



ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ  
ГОРОДА МОСКВЫ

Государственное бюджетное профессиональное  
образовательное учреждение города Москвы  
«Колледж малого бизнеса № 4»  
(ГБПОУ КМБ № 4)

## Отчет гит

Специальность: 09.02.07 Информационные системы и программирование

Форма обучения: очная

Студент(ка): Козлова София Евгеньевна

Группа: ИПО-21.24

Руководитель: Рыбаков Александр Сергеевич

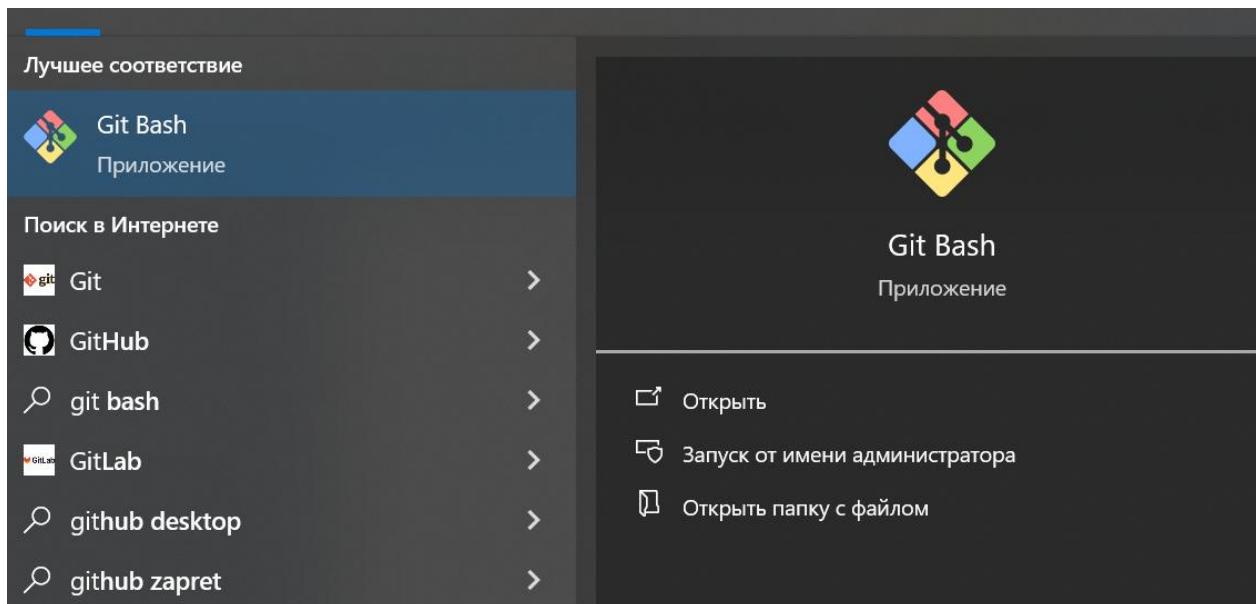
Отчётная работа защищена с оценкой «\_\_\_\_\_» \_\_\_\_\_

Москва, 2025 г.

## Оглавление

Лабораторная работа №2.....	3
Вопросы: .....	5
Лабораторная работа №3.....	6
Самостоятельная работа.....	10
Вопросы:.....	13
Лабораторная работа №4.....	14
Самостоятельная работа .....	15
Вопросы:.....	16
Лабораторная работа №5.....	17
Лабораторная работа №6.....	19
Вопросы:.....	21
Лабораторная работа №7.....	22
Самостоятельная работа .....	31
ССЫЛКА НА ГИТХАБ: <a href="https://github.com/kozlovalova/soniaaa.git">https://github.com/kozlovalova/soniaaa.git</a> Вопросы:.....	41

# Лабораторная работа №2



## Линукс команды

Ls для вывода файлов в папке (модификатор –a(вывод скрытых файлов)

Cd для перемещения

Pwd – вывод директории

## Регистрация

```
User@S_BMESH_05 MINGW64 ~
$ cd C:/D/tmp
```

```
User@S_BMESH_05 MINGW64 /c/D/tmp
$ git config --global user.name "lova"
User@S_BMESH_05 MINGW64 /c/D/tmp
$ git config --global user.email "kozlova114930@gmail.com"
```

Проверка репозитория (пустой или нет)

```
User@S_BMESH_05 MINGW64 /c/D/tmp
$ ls

User@S_BMESH_05 MINGW64 /c/D/tmp
$ git init
Initialized empty Git repository in C:/D/tmp/.git/

User@S_BMESH_05 MINGW64 /c/D/tmp (master)
$ ls

User@S_BMESH_05 MINGW64 /c/D/tmp (master)
$ ls -a
./ ../ .git/
```

```
User@S_BMESH_05 MINGW64 /c/D/tmp (master)
$ C:/D/GitRepo/tmp
bash: C:/D/GitRepo/tmp: Is a directory
```

```
User@S_BMESH_05 MINGW64 /c/D/tmp
$ cd C:/D/GitRepo/tmp

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp (master)
$ ls -a
./ ../ .git/

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp (master)
$ |
```

## Повседневные команды

Git bash – консольный интерфейс взаимодействующих с линукс ядром

Gti init – инициализация репозитория

Git status – просмотр текущего статуса репозитория

Global - для настройки общих параметров

Git add – передает файл в систему гит, в состоянии подготовленное (stages)

## Вопросы:

### 1. Что такое GitBash?

**GitBash** — это терминал (командная строка), который устанавливается вместе с Git в Windows и позволяет выполнять команды Git и Linux-команды.

---

### 2. Для чего нужна команда ls?

Команда `ls` выводит список файлов и папок в текущей директории.

---

### 3. Как сменить директорию?

Командой:

```
cd имя_папки
```

Например:

```
cd Documents
```

---

### 4. Как отобразить текущую директорию?

Команда:

```
pwd
```

Показывает полный путь к текущей папке.

