вас есть не слитые пути.
(разрешите конфликты, затем запустите «git commit»)
(используйте «git merge --abort», чтобы остановить операцию слияния)

Не слитые пути:
(используйте «git add <файл>...», чтобы пометить разрешение конфликта)
оба добавлены: index.html

индекс пуст (используйте «git add» и/или «git commit -a»)
timur@timur-B450-AORUS-M: $ nano index.html
timur@timur-B450-AORUS-M: $ git add index.html
timur@timur-B450-AORUS-M: $ git commit
[master d5bfa8f] Merge branch 'iss53'
timur@timur-B450-AORUS-M: $ git branch -d iss53
Ветка iss53 удалена (была da9ec3a).
timur@timur-B450-AORUS-M: $ █

```

Выводы

1. Все этапы работы с ветками выполнены:
 - ☐ Создание веток
 - ☐ Внесение изменений
 - ☐ Слияние веток
2. Успешно разрешен конфликт при слиянии
3. Работа соответствует методическим указаниям

Отчет по выполнению лабораторной работы "Работа с ветками в Git"

1. Подготовка репозитория

Выполнено:

- ☐ Создан файл **test.txt** с содержимым "Тестовый файл"
- ☐ Выполнен первый коммит с сообщением "Тестовый коммит"

Скриншот для отчета:

```

timur@timur-B450-AORUS-M: $ echo "Тестовый файл" > test.txt
git add test.txt
git commit -m "Тестовый коммит"
Текущая ветка: master
нечего коммитить, нет изменений в рабочем каталоге
timur@timur-B450-AORUS-M: $ git log --oneline
commit HEAD master Tag: v1.0.0 new-branch! Добавлен тестовый файл
timur@timur-B450-AORUS-M: $ █

```

2. Работа с ветками

Выполнено:

1. Созданы ветки:
 - ☐ **new-branch**
 - ☐ **testing**
 - ☐ **feature**
 - ☐ **se**
 - ☐ **cl**
2. Просмотр списка веток с информацией о коммитах

Скриншоты:

```
timur@timur-B450-AORUS-M: $ git branch
master
* new-branch
timur@timur-B450-AORUS-M: $ git branch -v
master      29e0b77 Добавлен тестовый файл
* new-branch 29e0b77 Добавлен тестовый файл
timur@timur-B450-AORUS-M: $
```

```
timur@timur-B450-AORUS-M: $ git checkout master
Переключились на ветку «master»
timur@timur-B450-AORUS-M: $ git branch testing
git branch --merged
* master
  new-branch
  testing
timur@timur-B450-AORUS-M: $
```

3. Удаление веток

Выполнено:

- ☐ Удалены ветки:
- ☐ **testing**
- ☐ **new-remote-branch**
- ☐ **se**
- ☐ **cl**

Скриншоты:

```
timur@timur-B450-AORUS-M: $ git branch --no-merged
timur@timur-B450-AORUS-M: $ git branch -d testing
Ветка testing удалена (была 29e0b77).
timur@timur-B450-AORUS-M: $ git remote show myremote
fatal: 'myremote' does not appear to be a git repository
fatal: Не удалось прочитать из внешнего репозитория.
```

Удостоверьтесь, что у вас есть необходимые права доступа и репозиторий существует.

```
timur@timur-B450-AORUS-M: $
```

```
timur@timur-B450-AORUS-M: $ git branch -d new-remote-branch
Ветка new-remote-branch удалена (была 29e0b77).
timur@timur-B450-AORUS-M: $ git branch
* master
  new-branch
timur@timur-B450-AORUS-M: $
```

```
timur@timur-B450-AORUS-M: $ git branch -d cl
git branch -d se
Ветка cl удалена (была 29e0b77).
Ветка se удалена (была 29e0b77).
timur@timur-B450-AORUS-M: $ git branch
  feature
* master
  new-branch
timur@timur-B450-AORUS-M: $
```

4. Проблемные моменты

1. Ошибка подключения к удаленному репозиторию:

- ☐ Не настроено подключение к origin

Скриншот:

```
timur@timur-B450-AORUS-M: $ git branch --no-merged
timur@timur-B450-AORUS-M: $ git branch -d testing
Ветка testing удалена (была 29e0b77).
timur@timur-B450-AORUS-M: $ git remote show myremote
fatal: 'myremote' does not appear to be a git repository
fatal: Не удалось прочитать из внешнего репозитория.

