



**ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ
ГОРОДА МОСКВЫ**

**Государственное бюджетное профессиональное
образовательное учреждение города Москвы**

«Колледж малого бизнеса № 4»

(ГБПОУ КМБ № 4)

Лабораторная работа Git

**Специальность: 09.02.07 Информационные системы и
программирование**

Форма обучения: очная

Студент(ка): Пигорева Арианна Сергеевна

Группа: ИПО-21.24

Руководитель: Александр Сергеевич Рыбаков

Отчётная работа защищена с оценкой «___» _____

Москва, 2025 г

Оглавление

Лабораторная работа №2	3
Ответы на вопросы:	4
Лабораторная работа №3	5
Самостоятельная работа.	9
Ответы на вопросы:	12
Лабораторная работа №4	13
Самостоятельна работа.	15
Ответы на вопросы:	16
Лабораторная работа №5	17
Ответы на вопрос:.....	19
Лабораторная работа №6	21
Ответы на вопросы:	22
Лабораторная работа №7	23
Самостоятельная работа.	23
Ответы на вопросы:	29

Лабораторная работа №2

```
user@ARIANA MINGW64 ~  
$ cd C:/D/tmp
```

1. Создание папки

```
user@ARIANA MINGW64 /c/D/tmp  
$ git config --global user.name = "Arianna"  
  
user@ARIANA MINGW64 /c/D/tmp  
$ git config --global user.email = "ariannapigoreva@gmail.com"
```

2. Задаем настройки Git. Они используются для того, чтобы отслеживать авторов изменений.

```
user@ARIANA MINGW64 /c/D/tmp  
$ ls
```

3. Проверяем с помощью команды, скрытые файлы.

```
user@ARIANA MINGW64 /c/D/tmp  
$ git init  
Initialized empty Git repository in C:/D/tmp/.git/
```

4. Команда инициализирует репозиторий в текущей пустой папке, о чем выведется сообщение

```
user@ARIANA MINGW64 /c/D/tmp (master)
$ ls

user@ARIANA MINGW64 /c/D/tmp (master)
$ ls -a
./ ../ .git/

user@ARIANA MINGW64 /c/D/tmp (master)
$ |
```

5. Выполняем команду ls. Папка по-прежнему пуста.
6. Теперь выведем ту же команду с ключом -a:

```
user@ARIANA MINGW64 /c/D/tmp (master)
$ cd C:/D/GitRepo/tmp
bash: $'\302\203\302\223\302\215\302\223cd': command not found

user@ARIANA MINGW64 /c/D/tmp (master)
$ git status
On branch master

No commits yet

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

user@ARIANA MINGW64 /c/D/tmp (master)
$
```

7. Создать пустую папку GitRepo и перенести туда данную папку tmp .
Далее в GitBash перейдите в эту папку (команда cd).

Ответы на вопросы:

1. Что такое Git Bash?

Git Bash — это программа (терминал), которая позволяет работать с Git с помощью командной строки. Она предоставляет среду, похожую на Linux/Unix, в Windows и используется для выполнения Git-команд и обычных команд терминала.

2. Для чего нужна команда ls?

Команда ls используется для отображения списка файлов и папок в текущей директории.

3. Как сменить директорию?

Для смены директории используется команда:

cd имя_папки

Например:

cd Documents

Чтобы перейти на уровень выше:

```
cd ..
```

4. Как отобразить текущую директорию?

Чтобы узнать, в какой директории вы сейчас находитесь, используется команда:

```
pwd
```

5. Для чего задаются настройки Git?

Настройки Git задаются для указания информации о пользователе (имя и email), которая будет сохраняться в каждом коммите. Это нужно для идентификации автора изменений.



6. Как применить одинаковые настройки для всех репозиториев?

Нужно задать глобальные настройки с помощью флага `--global`:

```
git config --global user.name "Ваше Имя"  
git config --global user.email "your@email.com"
```

Такие настройки будут применяться ко всем репозиториям на компьютере.