---

### 5. Для чего задаются настройки Git?

Чтобы указать имя пользователя и email — они будут записываться в каждый коммит для идентификации автора.

---

### 6. Как применить одинаковые настройки для всех репозиториев?

Использовать флаг `--global`:

```
git config --global user.name "Имя"  
git config --global user.email "email"
```

---

# Лабораторная работа №3

Создаем файл README.md в папке tmp

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   README.md
    new file:   README.txt

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    README.txt
```

Добавление файла README в подготовленное состояние и проверяем статус репозитория

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp (master)
$ git commit -m "Добавление файла README"
[master (root-commit) 29b13fa] Добавление файла README
 2 files changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 README.md
  create mode 100644 README.txt
```

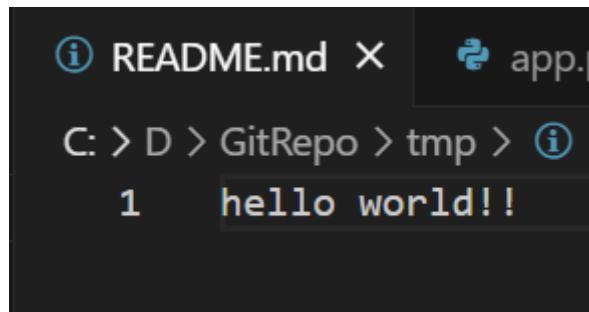
```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp (master)
$ git add .

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    README.txt
```

Git commit – для создание коммита (-m далее пишем текст нашего коммита)

Git commit – я попадаю в текстовый редактор vim

Открываем файл README.md и пишем “hello world!!”



```
① README.md X  ↗ app.p
C: > D > GitRepo > tmp > ①
1    hello world!!
```

Видим, что файл README.md перешел в статус “modified”

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp (master)
$ git add .

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   README.md
    deleted:    README.txt
```

```
User@S_BMESH_05 MINGW64 ~
$ git bash
git: 'bash' is not a git command. See 'git --help'.

The most similar command is
  stash

User@S_BMESH_05 MINGW64 ~
$ git clone https://github.com/kozlovalova/GitTest.git
Cloning into 'GitTest'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

User@S_BMESH_05 MINGW64 ~
$ |
```

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp (master)
$ cd GitTest

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/GitTest (main)
$ git add hi.txt

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/GitTest (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   hi.txt
```

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp (master)
$ cd GitTest

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/GitTest (main)
$ git add hi.txt

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/GitTest (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   hi.txt

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/GitTest (main)
$ git remote -v
origin  https://github.com/kozlovalova/GitTest.git (fetch)
origin  https://github.com/kozlovalova/GitTest.git (push)

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/GitTest (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   hi.txt
```

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/GitTest (main)
$ git commit -m "Hi.txt"
[main a77ee20] Hi.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 hi.txt
```

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/GitTest (main)
$ git push
info: please complete authentication in your browser...
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 270 bytes | 67.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/kozlovalova/GitTest.git
  d52b5c1..a77ee20  main -> main
```

```
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 943 bytes | 55.00 KiB/s, done.
From https://github.com/kozlovalova/GitTest
  a77ee20..f71459d  main      -> origin/main
Updating a77ee20..f71459d
Fast-forward
 README.md | 3 ++
 1 file changed, 2 insertions(+), 1 deletion(-)

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/GitTest (main)
$ git log
commit f71459da225f7c9e5d55b0aa4ac9b19a28853806 (HEAD -> main, origin/main, origin/HEAD)
Author: kozlovalova <kozlova114930@gmail.com>
Date:   Thu Jan 22 11:52:08 2026 +0300

    Update README.md

commit a77ee2043f5775fdb003a1f877a14f72b8943cae
Author: lova <kozlova114930@gmail.com>
Date:   Thu Jan 22 11:35:57 2026 +0300

    Hi.txt

commit d52b5c117745741e70d2f0fa8cc2ab5ac44c0c89
Author: kozlovalova <kozlova114930@gmail.com>
Date:   Thu Jan 22 11:16:06 2026 +0300

    Initial commit
```

Cat – для того , чтобы смотреть

git pull

git push

Git Remote -v – команда для отвязки от одного репозитория к другому

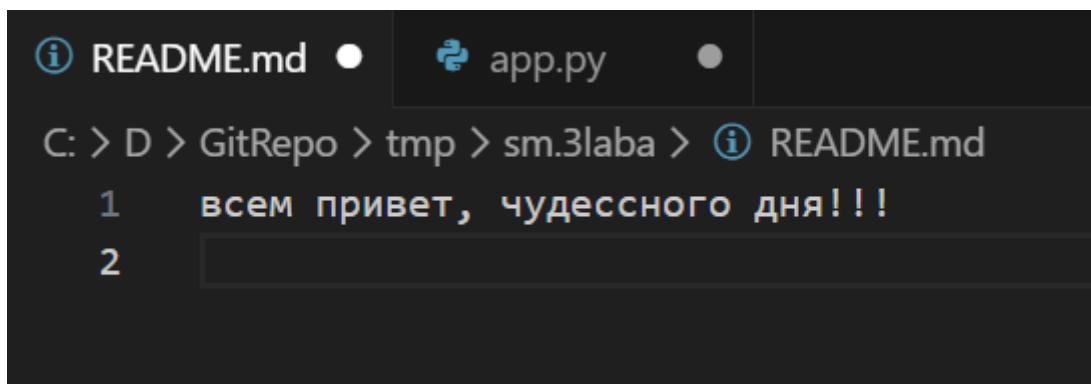
## Самостоятельная работа

Создаем папку для репозитория, переходим в эту папку через команду cd и инициализируем репозиторий.

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp (master)
$ cd C:/D/GitRepo/tmp/sm.3laba

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/sm.3laba (master)
$ git init
Initialized empty Git repository in C:/D/GitRepo/tmp/sm.3laba/.git/
```

Добавляем в папку файл README.md и пишем в нем «всем привет, чудесного дня!!!»



The screenshot shows a terminal window with two tabs: 'README.md' and 'app.py'. The current tab is 'README.md'. The file content is:

```
C: > D > GitRepo > tmp > sm.3laba > README.md
1      всем привет, чудесного дня!!!
2
```

Выполняем команду git status для просмотра текущего статуса проекта, видим что файл находится в неотслеживаемом состоянии, для того чтобы он изменил свое состояние до отслеживаемого изменяемого выполняем команду git add \*

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/sm.3laba (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    README.md

nothing added to commit but untracked files present (use "git add" to track)

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/sm.3laba (master)
$ git *
git: 'README.md' is not a git command. See 'git --help'.

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/sm.3laba (master)
$ git add *

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/sm.3laba (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   README.md
```

Сохраняем изменения через команду git commit -m "README.md"

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/sm.3laba (master)
$ git commit -m "README"
[master (root-commit) 4024cf2] README
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

После выполнения всех коммитов просматриваем список этих коммитов через команды git log

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/sm.3laba (master)
$ git add *

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/sm.3laba (master)
$ git commit -m "Добавлены 3 файла"
[master 86162e6] Добавлены 3 файла
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 1.txt
 create mode 100644 2.txt
 create mode 100644 3.txt
```

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/sm.3laba (master)
$ git log
commit 86162e6970b49553ebdeef961bb01ecb48d9103b (HEAD -> master)
Author: lova <kozlova114930@gmail.com>
Date:   Thu Jan 29 09:46:10 2026 +0300

    Добавлены 3 файла

commit 4024cf2c21469c33f49a355bb26855965159c3a1
Author: lova <kozlova114930@gmail.com>
Date:   Thu Jan 29 09:30:06 2026 +0300

    README
```

Имя	Дата изменения	Тип	Размер
.git	29.01.2026 9:50	Папка с файлами	
подпапка	29.01.2026 9:49	Папка с файлами	
1.txt	29.01.2026 9:38	Текстовый документ	0 КБ
2.txt	29.01.2026 9:38	Текстовый документ	0 КБ
3.txt	29.01.2026 9:38	Текстовый документ	0 КБ
README.md	29.01.2026 9:21	Исходный файл Mag...	1 КБ

Через коману mkdir создаем подпапку и добавляем в нее файлы

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/sm.3laba (master)
$ mkdir подпапка

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/sm.3laba (master)
$ git add подпапка
```

Через команду rm подпапка/\* удаляем все файлы в подпапке

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/sm.3laba (master)
$ git add подпапка

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/sm.3laba (master)
$ git commit -m "Добавление файлов в подпапку"
[master c7f4e77] добавление файлов в подпапку
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 "\320\277\320\276\320\264\320\277\320\260\320\277\320\272\320\260\aaa.txt"
```

Через команду rmdir подпапка удаляем подпапку

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/sm.3laba (master)
$ rm подпапка/*
User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/sm.3laba (master)
$ git add *

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/sm.3laba (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:   "\320\277\320\276\320\264\320\277\320\260\320\277\320\272\320\260\aaa.txt"

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/sm.3laba (master)
$ rmdir подпапка
User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/sm.3laba (master)
$ git add *
```

Основные заполнители (часто используемые)

%H — полный хэш коммита;

%h — сокращённый хэш (7 символов);

%an — имя автора;

%ae — email автора;

%cn — имя коммиттера (того, кто закоммитил);

%ce — email коммиттера;

%ad — дата автора (в формате repo);

%cd — дата коммиттера (в формате repo);

%ar — относительная дата (например, «2 дня назад»);

%s — сообщение коммита (subject);

%b — тело сообщения коммита (body);

%d — референсы (ветки, теги) для коммита;

%n — перевод строки;

%C(цвет) — смена цвета вывода (см. ниже).

## **Вопросы:**

1. Как отобразить текущий статус репозитория?

```
git status
```

---

2. Какие файлы называются Untracked files?

Это файлы, которые Git видит, но они **не добавлены под контроль версий**.

---

3. Как передать файл под контроль Git?

```
git add имя_файла
```

Или все файлы:

```
git add .
```

---

4. Как отобразить список изменений в репозитории?

```
git diff
```

---

5. Как отменить коммит и почему не рекомендуется?

Отмена:

```
git reset --hard HEAD~1
```

Не рекомендуется, потому что удаляется история изменений, и можно потерять данные.

---

6. Как Git идентифицирует этапы изменений?

Каждый коммит получает **уникальный хеш**

## Лабораторная работа №4

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/GitTest (main)
$ git clone https://github.com/kozlovalova/GitTest.git
Cloning into 'GitTest'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 9 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (9/9), done.

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/GitTest (main)
$ cd C:/D/GitRepo/tmp

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp (master)
$ cd GitTest

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/GitTest (main)
$ git add hi.txt

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/GitTest (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    GitTest/

nothing added to commit but untracked files present (use "git add" to track)
```

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/GitTest (main)
$ git remote -v
origin  https://github.com/kozlovalova/GitTest.git (fetch)
origin  https://github.com/kozlovalova/GitTest.git (push)

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/GitTest (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    GitTest/

nothing added to commit but untracked files present (use "git add" to track)

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp/GitTest (main)
$
```

Cat – для того , чтобы смотреть

git pull – для выкачивания данных с гитхаба

git push – для скачивание на гит хаб

Git Remote -v – команда для отвязки от одного репозитория к другому

## Самостоятельная работа

Создаем репозиторий на компьютере и инициализируем его

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo
$ git init
Initialized empty Git repository in C:/D/GitRepo/.git/
User@S_BMESH_05 MINGW64 /c/D/GitRepo (master)
$ git remote add origin https://github.com/kozlovalova/GitTest.git

User@S_BMESH_05 MINGW64 /c/D/GitRepo (master)
$ git remote -v
origin  https://github.com/kozlovalova/GitTest.git (fetch)
origin  https://github.com/kozlovalova/GitTest.git (push)
```

Связываем его с первым репозиторием и проверяем и Скачиваем данные из гитхаба

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo (master)
$ git fetch origin
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 9 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (9/9), 1.98 KiB | 25.00 KiB/s, done.
From https://github.com/kozlovalova/GitTest
 * [new branch]      main      -> origin/main
```

## **Вопросы:**

1. Как отправить изменения на удаленный репозиторий?

```
git push origin main
```

---

2. Как влить изменения из удаленного репозитория?

```
git pull
```

---

3. Как зафиксировать изменения?

```
git commit -m "Комментарий"
```

---

4. Как используется GitHub в реальной работе?

- Хранение кода
  - Совместная работа
  - Отслеживание изменений
  - Резервное копирование
- 

5. Принцип коллективной работы с GitHub

1. Создание ветки
2. Внесение изменений
3. Commit
4. Push
5. Pull Request
6. Проверка и слияние

# Лабораторная работа №5

Работа с удаленными репозиториями

```
User@S_BMESH_05 MINGW64 /c/d/Repo2 (master)
$ git add .

User@S_BMESH_05 MINGW64 /c/d/Repo2 (master)
$ git commit -m "сонакласссполныйгаз"
[master (root-commit) 8fedf94] сонакласссполныйгаз
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md

User@S_BMESH_05 MINGW64 /c/d/Repo2 (master)
```

## Работа с удаленными репозиториями

Git remote – выводит связи

Git fetch – переносит все изменения из подключаемого репозитория

Git checkout – перемещение по коммитам

Git remote add название паки и путь папки – для связи репозитория

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp2 (master)
$ git remote -v
hihik    c:/D/Repo2 (fetch)
hihik    C:/D/Repo2 (push)

User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp2 (master)
$ git fetch hihik
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 223 bytes | 3.00 KiB/s, done.
From C:/D/Repo2
 * [new branch]      master      -> hihik/master
```

y

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp2 (master)
$ git checkout --track hihik/master
branch 'master' set up to track 'hihik/master'.
Already on 'master'
```

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/tmp2 (master)
$ git pull
Already up to date.
```

Вопросы:

1. Что такое ветка?

Ветка — это отдельная линия разработки проекта.

---

2. Типы репозиториев Git:

- Локальный
  - Удалённый
- 

3. Как добавить ссылку на удаленный репозиторий?

```
git remote add origin ссылка
```

---

4. Как вывести список подключённых репозиториев?

```
git remote -v
```

---

5. Что делает git remote show?

Показывает подробную информацию об удаленном репозитории.

# Лабораторная работа №6

## Работа с метками

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/branches (master)
$ git add .

User@S_BMESH_05 MINGW64 /c/D/GitRepo/branches (master)
$ git commit -m
error: switch `m' requires a value

User@S_BMESH_05 MINGW64 /c/D/GitRepo/branches (master)
$ git commit -m "гудкоб"
[master (root-commit) c4f4cdd] гудкоб
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 "\320\276\321\203\321\203.txt"

User@S_BMESH_05 MINGW64 /c/D/GitRepo/branches (master)
$ git tag 1.0

User@S_BMESH_05 MINGW64 /c/D/GitRepo/branches (master)
$ git tag
1.0

User@S_BMESH_05 MINGW64 /c/D/GitRepo/branches (master)
$ git show 1.0
commit c4f4cdd5a1bf5a97b8736a5511de7977d99d728e (HEAD -> master, tag: 1.0)
Author: lova <kozloval14930@gmail.com>
Date:   Thu Jan 29 11:47:23 2026 +0300

    гудкоб

diff --git "a/\320\276\321\203\321\203.txt" "b/\320\276\321\203\321\203.txt"
new file mode 100644
index 0000000..e69de29

User@S_BMESH_05 MINGW64 /c/D/GitRepo/branches (master)
$
```

```
User@S_BMESH_05 MINGW64 ~
$ cd C:/D/GitRepo/branches

User@S_BMESH_05 MINGW64 /c/D/GitRepo/branches (master)
$ git log --oneline
c4f4cdd (HEAD -> master, tag: 1.0) гудкоб
```

```
User@S_BMESH_05 MINGW64 /c/d/Repo2 (master)
$ cd C:/D/GitRepo/branches

User@S_BMESH_05 MINGW64 /c/D/GitRepo/branches
$ git init
Initialized empty Git repository in C:/D/GitRepo/branches/.git/

User@S_BMESH_05 MINGW64 /c/D/GitRepo/branches (master)
```

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/branches (master)
$ git tag 1.1 c4f4cdd
```

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/branches (master)
$ git submodule add https://github.com/kozlovalova/lalala.git
Cloning into 'C:/D/GitRepo/branches/lalala'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
warning: in the working copy of '.gitmodules', LF will be replaced by CRLF the next time Git touches it
```

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/branches (master)
$ ls
lalala/  oyy.txt

User@S_BMESH_05 MINGW64 /c/D/GitRepo/branches (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .gitmodules
    new file:   lalala
```

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/branches (master)
$ cat .gitmodules
[submodule "lalala"]
  path = lalala
  url = https://github.com/kozlovalova/lalala.git

User@S_BMESH_05 MINGW64 /c/D/GitRepo/branches (master)
$ git add .
```

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/branches (master)
$ git commit -m "new submodule"
[master 7eed436] new submodule
 2 files changed, 4 insertions(+)
 create mode 100644 .gitmodules
 create mode 160000 lalala
```

## **Вопросы:**

### 1. Что такое метка (tag)?

Метка — это указатель на определённый коммит (обычно для версии проекта).

---

### 2. Типы меток:

- Легковесные
  - Аннотированные
- 

### 3. Как добавить метку?

```
git tag v1.0
```

---

### 4. Как вывести информацию о метке?

```
git show v1.0
```

---

### 5. Для чего нужен сабмодуль?

Чтобы подключить один Git-репозиторий внутрь другого.

---

### 6. Как добавить сабмодуль?

```
git submodule add ссылка
```

## Лабораторная работа №7

Инициализируйте репозиторий в этой папке

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches
$ git init
Initialized empty Git repository in C:/D/GitRepo/ProGit/brunches/.git/

User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

Вызовите git log, и вы увидите, что существует только один «Первый коммит».

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (master)
$ git log
fatal: your current branch 'master' does not have any commits yet

User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (master)
$ git add .

User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (master)
$ git commit -m "PPP"
[master (root-commit) d656fb8] PPP
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 readme.md
```

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (master)
$ git log
commit d656fb896b82b844e4f800ca51c135295d98934e (HEAD -> master)
Author: lova <kozloval14930@gmail.com>
Date:   Thu Feb 5 11:05:37 2026 +0300

    PPP
```

Однако, мы до сих пор находимся в ветке master!!!Проверьте это командой git status.Это потому, что команда git branch ветку только создает.

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (master)
$ git status
On branch master
nothing to commit, working tree clean

User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (master)
$ git branch
* master
```

Теперь создаем новую ветку.

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (master)
$ git branch testing
```

Однако, мы до сих пор находимся в ветке master!!!Проверьте это командой git status.Это потому, что команда git branch ветку только создает.

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (master)
$ git status
On branch master
nothing to commit, working tree clean

User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (master)
$ git checkout testing
Switched to branch 'testing'
```

Просмотр

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (testing)
$ git branch
  master
* testing
```

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (master)
$ git status
On branch master
nothing to commit, working tree clean

User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (master)
$ git checkout testing
Switched to branch 'testing'
```

Перейдите на ветку testing.

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (testing)
$ git checkout master
Switched to branch 'master'

User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (master)
$ git add .

User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (master)
$ git commit -m "суперпупер"
[master bb18398] суперпупер
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 index.txt
```

Также можно проверить текущую ветку с помощью gitstatus:

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (master)
$ git status
On branch master
nothing to commit, working tree clean

User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (master)
$ git log
commit bb18398892e871f289a2cc4cf40ee4f8d50f4f60 (HEAD -> master)
Author: lova <kozlova114930@gmail.com>
Date:   Thu Feb 5 11:17:26 2026 +0300

    суперпупер

commit d656fb896b82b844e4f800ca51c135295d98934e (testing)
Author: lova <kozlova114930@gmail.com>
Date:   Thu Feb 5 11:05:37 2026 +0300

    PPP
```

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (master)
$ git checkout testing
Switched to branch 'testing'

User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (testing)
$ git log
commit d656fb896b82b844e4f800ca51c135295d98934e (HEAD -> testing)
Author: lova <kozloval14930@gmail.com>
Date:   Thu Feb 5 11:05:37 2026 +0300

    PPP
```

Проверьте лог, вы увидите все три коммита. При этом третий коммит относится только к ветке hotfix:

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (master)
$ git branch hotfix

User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (master)
$ git merge hotfix
Already up to date.

User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (master)
$ git checkout hotfix
Switched to branch 'hotfix'

User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (hotfix)
$ git log
commit bb18398892e871f289a2cc4cf40ee4f8d50f4f60 (HEAD -> hotfix, master)
Author: lova <kozloval14930@gmail.com>
Date:   Thu Feb 5 11:17:26 2026 +0300

    суперпупер

commit d656fb896b82b844e4f800ca51c135295d98934e (testing)
Author: lova <kozloval14930@gmail.com>
Date:   Thu Feb 5 11:05:37 2026 +0300

    PPP
```

Переключитесь на ветку master и выведите лог. Вы увидите два коммита ветки master.

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (hotfix)
$ git status
On branch hotfix
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:   index.txt
```



index.txt – Блокнот

Файл Правка Формат Вид Справка

## главный файл

Создали в программе красную кнопку

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/branches (hotfix)
$ git add .

User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/branches (hotfix)
$ git commit -m "Добавили красную кнопку"
[hotfix 338bc17] Добавили красную кнопку
 1 file changed, 3 insertions(+)
```

Переходим в ветку мастер

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/branches (hotfix)
$ git checkout master
Switched to branch 'master'
```

Смотрим список коммитов

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/branches (master)
$ git log
commit bb18398892e871f289a2cc4cf40ee4f8d50f4f60 (HEAD -> master)
Author: lova <kozlova114930@gmail.com>
Date:   Thu Feb 5 11:17:26 2026 +0300

    суперпупер

commit d656fb896b82b844e4f800ca51c135295d98934e (testing)
Author: lova <kozlova114930@gmail.com>
Date:   Thu Feb 5 11:05:37 2026 +0300

    PPP
```

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (master)
$ git merge hotfix
Updating bb18398..338bc17
Fast-forward
 index.txt | 3 +++
 1 file changed, 3 insertions(+)
```

Смотрим список коммитов

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (master)
$ git log
commit 338bc178c048e1804ee23eb29eb3211edebfe713 (HEAD -> master, hotfix)
Author: lova <kozlova114930@gmail.com>
Date:   Thu Feb 5 11:38:38 2026 +0300

    добавили красную кнопку

commit bb18398892e871f289a2cc4cf40ee4f8d50f4f60
Author: lova <kozlova114930@gmail.com>
Date:   Thu Feb 5 11:17:26 2026 +0300

    суперпупер

commit d656fb896b82b844e4f800ca51c135295d98934e (testing)
Author: lova <kozlova114930@gmail.com>
Date:   Thu Feb 5 11:05:37 2026 +0300

    PPP
```

Вызовите команду git branch -v

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (master)
$ git branch -v
  hotfix  338bc17 Добавили красную кнопку
* master  338bc17 Добавили красную кнопку
  testing d656fb8 PPP
```

Этак команда выводит полный перечень веток репозитория.

Звездочкой отмечена текущая ветка, на которую смотрит HEAD.

Проверьте статус, вы должны находиться в ветке master:

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/brunches (master)
$ git checkout master
Already on 'master'
```

Она уже ушла достаточно далеко от ветки testing, созданной ранее. Убедимся в этом, выведем лог для ветки master, затем перейдем в ветку testing и проверим лог для неё.

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/branches (master)
$ git log
commit 338bc178c048e1804ee23eb29eb3211edebfe713 (HEAD -> master, hotfix)
Author: lova <kozlova114930@gmail.com>
Date:   Thu Feb 5 11:38:38 2026 +0300

    добавили красную кнопку

commit bb18398892e871f289a2cc4cf40ee4f8d50f4f60
Author: lova <kozlova114930@gmail.com>
Date:   Thu Feb 5 11:17:26 2026 +0300

    суперпупер

commit d656fb896b82b844e4f800ca51c135295d98934e (testing)
Author: lova <kozlova114930@gmail.com>
Date:   Thu Feb 5 11:05:37 2026 +0300

    PPP
```

Переходим в testing:

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/branches (master)
$ git checkout testing
Switched to branch 'testing'
```

Снова переключаемся в master и вливаем неё ветку testing (git merge testing). Начнется слияние

Коммитим

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/branches (master)
$ git add .

User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/branches (master)
$ git commit -m "создали файл"
[master 64b5d09] создали файл
 1 file changed, 1 insertion(+)
```

Создаем ветку и переходим в нее

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/branches (master)
$ git checkout -b feature/1
Switched to a new branch 'feature/1'
```

Мы поменяли текстовый файл добавили новую надпись и закоммитили изменения

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/branches (feature/1)
$ git add .

User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/branches (feature/1)
$ git commit -m "Добавили красную кнопку"
[feature/1 1d6a5de] добавили красную кнопку
 1 file changed, 1 insertion(+), 1 deletion(-)
```

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/branches (feature/2)
$ git checkout feature/1
Switched to branch 'feature/1'
```

Перешли обратно в ветку мастер и создали вторую ветку и перешли в нее

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/branches (feature/1)
$ git merge feature/2
Auto-merging first.txt
CONFLICT (content): Merge conflict in first.txt
Automatic merge failed; fix conflicts and then commit the result.
```

Переходим в первую ветку где красная кнопка объединяем ее со второй где зеленые

```
first.txt – Блокнот
Файл Правка Формат Вид Справка
<<<<< HEAD
Добавили красную кнопку
=====
все еще хочу домой
добавль две зеленых кнопки

>>>>> feature/2

User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/branches (feature/1|MERGING)
$ git status
On branch feature/1
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:  first.txt

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

Добавили надпись две зеленых кнопки и закоммитили изменения

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/branches (feature/1|MERGING)
$ git add .

User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/branches (feature/1|MERGING)
$ git commit -m "конфлмкт решен"
[feature/1 07767fc] конфлмкт решен
```

## Самостоятельная работа

Создаем новый пустой репозиторий на гитхабе

soniaaa Public

Updated Feb 9, 2026

Клонируем его в папку D вместо диска D

```
User@S_BMESH_05 MINGW64 /c/D/GitRepo/ProGit/branchesies (feature/1)
$ git clone https://github.com/kozlovalova/soniaaa.git
Cloning into 'soniaaa'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
```

Создаем файл HTML с моим профилем

Restricted Mode is intended for safe code browsing. Trust this folder to enable all features. [Manage](#) [Learn More](#)

```
↳ index.html X
↳ index.html > ↳ html > ↳ body > ↳ h1
1  <!DOCTYPE html>
2  <html lang="ru">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <title>Мой профиль</title>
6  |   <link rel="stylesheet" href="style.css">
7  </head>
8  <body>
9  |   <h1>Алим Абрамов</h1>
10 |   <p>Веб-разработчик. Специализируюсь на HTML/CSS и JavaScript.</p>
11 |   <p>Люблю создавать адаптивные интерфейсы и оптимизировать производительность.</p>
12 </body>
13 </html>
```

Переходим в папку репозитория

Добавляем добавленный файл по контроль гита

И коммитим изменения

```
User@S_BMESH_05 MINGW64 ~
$ cd C:/D/soniaaa

User@S_BMESH_05 MINGW64 /c/D/soniaaa (main)
$ git add .

User@S_BMESH_05 MINGW64 /c/D/soniaaa (main)
$ git commit -m "Добавила index.html с базовой информацией о чувачке"
[main d41a990] Добавила index.html с базовой информацией о чувачке
 1 file changed, 13 insertions(+)
 create mode 100644 index.html
```

Пушим в гитхаб в ветку майн

```
create mode 100644 index.html

User@S_BMESH_05 MINGW64 /c/D/soniaaa (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 680 bytes | 340.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/kozlovalova/soniaaa.git
  2459a50..d41a990  main -> main
```

добавляем ветку и сразу же переходим в нее

```
User@S_BMESH_05 MINGW64 /c/D/soniaaa (main)
$ git checkout -b add-subscribe-button
Switched to a new branch 'add-subscribe-button'
```

Добавляем изменения под контроль гит и коммитим

```
User@S_BMESH_05 MINGW64 /c/D/soniaaa (add-subscribe-button)
$ git commit -m "Добавила кнопку 'Подписаться на новости'"
[add-subscribe-button 0e51326] Добавила кнопку 'Подписаться на новости'
 1 file changed, 2 insertions(+)
```

Пушим на гит хаб через ветку созданную для кнопки

```
User@S_BMESH_05 MINGW64 /c/D/soniaaa (add-subscribe-button)
$ git push origin add-subscribe-button
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 467 bytes | 467.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'add-subscribe-button' on GitHub by visiting:
remote:     https://github.com/kozlovalova/soniaaa/pull/new/add-subscribe-button
remote:
remote: To https://github.com/kozlovalova/soniaaa.git
 * [new branch]      add-subscribe-button -> add-subscribe-button
```

Переходим в ветку main

```
User@S_BMESH_05 MINGW64 /c/D/soniaaa (add-subscribe-button)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