Удостоверьтесь, что у вас есть необходимые права доступа
и репозиторий существует.
timur@timur-B450-AORUS-M: $ █
```

2. Ошибки при перебазировании:

- ☐ Некорректный синтаксис команды rebase
- ☐ Отсутствие изменений для коммита

Скриншоты:

```
timur@timur-B450-AORUS-M: $ git branch
* master
  new-branch
timur@timur-B450-AORUS-M: $ git branch feature
git checkout feature
echo "Изменения в feature" >> test.txt
git commit -am "Добавлены изменения в feature"
git checkout master
git merge feature
Переключились на ветку «feature»
Текущая ветка: feature
нечего коммитить, нет изменений в рабочем каталоге
Переключились на ветку «master»
Уже актуально.
timur@timur-B450-AORUS-M: $ █
```

```

tinur@tinur-B450-AORUS-M: $ git branch se
git checkout se
echo "Изменения в se" >> test.txt
git commit -am "Изменения в se"
git branch cl
git checkout cl
echo "Изменения в cl" >> test.txt
git commit -am "Изменения в cl"
git checkout se
git rebase --onto master se^ cl
git checkout master
git merge cl
git merge se
Переключились на ветку «se»
Текущая ветка: se
нечего коммитить, нет изменений в рабочем каталоге
Переключились на ветку «cl»
Текущая ветка: cl
нечего коммитить, нет изменений в рабочем каталоге
Переключились на ветку «se»
fatal: invalid upstream 'se^'
Переключились на ветку «master»
Уже актуально.
Уже актуально.
tinur@tinur-B450-AORUS-M: $ █

```

5. Итоговое состояние

Текущие ветки:

- ☐ **master** (активна)
- ☐ **feature**
- ☐ **new-branch**

Скриншот:

```

tinur@tinur-B450-AORUS-M: $ git branch -d cl
git branch -d se
Ветка cl удалена (была 29e0b77).
Ветка se удалена (была 29e0b77).
tinur@tinur-B450-AORUS-M: $ git branch
  feature
* master
  new-branch
tinur@tinur-B450-AORUS-M: $ █

```

Выводы и рекомендации:

1. Основные операции выполнены:

- ☐ Создание/удаление веток ✓
- ☐ Просмотр информации ✓
- ☐ Фильтрация веток ✓

ОТЧЕТ

о прохождении интерактивного курса "Learn Git Branching"

Студент: Алиев Тимур Заурович

Группа: ВДКИП 111-прог

1. Введение

Освоены фундаментальные концепции Git:

- Создание коммитов (`git commit -m "Сообщение"`)
- Создание и переключение между ветками (`git branch`, `git checkout`)
- Два метода слияния веток:
 - `git merge` - создание коммита слияния
 - `git rebase` - линейное перенесение истории

2. Навигация по истории коммитов

Изучены продвинутые техники:

- Работа с указателем HEAD (`git checkout C4`)
- Относительные ссылки:
 - `^` - переход к родительскому коммиту
 - `~` - перемещение на несколько коммитов назад
- Отмена изменений:
 - `git reset` - для локальных изменений
 - `git revert` - для удаленных коммитов

3. Управление изменениями

Практическое применение:

- `git cherry-pick` - выборочное копирование коммитов
- Интерактивный ребейз (`git rebase -i`) для:
 - Изменения порядка коммитов
 - Объединения (`squash`)
 - Редактирования сообщений

4. Работа с тегами

- Создание легковесных и аннотированных тегов (git tag)
- Использование тегов для маркировки версий (v1.0, v2.0)

5. Продвинутые сценарии

Решены комплексные задачи:

- Многократный ребейз нескольких веток
- Определение родительских коммитов в сложных ветвлениях
- Восстановление "запутанных" веток с помощью cherry-pick

6. Удаленные репозитории

Полный цикл работы:

- Клонирование (git clone)
- Синхронизация (git fetch, git pull)
- Публикация изменений (git push)
- Разрешение конфликтов:
 - --rebase при расхождении
 - Принудительный сброс (reset --hard)

7. Специальные техники

Освоены:

- Тонкая настройка push/pull с аргументами:
 - git push origin main:feature
 - git pull origin foo:bar
- Управление удаленными ветками:
 - Создание отслеживаемых веток
 - Удаление веток на сервере (git push origin :branch)

Результаты:

- Все 56 упражнений курса выполнены
- Ключевые концепции Git освоены на практике
- Сформировано понимание workflow реальных проектов

Приложения:

1. Скриншоты выполненных заданий (с индивидуальными пометками)
2. Лог выполненных команд

Скрины:

