

Отчёт по лабораторной работе №1

Работа с git

Тимур Дмитриевич Калинин

Содержание

1 Цель работы	8
2 Задание	9
3 Теоретическое введение	10
4 Выполнение лабораторной работы	11
4.1 Создание проекта	11
4.1.1 Создание репозитория	11
4.1.2 Добавление файла в репозиторий	12
4.1.3 Проверка состояния репозитория	12
4.2 Внесение изменений	12
4.2.1 Измените страницу «Hello, World»	12
4.3 Индексация изменений	13
4.3.1 Коммит изменений	13
4.3.2 Добавьте стандартные теги страницы	14
4.3.3 История	16
4.3.4 Получение старых версий	17
4.3.5 Создание тегов версий	18
4.3.6 Переключение по имени тега	19
4.3.7 Просмотр тегов с помощью команды tag	19
4.4 Отмена локальных изменений (до индексации)	20
4.4.1 Переключитесь на ветку master	20
4.4.2 Измените hello.html	20
4.4.3 Отмена изменений в рабочем каталоге	20
4.5 Отмена проиндексированных изменений (перед коммитом)	21
4.5.1 Измените файл и проиндексируйте изменения	21
4.5.2 Выполните сброс буферной зоны	22
4.5.3 Переключитесь на версию коммита	22
4.6 Отмена коммитов	23
4.6.1 Измените файл и сделайте коммит	23
4.6.2 Сделайте коммит с новыми изменениями, отменяющими предыдущие	23
4.6.3 Проверьте лог	24
4.7 Удаление коммитов из ветки	24
4.7.1 Для начала отметьте эту ветку	24
4.7.2 Сброс коммитов к предшествующим коммиту Oops	24

4.8 Удаление тега oops	25
4.9 Внесение изменений в коммиты	26
4.10 Перемещение файлов	27
4.11 Подробнее о структуре	28
4.12 Git внутри: Каталог .git	29
4.13 Работа непосредственно с объектами git	30
4.14 Создание ветки	32
4.15 Изменения в ветке master	34
4.16 Слияние	35
4.17 Создание конфликта	36
4.18 Клонирование репозиториев	40
5 Выводы	45

List of Figures

4.1	Создание каталога	11
4.2	Инициализация репозитория	11
4.3	Добавление файла в репозиторий	12
4.4	Проверка состояния репозитория	12
4.5	Добавление тегов	13
4.6	Проверка состояния	13
4.7	Индексация	13
4.8	Коммит	14
4.9	Модификация файла	14
4.10	Индексация	14
4.11	Изменение файла	15
4.12	Статус	15
4.13	Коммит	15
4.14	Индексация второго изменения	16
4.15	Коммит второго изменения	16
4.16	История изменений	17
4.17	Однострочный формат истории	17
4.18	Хэши	17
4.19	Просмотр hello.html	18
4.20	Ветка master	18
4.21	Создание тегов	18
4.22	Возвращение к тегу предыдущей версии	18
4.23	Просмотр файла hello.html	19
4.24	Добавление тега	19
4.25	Переключение между тегами	19
4.26	Просмотр тегов	19
4.27	Переключение на master	20
4.28	Изменение файла hello.html	20
4.29	Использование git checkout	21
4.30	Нежелательный комментарий	21
4.31	Индексация	22
4.32	Сброс	22
4.33	Переключение	22
4.34	Изменение файла	23
4.35	Коммит	23
4.36	Отменяющий коммит	23
4.37	Проверка лога	24

4.38 Тегирование последнего коммита	24
4.39 Сброс ветки	25
4.40 Проверка лога	25
4.41 Удаление тега	25
4.42 Просмотр лога	25
4.43 Добавление комментария	26
4.44 Коммит	26
4.45 Добавление email	26
4.46 Изменение предыдущего коммита	27
4.47 Просмотр истории	27
4.48 Перемещение файлов	27
4.49 Создание нового коммита	28
4.50 Создание файла index.html	28
4.51 index.html	28
4.52 index.html	28
4.53 index.html	29
4.54 .git/objects	29
4.55 Внутренний каталог	29
4.56 .git/config	29
4.57 Каталоги с тегами, ветками и т.д.	30
4.58 Каталог с ссылкой на текущую ветку	30
4.59 Последний коммит	30
4.60 cat-file	31
4.61 cat-file	31
4.62 Оригинальный hello.html	31
4.63 Ветка	32
4.64 css файл	32
4.65 Коммит	32
4.66 Обновление hello.html	33
4.67 Коммит	33
4.68 Обновление index.html	33
4.69 Коммит	34
4.70 Переключение	34
4.71 Ветка style	34
4.72 README	35
4.73 Коммит	35
4.74 Ветки	35
4.75 Слияние	36
4.76 Графический лог	36
4.77 Внесение конфликта	36
4.78 Коммит	37
4.79 Просмотр веток	37
4.80 Ветка style	37
4.81 Конфликт	38

4.82 Конфликт	38
4.83 Коммит	38
4.84 Лог	39
4.85 Сброс ветки style	39
4.86 Сброс ветки master	39
4.87 Перебазирование	39
4.88 Слияние	40
4.89 Клонирование	40
4.90 Просмотр коммитов	40
4.91 Удаленный репозиторий	41
4.92 Список веток	41
4.93 Изменение оригинального репозитория	41
4.94 Коммит	41
4.95 Извлечение изменений	42
4.96 Слияние изменений	42
4.97 Ветка наблюдения	42
4.98 Чистый репозиторий	43
4.99 Добавление удаленного репозитория	43
4.100 Изменение README	43
4.101 Отправка изменений	44
4.102 Извлечение новых изменений	44

List of Tables

1 Цель работы

Познакомиться с системой контроля версий git. Научиться создавать отчеты при помощи языка Markdown.

2 Задание

Требуется познакомиться с системой контроля версий git. Опробовать механизмы создания коммитов, тегов. Изучить ветки, слияния, перебазирования. Начать работать с удаленными репозиториями.

3 Теоретическое введение

4 Выполнение лабораторной работы

4.1 Создание проекта

Начните работу в пустом рабочем каталоге с создания пустого каталога с именем hello, затем войдите в него и создайте там файл с именем hello.html (Рис. 4.1).



```
MINGW64:/c/Users/TIMUR/work/hello
TIMUR@DESKTOP-OG009CH MINGW64 ~/work
$ mkdir hello

TIMUR@DESKTOP-OG009CH MINGW64 ~/work
$ cd hello

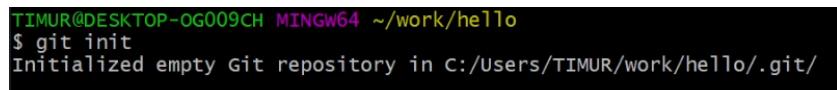
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello
$ touch hello.html

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello
$ echo "Hello, world!" > hello.html
```

Figure 4.1: Создание каталога

4.1.1 Создание репозитория

Чтобы создать git репозиторий из этого каталога, выполните команду git init (Рис. 4.2).



```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello
$ git init
Initialized empty Git repository in c:/Users/TIMUR/work/hello/.git/
```

Figure 4.2: Инициализация репозитория

4.1.2 Добавление файла в репозиторий

Добавим файл в репозиторий (Рис. 4.3).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git add hello.html

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git commit -m "Initial commit"
[master (root-commit) a3c77b4] Initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 hello.html
```

Figure 4.3: Добавление файла в репозиторий

4.1.3 Проверка состояния репозитория

Проверим текущее состояние репозитория (Рис. 4.4).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git status
on branch master
nothing to commit, working tree clean
```

Figure 4.4: Проверка состояния репозитория

Команда проверки состояния сообщает, что коммитить нечего.

4.2 Внесение изменений

4.2.1 Измените страницу «Hello, World»

Добавим кое-какие HTML-теги к нашему приветствию (Рис. 4.5).

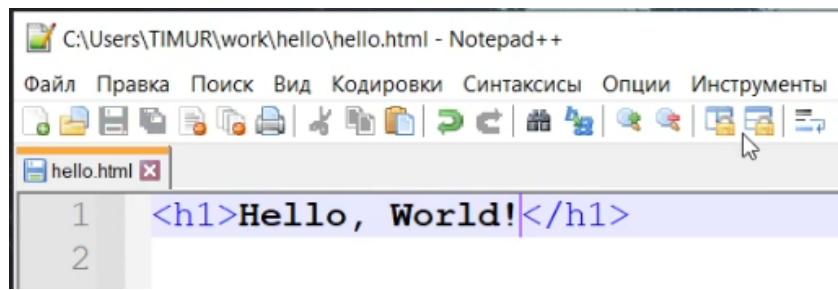


Figure 4.5: Добавление тегов

Проверьте состояние рабочего каталога (Рис. 4.6).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   hello.html

no changes added to commit (use "git add" and/or "git commit -a")
```

Figure 4.6: Проверка состояния

4.3 Индексация изменений

Теперь выполните команду git, чтобы проиндексировать изменения. Проверьте состояние (Рис. 4.7).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git add hello.html

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   hello.html
```

Figure 4.7: Индексация

4.3.1 Коммит изменений

Сделайте коммит и проверьте состояние (Рис. 4.8).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git commit
```

Figure 4.8: Коммит

4.3.2 Добавьте стандартные теги страницы

Измените страницу «Hello, World», чтобы она содержала стандартные теги (Рис. 4.9).

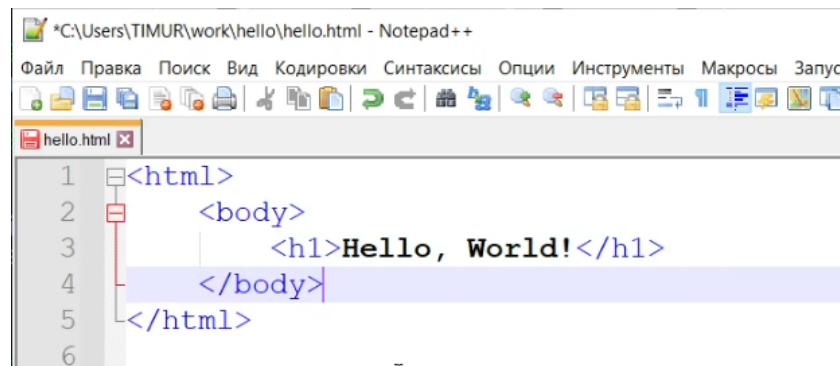


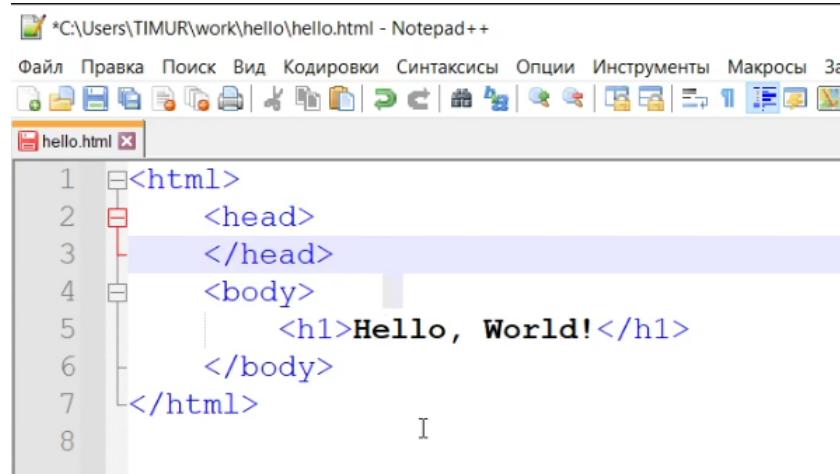
Figure 4.9: Модификация файла

Теперь добавьте это изменение в индекс git (Рис. 4.10).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git add hello.html
```

Figure 4.10: Индексация

Теперь добавьте заголовки HTML (секцию head) к странице «Hello, World» (Рис. 4.11).



```
*C:\Users\TIMUR\work\hello\hello.html - Notepad++
Файл Правка Поиск Вид Кодировка Синтаксисы Опции Инструменты Макросы За
[File Edit Search View Encoding Syntax Options Tools Macros Help]
hello.html x
1 <html>
2   <head>
3     </head>
4   <body>
5     <h1>Hello, World!</h1>
6   </body>
7 </html>
I
```

Figure 4.11: Изменение файла

Проверьте текущий статус (Рис. 4.12):

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   hello.html

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   hello.html
```

Figure 4.12: Статус

Произведите коммит проиндексированного изменения (значение по умолчанию), а затем еще раз проверьте состояние (Рис. 4.13).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git commit -m "Added standard HTML page tags"
[master 576b12d] Added standard HTML page tags
 1 file changed, 5 insertions(+), 1 deletion(-)

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   hello.html

no changes added to commit (use "git add" and/or "git commit -a")
```

Figure 4.13: Коммит

Теперь добавьте второе изменение в индекс, а затем проверьте состояние с

помощью команды git status (Рис. 4.14).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git add .

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git stats
git: 'stats' is not a git command. See 'git --help'.

The most similar command is
      status

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   hello.html
```

Figure 4.14: Индексация второго изменения

Сделайте коммит второго изменения (Рис. 4.15).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git commit -m "Added HTML header"
[master 92caf88] Added HTML header
  1 file changed, 2 insertions(+)
```

Figure 4.15: Коммит второго изменения

4.3.3 История

Получим список произведенных изменений (Рис. 4.16):

```

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git log
commit 92caf88022b08f0a33658df1565ba526ba4a0482 (HEAD -> master)
Author: tdkalinin <timurcalinin@yandex.ru>
Date:   Tue Feb 8 18:44:18 2022 +0300

    Added HTML header

commit 576b12d709749639664b3be1460f2cda5b1f94bc
Author: tdkalinin <timurcalinin@yandex.ru>
Date:   Tue Feb 8 18:42:47 2022 +0300

    Added standard HTML page tags

commit 3f46d581d0763d2ad343e9badf54bdd1c9769226
Author: tdkalinin <timurcalinin@yandex.ru>
Date:   Tue Feb 8 18:38:52 2022 +0300

    Added h1 tag

commit a3c77b4a94cff2c5dfd3f34c185606741cca238a
Author: tdkalinin <timurcalinin@yandex.ru>
Date:   Tue Feb 8 18:34:24 2022 +0300

Initial commit

```

Figure 4.16: История изменений

Однострочный формат истории (Рис. 4.17):

```

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git log --pretty=oneline
92caf88022b08f0a33658df1565ba526ba4a0482 (HEAD -> master) Added HTML header
576b12d709749639664b3be1460f2cda5b1f94bc Added standard HTML page tags
3f46d581d0763d2ad343e9badf54bdd1c9769226 Added h1 tag
a3c77b4a94cff2c5dfd3f34c185606741cca238a Initial commit

```

Figure 4.17: Однострочный формат истории

4.3.4 Получение старых версий

Получите хэши предыдущих версий. Перейдем к первому коммиту по его хэшу (Рис. 4.18 , Рис. 4.19).

```

commit a3c77b4a94cff2c5dfd3f34c185606741cca238a
Author: tdkalinin <timurcalinin@yandex.ru>
Date:   Tue Feb 8 18:34:24 2022 +0300

Initial commit

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git checkout a3c77

```

Figure 4.18: Хэши

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello ((a3c77b4...))
$ cat hello.html
Hello, world!
```

Figure 4.19: Просмотр hello.html

Вернитесь к последней версии в ветке master (Рис. 4.20)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello ((a3c77b4...))
$ git checkout master
Previous HEAD position was a3c77b4 Initial commit
Switched to branch 'master'

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ cat hello.html
<html>
    <head>
    </head>
    <body>
        <h1>Hello, world!</h1>
    </body>
</html>
```

Figure 4.20: Ветка master

4.3.5 Создание тегов версий

Создайте тег первой версии(Рис. 4.21)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git tag v1
```

Figure 4.21: Создание тегов

Вернемся к предыдущей тегу версии(Рис. 4.22, Рис. 4.23)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git checkout v1^
Note: switching to 'v1^'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.
```

Figure 4.22: Возвращение к тегу предыдущей версии

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello ((576b12d...))
$ cat hello.html
<html>
    <body>
        <h1>Hello, world!</h1>
    </body>
</html>
```

Figure 4.23: Просмотр файла hello.html

Дадим ей тег(Рис. 4.24)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello ((576b12d...))
$ git tag v1-beta
```

Figure 4.24: Добавление тега

4.3.6 Переключение по имени тега

Теперь попробуйте попереключаться между двумя отмеченными версиями (Рис. 4.25).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello ((v1-beta))
$ git checkout v1
Previous HEAD position was 576b12d Added standard HTML page tags
HEAD is now at 92caf88 Added HTML header

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello ((v1))
$ git checkout v1-beta
Previous HEAD position was 92caf88 Added HTML header
HEAD is now at 576b12d Added standard HTML page tags
```

Figure 4.25: Переключение между тегами

4.3.7 Просмотр тегов с помощью команды tag

Посмотрим теги (Рис. 4.26)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello ((v1-beta))
$ git tag
v1
v1-beta
```

Figure 4.26: Просмотр тегов

4.4 Отмена локальных изменений (до индексации)

4.4.1 Переключитесь на ветку master

Переключимся на ветку master(Рис. 4.27)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello ((v1-beta))
$ git checkout master
Previous HEAD position was 576b12d Added standard HTML page tags
Switched to branch 'master'
```

Figure 4.27: Переключение на master

4.4.2 Измените hello.html

Изменим hello.html(Рис. 4.28)



```
hello.html ×
C: > Users > TIMUR > work > hello > hello.html > html > body
1  <html>
2   |  <head>
3   |  </head>
4   |  <body>
5   |    <h1>Hello, World!</h1>
6   |    |  <!-- This is a bad comment. We want to revert it. --&gt;
7   |    |  &lt;/body&gt;
8   |  &lt;/html&gt;
9</pre>
```

Figure 4.28: Изменение файла hello.html

4.4.3 Отмена изменений в рабочем каталоге

Используйте команду git checkout для переключения версии файла hello.html в репозитории (Рис. 4.29).

```

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git checkout hello.html
Updated 1 path from the index

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git status
on branch master
nothing to commit, working tree clean

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ cat hello.html
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>

```

Figure 4.29: Использование git checkout

4.5 Отмена проиндексированных изменений (перед коммитом)

4.5.1 Измените файл и проиндексируйте изменения

Внесите изменение в файл hello.html в виде нежелательного комментария (Рис. 4.30)

```

hello.html ×
C: > Users > TIMUR > work > hello > hello.html > html > head
1  <html>
2  |   <head>
3  |   |   /* This is an unwanted but staged comment */
4  |   |   </head>
5  |   |   <body>
6  |   |   |   <h1>Hello, World!</h1>
7  |   |   </body>
8  |   </html>
9

```

Figure 4.30: Нежелательный комментарий

Проиндексируйте это изменение. Проверьте состояние нежелательного изменения (Рис. 4.31).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git add hello.html

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   hello.html
```

Figure 4.31: Индексация

4.5.2 Выполните сброс буферной зоны

Команда git reset сбрасывает буферную зону к HEAD. Это очищает буферную зону от изменений, которые мы только что проиндексировали (Рис. 4.32).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git reset HEAD hello.html
Unstaged changes after reset:
M      hello.html
```

Figure 4.32: Сброс

4.5.3 Переключитесь на версию коммита

Переключимся (Рис. 4.33)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git checkout hello.html
Updated 1 path from the index

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git status
on branch master
nothing to commit, working tree clean
```

Figure 4.33: Переключение

4.6 Отмена коммитов

4.6.1 Измените файл и сделайте коммит

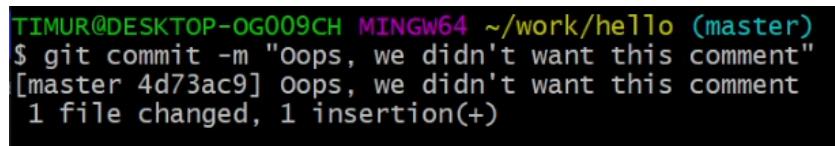
Измените файл hello.html на следующий (Рис. 4.34).



```
hello.html X
C: > Users > TIMUR > work > hello > hello.html > html > body
1   <html>
2     <head>
3     </head>
4     <body>
5       <h1>Hello, World!</h1>
6       <!-- This is an unwanted but committed change -->
7     </body>
8   </html>
9
```

Figure 4.34: Изменение файла

Выполните (Рис. 4.35):

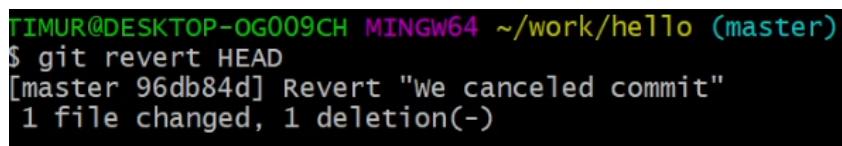


```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git commit -m "Oops, we didn't want this comment"
[master 4d73ac9] oops, we didn't want this comment
 1 file changed, 1 insertion(+)
```

Figure 4.35: Коммит

4.6.2 Сделайте коммит с новыми изменениями, отменяющими предыдущие

Чтобы отменить коммит, нам необходимо сделать коммит, который удаляет изменения, сохраненные нежелательным коммитом (Рис. 4.36).



```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git revert HEAD
[master 96db84d] Revert "We canceled commit"
 1 file changed, 1 deletion(-)
```

Figure 4.36: Отменяющий коммит

4.6.3 Проверьте лог

Проверка лога показывает нежелательные и отмененные коммиты в наш репозиторий (Рис. 4.37).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git log
commit 96db84deb9d3d8173621a367bac2444dc5696263 (HEAD -> master)
Author: tdkalinin <timurcalinin@yandex.ru>
Date:   Tue Feb 8 18:58:19 2022 +0300

    Revert "we canceled commit"

    This reverts commit 4d73ac95ad992d9c9e74206f6cc55d84b73b48cd.

commit 4d73ac95ad992d9c9e74206f6cc55d84b73b48cd
Author: tdkalinin <timurcalinin@yandex.ru>
Date:   Tue Feb 8 18:57:36 2022 +0300

    oops, we didn't want this comment
```

Figure 4.37: Проверка лога

4.7 Удаление коммитов из ветки

4.7.1 Для начала отметьте эту ветку

Отметим последний коммит тегом (Рис. 4.38)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git tag oops
```

Figure 4.38: Тегирование последнего коммита

4.7.2 Сброс коммитов к предшествующим коммиту Oops

Сбросим ветку до точки v1 (Рис. 4.39, Рис. 4.40). Мы все еще видим ошибочные коммиты

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git reset --hard v1
HEAD is now at 92caf88 Added HTML header

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git log
commit 92caf88022b08f0a33658df1565ba526ba4a0482 (HEAD -> master, tag: v1)
Author: tdkalinin <timurcalinin@yandex.ru>
Date:   Tue Feb 8 18:44:18 2022 +0300

    Added HTML header
```

Figure 4.39: Сброс ветки

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git log --all
commit 96db84deb9d3d8173621a367bac2444dc5696263 (tag: oops)
Author: tdkalinin <timurcalinin@yandex.ru>
Date:   Tue Feb 8 18:58:19 2022 +0300

    Revert "We canceled commit"

    This reverts commit 4d73ac95ad992d9c9e74206f6cc55d84b73b48cd.

commit 4d73ac95ad992d9c9e74206f6cc55d84b73b48cd
Author: tdkalinin <timurcalinin@yandex.ru>
Date:   Tue Feb 8 18:57:36 2022 +0300

    Oops, we didn't want this comment
```

Figure 4.40: Проверка лога

4.8 Удаление тега oops

Удалим тег oops(Рис. 4.41, Рис. 4.42)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git tag -d oops
Deleted tag 'oops' (was 96db84d)
```

Figure 4.41: Удаление тега

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git log --all
commit 92caf88022b08f0a33658df1565ba526ba4a0482 (HEAD -> master, tag: v1)
Author: tdkalinin <timurcalinin@yandex.ru>
Date:   Tue Feb 8 18:44:18 2022 +0300

    Added HTML header
```

Figure 4.42: Просмотр лога

4.9 Внесение изменений в коммиты

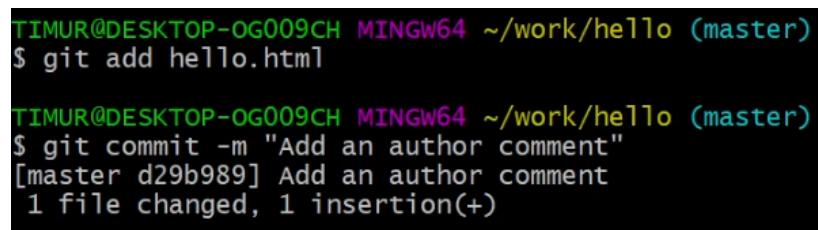
Добавим на страницу комментарий (Рис. 4.43)



```
<!-- Author: Timur D. Kalinin -->
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

Figure 4.43: Добавление комментария

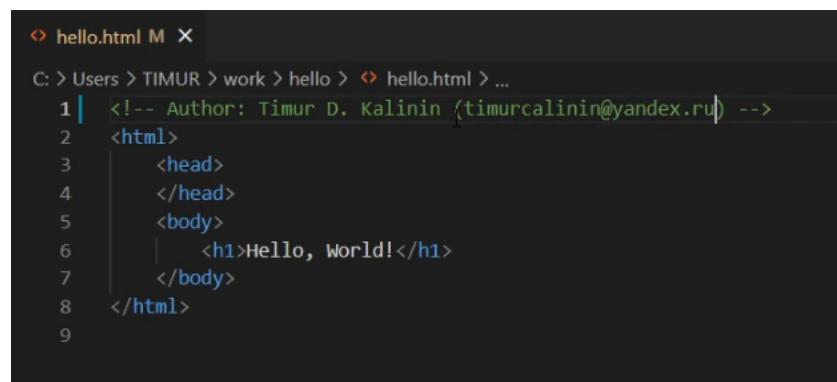
Сделаем коммит (Рис. 4.44)



```
$ git add hello.html
$ git commit -m "Add an author comment"
[master d29b989] Add an author comment
 1 file changed, 1 insertion(+)
```

Figure 4.44: Коммит

Добавим email на страницу (Рис. 4.45)



```
<!-- Author: Timur D. Kalinin timurcalinin@yandex.ru -->
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

Figure 4.45: Добавление email

Изменим предыдущий коммит (Рис. 4.46)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git add hello.html

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git commit --amend -m "Add an author/email comment"
[master 1383ea7] Add an author/email comment
  Date: Tue Feb 8 19:04:45 2022 +0300
    1 file changed, 1 insertion(+)

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$
```

Figure 4.46: Изменение предыдущего коммита

Посмотрим историю(Рис. 4.47)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git log
commit 1383ea7af2d8ec8671726ff7f01bea25e034e190 (HEAD -> master)
Author: tdkalinin <timurcalinin@yandex.ru>
Date:   Tue Feb 8 19:04:45 2022 +0300

  Add an author/email comment
```

Figure 4.47: Просмотр истории

4.10 Перемещение файлов

Перемесим hello.html в каталог lib (Рис. 4.48)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ mkdir lib

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git mv hello.html lib

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    renamed:    hello.html -> lib/hello.html
```

Figure 4.48: Перемещение файлов

Сделаем новый коммит (Рис. 4.49)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git commit -m "Moved hello.html to lib"
[master a7f0755] Moved hello.html to lib
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename hello.html => lib/hello.html (100%)
```

Figure 4.49: Создание нового коммита

4.11 Подробнее о структуре

Создадим файл index.html (Рис. 4.50, Рис. 4.49)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ touch index.html
```

Figure 4.50: Создание файла index.html



```
<index.html U X>
C: > Users > TIMUR > work > hello > <index.html> ↗ html
1   <html>
2     <body>
3       <iframe src="lib/hello.html" width="200" height="200"></iframe>
4     </body>
5   </html>
```

Figure 4.51: index.html

Добавим файл и сделаем коммит (Рис. 4.52)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git add index.html

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git commit -m "Added index.html"
[master 6261c2a] Added index.html
 1 file changed, 5 insertions(+)
 create mode 100644 index.html
```

Figure 4.52: index.html

4.12 Git внутри: Каталог .git

Посмотрим каталог .git (Рис. 4.53)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ ls -c .git
COMMIT_EDITMSG      HEAD      config      hooks/    info/    objects/    refs/
COMMIT_EDITMSG.bak  ORIG_HEAD  description  index     logs/    packed-refs
```

Figure 4.53: index.html

Откроем .git/objects (Рис. 4.54)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ ls -c .git/objects
13/  23/  4d/  5a/  65/  8b/  93/  96/  a4/  aa/  d0/  d4/  f7/  pack/
1c/  3f/  57/  62/  69/  92/  95/  a3/  a7/  af/  d2/  da/  info/
```

Figure 4.54: .git/objects

Откроем один из внутренних каталогов (Рис. 4.55)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ ls -C .git/objects/13
83ea7af2d8ec8671726ff7f01bea25e034e190
```

Figure 4.55: Внутренний каталог

Откроем .git/config (Рис. 4.56)

```
$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = false
    bare = false
    logallrefupdates = true
    symlinks = false
    ignorecase = true
```

Figure 4.56: .git/config

Посмотрим каталоги с тегами, ветками и т.д. (Рис. 4.57)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ ls .git/refs
heads/ tags/

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ ls .git/refs/heads
master

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ ls .git/refs/tags
v1 v1-beta

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ cat .git/refs/tags/v1
92caf88022b08f0a33658df1565ba526ba4a0482
```

Figure 4.57: Каталоги с тегами, ветками и т.д.

Посмотрим каталог с ссылкой на текущую ветку (Рис. 4.58)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ cat .git/HEAD
ref: refs/heads/master
```

Figure 4.58: Каталог с ссылкой на текущую ветку

4.13 Работа непосредственно с объектами git

Найдем последний коммит (Рис. 4.59):

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git log --max-count=1
commit 6261c2af015bc593bf19884d0c136984bd57b451 (HEAD -> master)
Author: tdkalinin <timurcalinin@yandex.ru>
Date:   Tue Feb 8 19:12:36 2022 +0300

    Added index.html

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
```

Figure 4.59: Последний коммит

Выполните (Рис. 4.60):

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git log --max-count=1
commit 6261c2af015bc593bf19884d0c136984bd57b451 (HEAD -> master)
Author: tdkalinin <timurcalinin@yandex.ru>
Date:   Tue Feb 8 19:12:36 2022 +0300

        Added index.html

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git cat-file -t 6261c
commit

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git cat-file -p 6261c
tree dad4d93613b9fc9a9b77c84b855dd014c510554a4
parent a7f075535a4e3de2274c713fbe62501d705d54d
author tdkalinin <timurcalinin@yandex.ru> 1644336756 +0300
committer tdkalinin <timurcalinin@yandex.ru> 1644336756 +0300

Added index.html
```

Figure 4.60: cat-file

Выполните (Рис. 4.61):

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git cat-file -p dad4d9
100644 blob 69c96dcfb3b74c0d65000149d9832b459bea26e7    index.html
040000 tree 5aba16bc338dc667c6f086c1bd74e3f9d07951b1    lib

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git cat-file -p 5aba16
100644 blob 8b0d2bf78f9626401612c744ef7ca58b14e83ba0    hello.html

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git cat-file -p 8b0d2b
<!-- Author: Timur D. Kalinin (timurcalinin@yandex.ru) -->
<html>
    <head>
    </head>
    <body>
        <h1>Hello, world!</h1>
    </body>
</html>
```

Figure 4.61: cat-file

Смотрите, удастся ли вам найти оригинальный файл hello.html с самого первого коммита вручную по ссылкам SHA1 хэша в последнем коммите (Рис. 4.62).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git cat-file -p af5626b
Hello, world!
```

Figure 4.62: Оригинальный hello.html

4.14 Создание ветки

Создадим ветку и css файл (Рис. 4.63, Рис. 4.64).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git checkout -b style
Switched to a new branch 'style'

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (style)
$ git status
On branch style
nothing to commit, working tree clean

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (style)
$ touch lib/style.css
```

Figure 4.63: Ветка

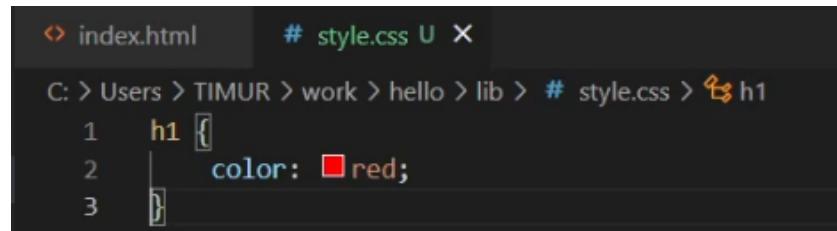


Figure 4.64: css файл

Выполним (Рис. 4.65):

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (style)
$ git add lib/style.css

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (style)
$ git commit -m "Added css stylesheet"
[style f04f42c] Added css stylesheet
 1 file changed, 3 insertions(+)
 create mode 100644 lib/style.css
```

Figure 4.65: Коммит

Обновите файл hello.html (Рис. 4.66)

```
index.html      # style.css      hello.html
C: > Users > TIMUR > work > hello > lib > hello.html > html > head > link
1   <!-- Author: Timur D. Kalinin (timurcalinin@yandex.ru) -->
2   <html>
3     <head>
4       <link type="text/css" rel="stylesheet"
5         media="all" href="style.css" />
6     </head>
7     <body>
8       <h1>Hello, World!</h1>
9     </body>
10    </html>
11
```

Figure 4.66: Обновление hello.html

Выполните (Рис. 4.67):

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (style)
$ git add lib/hello.html

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (style)
$ git commit -m "Hello uses style.css"
[style 8b92f91] Hello uses style.css
 1 file changed, 1 insertion(+)
```

Figure 4.67: КОММИТ

Обновите файл index.html (Рис. 4.68)

```
index.html      # style.css      hello.html
C: > Users > TIMUR > work > hello > index.html > html > head > link
1   <html>
2     <head>
3       <link type="text/css" rel="stylesheet" media="all" href="lib/style.
4         cs" />
5     </head>
6     <body>
7       <iframe src="lib/hello.html" width="200" height="200"></iframe>
8     </body>
9   </html>
```

Figure 4.68: Обновление index.html

Выполните (Рис. 4.69):

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (style)
$ git add index.html

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (style)
$ git commit -m "Updated index.html"
[style 2dd67dc] Updated index.html
 1 file changed, 3 insertions(+)
```

Figure 4.69: Коммит

Используйте команду git checkout для переключения между ветками (Рис. 4.70):

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (style)
$ git checkout master
Switched to branch 'master'

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)|
$ cat lib/hello.html
<!-- Author: Timur D. Kalinin (timurcalinin@yandex.ru) -->
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

Figure 4.70: Переключение

Вернемся к ветке style. Выполните (Рис. 4.71):

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git checkout style
switched to branch 'style'

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (style)|
$ cat lib/hello.html
<!-- Author: Timur D. Kalinin (timurcalinin@yandex.ru) -->
<html>
  <head>
    <link type="text/css" rel="stylesheet" media="all" href="style.css"
  >
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

Figure 4.71: Ветка style

4.15 Изменения в ветке master

Создайте файл README в ветке master. Выполните (Рис. 4.72):

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (style)
$ git checkout master
Switched to branch 'master'

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ echo "This is the Hello world example from the git tutorial" > README.md
```

Figure 4.72: README

Сделайте коммит изменений README.md в ветку master (Рис. 4.73).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git add README.md

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git commit -m "Added README"
[master c230d56] Added README
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

Figure 4.73: КОММИТ

Просмотрим ветки (Рис. 4.74)

```
* commit c230d56e678db660bffe0e76b1d2bc32e128fb08 (HEAD -> master)
| Author: tdkalinin <timurcalinin@yandex.ru>
| Date:   Tue Feb 8 19:38:45 2022 +0300
|
|       Added README
|
* commit 2dd67dce44a87fe41c21ed662d53d6007de45d9e (style)
| Author: tdkalinin <timurcalinin@yandex.ru>
| Date:   Tue Feb 8 19:35:21 2022 +0300
|
|       Updated index.html
|
* commit 8b92f91cc53bf3c7287c0ed8c8665f642acb9663
| Author: tdkalinin <timurcalinin@yandex.ru>
| Date:   Tue Feb 8 19:32:51 2022 +0300
|
|       Hello uses style.css
|
* commit f04f42ca76ac0b48083a47c6ab082554a14eab56
| Author: tdkalinin <timurcalinin@yandex.ru>
| Date:   Tue Feb 8 19:31:31 2022 +0300
|
|       Added css stylesheet
|
* commit 6261c2af015bc593bf19884d0c136984bd57b451
|:
```

Figure 4.74: Ветки

4.16 Слияние

Сольем ветки master и style (Рис. 4.75, Рис. 4.76).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git checkout style
Switched to branch 'style'

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (style)
$ git merge master
Merge made by the 'ort' strategy.
 README.md | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

Figure 4.75: Слияние

```
$ git log --graph --all
* commit abafaa31ba2bfc147d345133bd6dca566e6050e3 (HEAD -> style)
| Merge: 2dd67dc c230d56
| Author: tdkalinin <timurcalinin@yandex.ru>
| Date:   Tue Feb 8 19:40:39 2022 +0300
|
|       Merge branch 'master' into style
|
* commit c230d56e678db660bffe0e76b1d2bc32e128fb08 (master)
| Author: tdkalinin <timurcalinin@yandex.ru>
| Date:   Tue Feb 8 19:38:45 2022 +0300
|
|       Added README
```

Figure 4.76: Графический лог

4.17 Создание конфликта

Вернемся в ветку master и внесем следующие изменения (Рис. 4.77):

```
1  <!-- Author: Timur D. Kalinin (timurcalinin@yandex.ru) -->
2  <html>
3      <head>
4          |  <!-- no style -->
5          |  </head>
6          <body>
7              |  <h1>Hello, World!</h1>
8          </body>
9      </html>
10
```

Figure 4.77: Внесение конфликта

Закоммитим (Рис. 4.78)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git add lib/hello.html

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git commit -m "Life is great"
[master a6f124c] Life is great
 1 file changed, 1 insertion(+)
```

Figure 4.78: Коммит

Просмотрим ветки (Рис. 4.79)

```
$ git log --graph --all
* commit a6f124c4b4b24bc74846ed...
| Author: tdkalinin <timurcalinin@gmail.com>
| Date:   Tue Feb 8 19:44:04 2022 +0300
|
|     Life is great
|
* commit abafaa31ba2bfc147d...
```

Figure 4.79: Просмотр веток

Теперь вернемся к ветке style и попытаемся объединить ее с новой веткой master (Рис. 4.80, Рис. 4.81).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git checkout style
Switched to branch 'style'

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (style)
$ git merge master
Auto-merging lib/hello.html
CONFLICT (content): Merge conflict in lib/hello.html
Automatic merge failed; fix conflicts and then commit the result.
```

Figure 4.80: Ветка style

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git log --graph --all
* commit a6f124c4b4b24bc74846ed528196d9f6ac12116d (HEAD -> master)
| Author: tdkalinin <timurcalinin@yandex.ru>
| Date:   Tue Feb 8 19:44:04 2022 +0300
|
|   Life is great
|
* commit abafaa31ba2bfc147d345133bd6dca566e6050e3 (style)
| Merge: 2dd67dc c230d56
| Author: tdkalinin <timurcalinin@yandex.ru>
| Date:   Tue Feb 8 19:40:39 2022 +0300
|
|   Merge branch 'master' into style
```

Figure 4.81: Конфликт

Необходимо разрешить конфликт (Рис. 4.82)

```
C: > Users > TIMUR > work > hello > lib > hello.html > html > body
1   <!-- Author: Timur D. Kalinin (timurcalinin@yandex.ru) -->
2   <html>
3   |   <head>
4   |   |   Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
4   |   <<<< HEAD (Current Change)
5   |   |   <link type="text/css" rel="stylesheet" media="all" href="style.css" />
6   =====
7   |   |   <!-- no style -->
8   >>>> master (Incoming Change)
9   |   </head>
10  |   <body>
11  |   |   <h1>Hello, World!</h1>
12  |   </body>
13  </html>
14
```

Figure 4.82: Конфликт

Сделаем коммит (Рис. 4.83, Рис. 4.84)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (style|MERGING)
$ git add lib/hello.html

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (style|MERGING)
$ git commit -m "Merged master fixed conflict"
[style 694d5e2] Merged master fixed conflict
```

Figure 4.83: КОММИТ

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (style)
$ git log --graph
* commit 694d5e2226212e58949f3e5ecdc0694a7b3ad517 (HEAD -> style)
| Merge: abafaa3 a6f124c
| Author: tdkalinin <timurcalinin@yandex.ru>
| Date:   Tue Feb 8 19:48:22 2022 +0300
|
|       Merged master fixed conflict
|
* commit a6f124c4b4b24bc74846ed528196d9f6ac12116d (master)
| Author: tdkalinin <timurcalinin@yandex.ru>
| Date:   Tue Feb 8 19:44:04 2022 +0300
|
|       Life is great
|
* commit abafaa31ba2bfc147d345133bd6dca566e6050e3
| Merge: 2dd67dc c230d56
| Author: tdkalinin <timurcalinin@yandex.ru>
| Date:   Tue Feb 8 19:40:39 2022 +0300
|
|       Merge branch 'master' into style
```

Figure 4.84: Лог

Теперь попробуем сделать перебазирование. Для начала сбросим ветку style (Рис. 4.85).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (style)
$ git reset --hard 2dd67d
```

Figure 4.85: Сброс ветки style

Теперь сбросим ветку master (Рис. 4.86)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git reset --hard 6261c2a
HEAD is now at 6261c2a Added index.html
```

Figure 4.86: Сброс ветки master

Выполним перебазирование (Рис. 4.87).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git checkout style
Switched to branch 'style'

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (style)
$ git rebase master
Successfully rebased and updated refs/heads/style.
```

Figure 4.87: Перебазирование

Теперь сольем style в master (Рис. 4.88)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (style)
$ git checkout master
Switched to branch 'master'

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git merge style
Updating b14e3f6..cff8374
Fast-forward
 index.html      | 3 +++
 lib/hello.html  | 1 +
 lib/style.css   | 3 +++
 3 files changed, 7 insertions(+)
 create mode 100644 lib/style.css
```

Figure 4.88: Слияние

4.18 Клонирование репозиториев

Перейдем в рабочий каталог. Создадим клон репозитория (Рис. 4.89).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work
$ git clone hello cloned_hello
Cloning into 'cloned_hello'...
done.
```

Figure 4.89: Клонирование

Посмотрим коммиты (Рис. 4.90).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/cloned_hello (master)
$ git log --all
commit cff837451ea8cd2b9256860d9e4360d755b9370b (HEAD -> master, origin/style, origin/master, origin/HEAD)
Author: tdkalinin <timurcalinin@yandex.ru>
Date:   Tue Feb 8 19:35:21 2022 +0300

    Updated index.html
```

Figure 4.90: Просмотр коммитов

Мы видим, что клонированный репозиторий знает об имени по умолчанию удаленного репозитория (Рис. 4.91).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/cloned_hello (master)
$ git remote show origin
* remote origin
  Fetch URL: C:/Users/TIMUR/work/hello
  Push URL: C:/Users/TIMUR/work/hello
  HEAD branch: master
  Remote branches:
    master tracked
      style tracked
  Local branch configured for 'git pull':
    master merges with remote master
  Local ref configured for 'git push':
    master pushes to master (up to date)
```

Figure 4.91: Удаленный репозиторий

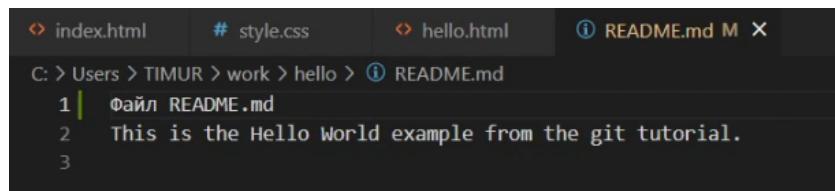
Посмотрим на список веток (Рис. 4.92).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/cloned_hello (master)
$ git branch
* master

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/cloned_hello (master)
$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
  remotes/origin/style
```

Figure 4.92: Список веток

Изменим оригинальный репозиторий (Рис. 4.93).