Выполняем слияние веток

```
User@S_BMESH_05 MINGW64 /c/D/soniaaa (main)
$ git merge add-subscribe-button
Updating d41a990..0e51326
Fast-forward
 index.html | 2 ++
 1 file changed, 2 insertions(+)
```

И пушим на гитхаб

```
User@S_BMESH_05 MINGW64 /c/D/soniaaa (main)
$ git push origin main
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/kozlovalova/soniaaa.git
  d41a990..0e51326  main -> main
```

Создаем ветку для изменения цвета кнопки

```
User@S_BMESH_05 MINGW64 /c/D/soniaaa (main)
$ git checkout -b change-button-color
Switched to a new branch 'change-button-color'
```

Изменяем цвет в файле html

```
<button
background-color: #5dss60
color whire;
border: none;
paddind: 10px 20px;>Подписаться на новости </button>
body>
```

Добавляем изменения под контроль гит и коммитим

```
User@S_BMESH_05 MINGW64 /c/D/soniaaa (change-button-color)
$ git commit -m "Измененён цвет кнопки на зеленых"
[change-button-color c4f309d] Измененён цвет кнопки на зеленых
 1 file changed, 5 insertions(+), 1 deletion(-)
```

Пушим на гитхаб

```
User@S_BMESH_05 MINGW64 /c/D/soniaaa (change-button-color)
$ git push origin change-button-color
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 431 bytes | 215.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'change-button-color' on GitHub by visiting:
remote:     https://github.com/kozlovalova/soniaaa/pull/new/change-button-color
remote:
To https://github.com/kozlovalova/soniaaa.git
 * [new branch]      change-button-color -> change-button-color
```

Переходим в ветку мэйн

Сливаем ветки

И пушим на гитхаб

```
User@S_BMESH_05 MINGW64 /c/D/soniaaa (change-button-color)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

User@S_BMESH_05 MINGW64 /c/D/soniaaa (main)
$ git merge change-button-color
Updating 0e51326..c4f309d
Fast-forward
 index.html | 6 +++++-
 1 file changed, 5 insertions(+), 1 deletion(-)
```

Добавляем ветку

```
User@S_BMESH_05 MINGW64 /c/D/soniaaa (main)
$ git push origin main
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/kozlovalova/soniaaa.git
  0e51326..c4f309d  main -> main
```

Добавляем форму входа через соцсети

```
User@S_BMESH_05 MINGW64 /c/D/soniaaa (main)
$ git checkout -b add-social-login
Switched to a new branch 'add-social-login'
```

Добавляем форму входа через соцсети

```
<div class="social-login">
  <button>Войти через Google</button>
  <button>Войти через VK</button>
</div>
```

Добавляем изменения под контроль гит и коммитим

```
User@S_BMESH_05 MINGW64 /c/D/soniaaa (add-social-login)
$ git add .

User@S_BMESH_05 MINGW64 /c/D/soniaaa (add-social-login)
$ git commit -m "Добавлена форма входа через соцсети"
[add-social-login 450d510] Добавлена форма входа через соцсети
 1 file changed, 3 insertions(+)
```

Пушим на гитхаб

```
User@S_BMESH_05 MINGW64 /c/D/soniaaa (add-social-login)
$ git push origin add-social-login
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 446 bytes | 446.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'add-social-login' on GitHub by visiting:
remote:     https://github.com/kozlovalova/soniaaa/pull/new/add-social-login
remote:
To https://github.com/kozlovalova/soniaaa.git
 * [new branch]      add-social-login -> add-social-login
```

Переходим в ветку main

Сливаем ветки

Пушим

```
User@S_BMESH_05 MINGW64 /c/D/soniaaaa (add-social-login)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

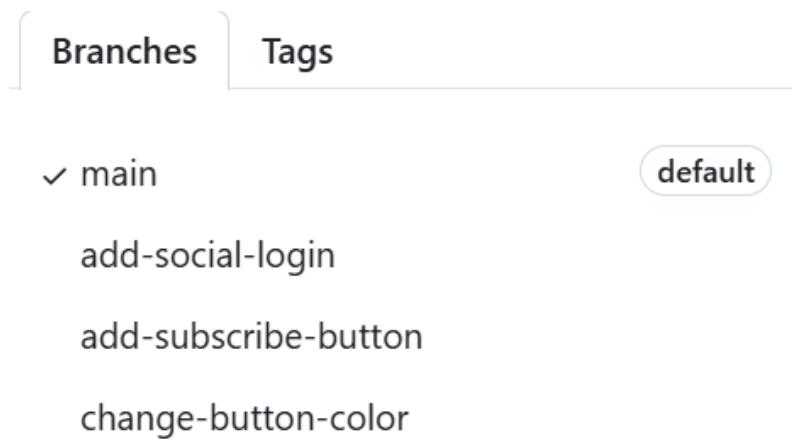
User@S_BMESH_05 MINGW64 /c/D/soniaaaa (main)
$ git merge add-social-login
Updating c4f309d..450d510
Fast-forward
 index.html | 3 +++
 1 file changed, 3 insertions(+)

User@S_BMESH_05 MINGW64 /c/D/soniaaaa (main)
$ git push origin main
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/kozlovalova/soniaaaa.git
  c4f309d..450d510  main -> main
```

Проверяем гитхаб Видим что было 4 коммита

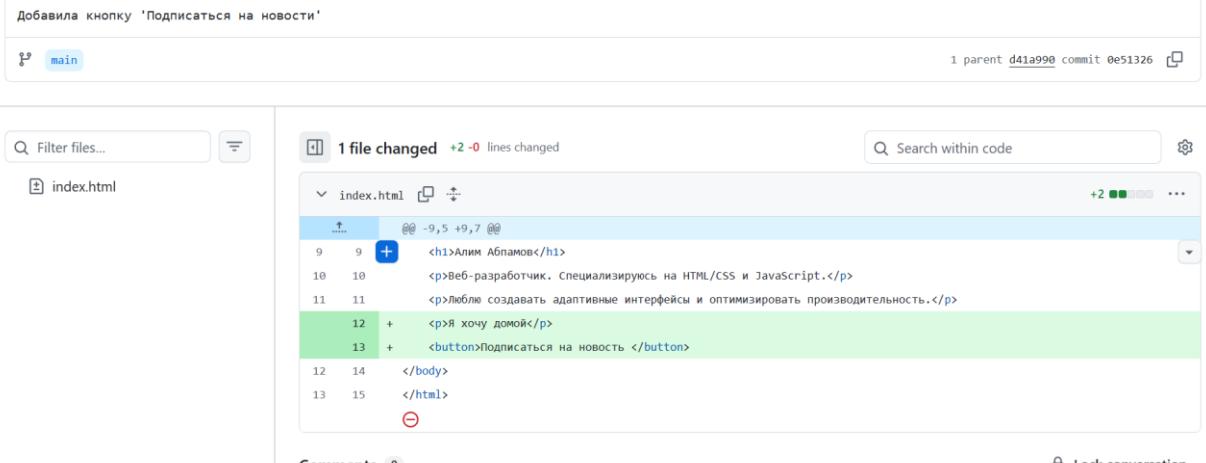
Добавлена форма входа через соцсети	450d510
Измененён цвет кнопки на зеленых	c4f309d
Добавила кнопку 'Подписаться на новости'	0e51326
Добавила index.html с базовой информацией о чувачке	d41a990

Видим что были добавлены ветки



Вывела второй коммит из гит хаба

Видим что произошло слияние и что есть родительский коммит и итоговый номер коммита, также видно что поменялось



Добавила кнопку 'Подписаться на новости'.

main

1 parent d41a990 commit 0e51326

index.html

1 file changed +2 -0 lines changed

index.html

Comments 0 Lock conversation

Search within code

index.html

12 + <button>Подписаться на новости </button>

13 + <button>Подписаться на новости </button>

14 </body>

15 </html>

Comments 0 Lock conversation

Search within code

Создаем ветку теста для создания конфликта

```
User@s_BMESH_05 MINGW64 /c/D/soniaaa (main)
$ git checkout -b conflict-test
Switched to a new branch 'conflict-test'
```

Изменяем имя кнопки на «Подписаться сейчас!»

```
<;>Подписаться сейчас! </button>
```

Добавляем под контроль гит пушим

```
User@s_BMESH_05 MINGW64 /c/D/soniaaa (conflict-test)
$ git commit -m "Изменено название кнопки в conflict-test"
[conflict-test e2859f2] Изменено название кнопки в conflict-test
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Переходим в ветку main

```
User@S_BMESH_05 MINGW64 /c/D/soniaaa (conflict-test)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

Добавляем изменения под контроль гит и коммитим

Слияем с веткой теста конфликта и видим что возник конфликт

```
User@S_BMESH_05 MINGW64 /c/D/soniaaa (main)
$ git add .

User@S_BMESH_05 MINGW64 /c/D/soniaaa (main)
$ git commit -m "Изменено название кнопки в main"
[main 0f856d9] Изменено название кнопки в main
 1 file changed, 1 insertion(+), 1 deletion(-)

User@S_BMESH_05 MINGW64 /c/D/soniaaa (main)
$ git merge conflict-test
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.

User@S_BMESH_05 MINGW64 /c/D/soniaaa (main) [MERGING]
```

Заходим в файл индекса и видим то что сконфликовало

## Алим Абпамов

Веб-разработчик. Специализируюсь на HTML/CSS и JavaScript.

Люблю создавать адаптивные интерфейсы и оптимизировать производительность.

Я хочу домой

[Подписаться](#) ===== paddind: 10px 20px;>Подписаться сейчас! >>>>> conflict-test  
[Войти через Google](#) [Войти через VK](#)

```
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
<<<<< HEAD (Current Change)
    paddind: 10px 20px;>Подписаться </button>
=====
    paddind: 10px 20px;>Подписаться сейчас! </button>
>>>>> conflict-test (Incoming Change)
    <div class="social-login">
        <button>Войти через Google</button>
        <button>Войти через VK</button>
```

Выбираем какое в итоге оставляем изменение

И добавляем это под контроль гит и коммитим

```
User@S_BMESH_05 MINGW64 /c/D/soniaaaa (main|MERGING)
$ git add .
User@S_BMESH_05 MINGW64 /c/D/soniaaaa (main|MERGING)
$ git commit -m "Зфзрешен конфликт в index.html"
[main 101e755] Зфзрешен конфликт в index.html
```

Пушим на гитхаб

```
User@S_BMESH_05 MINGW64 /c/D/soniaaaa (main)
$ git push origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 4 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 986 bytes | 197.00 KiB/s, done.
Total 9 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To https://github.com/kozlovalova/soniaaaa.git
  450d510..101e755  main -> main
```

Заходим на гит хаб и видим в коммитах все сохраненные коммиты

Commits on Feb 9, 2026

Зфзрешен конфликт в index.html  
kozlovalova committed 2 minutes ago

Изменено название кнопки в main  
kozlovalova committed 6 minutes ago

Изменено название кнопки в conflict-test  
kozlovalova committed 8 minutes ago

Так же видим коммит с разрешением конфликта

1 file changed +3 -3 lines changed

index.html

Comments 0 Lock conversation

**ССЫЛКА НА ГИТХАБ: <https://github.com/kozlovalova/soniaaa.git>**

## Вопросы:

1. Какая ветка создаётся первой?

Обычно **main** (или **master** в старых версиях).

---

2. Как создать новую ветку?

`git branch имя_ветки`

---

3. Что такое HEAD и связь с git checkout?

**HEAD** — это указатель на текущую ветку или коммит.  
`git checkout` переключает HEAD на другую ветку.

---

4. Как произвести слияние веток?

`git merge имя_ветки`

---

5. Что значит fast-forward?

Это слияние без создания нового коммита — когда ветка просто «перематывается вперёд».