```
index.html      # style.css      hello.html      README.md X
C: > Users > TIMUR > work > hello > README.md
1 |  Файл README.md
2   This is the Hello World example from the git tutorial.
3
```

Figure 4.93: Изменение оригинального репозитория

Сделаем коммит (Рис. 4.94).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git add README.md

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git commit -m "Changed README inn original repo"
[master 2af975e] Changed README inn original repo
 1 file changed, 1 insertion(+)
```

Figure 4.94: Коммит

Извлечем изменения из удаленного репозитория (Рис. 4.95).

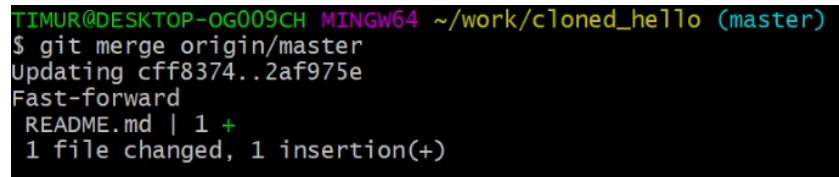


The screenshot shows a code editor with three tabs: index.html, # style.css, and hello.html. The index.html tab is active, displaying the following code:

```
C: > Users > TIMUR > work > hello > index.html > html > head > link  
1 | <html>  
2 | | <head>  
3 | | | <link type="text/css" rel="stylesheet" media="all" href="lib/style.  
4 | | | cs" />  
5 | | </head>  
6 | | <body>  
7 | | | <iframe src="lib/hello.html" width="200" height="200"></iframe>  
8 | </body>  
</html>
```

Figure 4.95: Извлечение изменений

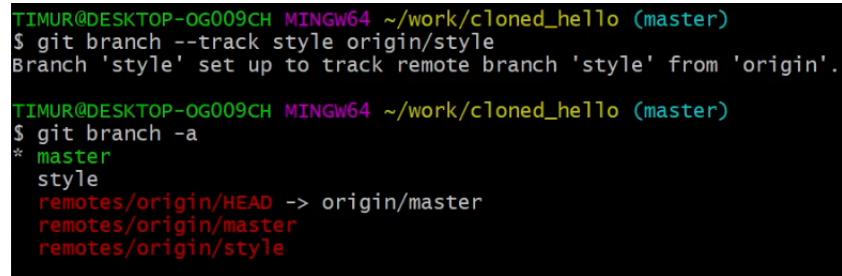
Теперь сольем извлеченные изменения (Рис. 4.96)



```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/cloned_hello (master)  
$ git merge origin/master  
Updating cff8374..2af975e  
Fast-forward  
 README.md | 1 +  
 1 file changed, 1 insertion(+)
```

Figure 4.96: Слияние изменений

Добавим ветку наблюдения (Рис. 4.97)



```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/cloned_hello (master)  
$ git branch --track style origin/style  
Branch 'style' set up to track remote branch 'style' from 'origin'.  
  
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/cloned_hello (master)  
$ git branch -a  
* master  
  style  
  remotes/origin/HEAD -> origin/master  
  remotes/origin/master  
  remotes/origin/style
```

Figure 4.97: Ветка наблюдения

Создадим чистый репозиторий (Рис. 4.98)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/cloned_hello (master)
$ cd ..

TIMUR@DESKTOP-OG009CH MINGW64 ~/work
$ git clone --bare hello hello.git
cloning into bare repository 'hello.git'...
done.

TIMUR@DESKTOP-OG009CH MINGW64 ~/work
$ ls hello.git
HEAD config description hooks/ info/ objects/ packed-refs refs/
```

Figure 4.98: Чистый репозиторий

Добавим удаленный репозиторий (Рис. 4.99)

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work
$ cd hello

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git remote add shared ../hello.git
```

Figure 4.99: Добавление удаленного репозитория

Изменим файл readme(Рис. 4.100). Закоммитим и отправим изменения в общий репозиторий (Рис. 4.101)

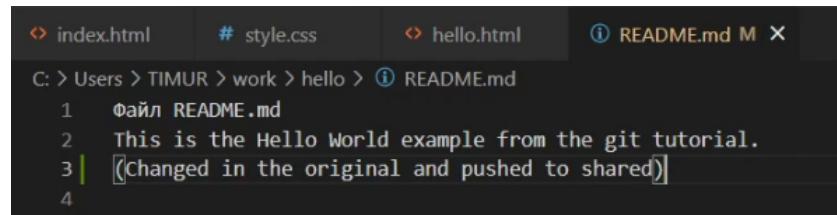


Figure 4.100: Изменение README

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git add README.md

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git commit -m "Added shared comment to readme"
[master fe8e3eb] Added shared comment to readme
 1 file changed, 1 insertion(+)

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/hello (master)
$ git push shared master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 432 bytes | 432.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To ../hello.git
  2af975e..fe8e3eb  master -> master
```

Figure 4.101: Отправка изменений

Вернемся в клонированный репозиторий и извлечем изменения (Рис. 4.102).

```
TIMUR@DESKTOP-OG009CH MINGW64 ~/work/cloned_hello (master)
$ git remote add shared ../hello.git

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/cloned_hello (master)
$ git branch --track shared master
Branch 'shared' set up to track local branch 'master'.

TIMUR@DESKTOP-OG009CH MINGW64 ~/work/cloned_hello (master)
$ git pull shared master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 412 bytes | 41.00 KiB/s, done.
From ../hello
 * branch           master      -> FETCH_HEAD
 * [new branch]     master      -> shared/master
Updating 2af975e..fe8e3eb
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)
```

Figure 4.102: Извлечение новых изменений

5 Выводы

В результате лабораторной работы мы познакомились с системой контроля версий git