Лабораторная работа №3

 .git	15.01.2026 11:43	Папка с файлами	
 README.md	15.01.2026 11:29	Исходный файл ...	0 КБ

1. Создайте в папке tmp файл README.md.

```
user@ARIANA MINGW64 /c/D/tmp (master)  
$ git add README.md  
  
user@ARIANA MINGW64 /c/D/tmp (master)  
$
```

```

user@ARIANA MINGW64 /c/D/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

user@ARIANA MINGW64 /c/D/tmp (master)
$

```

2. Вызовите команду git status.

```

user@ARIANA MINGW64 /c/D/tmp (master)
$ git commit -m "Добавлен файл README"
[master (root-commit) cfe5770] Добавлен файл README
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md

user@ARIANA MINGW64 /c/D/tmp (master)
$

```

3. Закоммитим их

```

user@ARIANA MINGW64 /c/D/tmp (master)
$ git status
On branch master
nothing to commit, working tree clean

user@ARIANA MINGW64 /c/D/tmp (master)
$

```

```

user@ARIANA MINGW64 /c/D/tmp (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:   README.md

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

user@ARIANA MINGW64 /c/D/tmp (master)
$

```

4. Откроем файл README.txt и напишите в нем «Hello, world!». Сохраняю файл. Проверяю git status снова. Git сообщает о том, что файл был изменён. И он не готов к фиксации (changes not staged for commit).

```

user@ARIANA MINGW64 /c/D/tmp (master)
$ git add README.md

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

```

5. Видим, что измененный файл *README.txt* готов к фиксации.

```

user@ARIANA MINGW64 /c/D/tmp (master)
$ git commit -m "Изменения в README"
[master e7da116] Изменения в README
1 file changed, 1 insertion(+)

```

6. Далее следует зафиксировать изменения (commit).

```

user@ARIANA MINGW64 /c/D/tmp (master)
$ git log
commit e7da1169e7c4138acb8c8b03fbfb8bbf4c4faae9 (HEAD -> master)
Author: = <=>
Date: Thu Jan 15 11:55:35 2026 +0300

    Изменения в README

commit cfe57708250317cd2000e5e34484016e6db9d601
Author: = <=>
Date: Thu Jan 15 11:38:35 2026 +0300

    Добавлен файл README

user@ARIANA MINGW64 /c/D/tmp (master)

```

7. Для того, чтобы просмотреть историю коммитов, нужно использовать команду `gitlog` – это лог всех изменений.

```

user@ARIANA MINGW64 /c/D/tmp (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:   README.md

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

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

```

8. Создайте в нашей папке ещё один файл `test.txt`, напишите внутри него `TEST`. В файле `README` добавьте пару восклицательных знаков или что-либо ещё. Таким образом мы получили два изменения в репозитории.

9. Проверяем текущий статус.

```
user@ARIANA MINGW64 ~  
$ git add README.md  
fatal: not a git repository (or any of the parent directories): .git  
  
user@ARIANA MINGW64 ~  
$ cd C:/D/tmp  
  
user@ARIANA MINGW64 /c/D/tmp (master)  
$ git add README.md  
  
user@ARIANA MINGW64 /c/D/tmp (master)  
$ git add test.txt  
  
user@ARIANA MINGW64 /c/D/tmp (master)  
$ git status  
On branch master  
Changes to be committed:  
  (use "git restore --staged <file>..." to unstage)  
        modified:   README.md  
        new file:   test.txt  
  
user@ARIANA MINGW64 /c/D/tmp (master)  
$ |
```

10. Далее выполняем git add для обоих файлов, чтобы уж окончательно проиндексировать изменения.

```
user@ARIANA MINGW64 /c/D/tmp (master)  
$ git reset HEAD  
Unstaged changes after reset:  
M   README.md  
  
user@ARIANA MINGW64 /c/D/tmp (master)  
$ |
```

11. Всё возвращаем в состояние до индексации.

```
user@ARIANA MINGW64 /c/D/tmp (master)  
$ git checkout test.txt  
Updated 0 paths from the index  
  
user@ARIANA MINGW64 /c/D/tmp (master)  
$ git checkout README.md  
Updated 0 paths from the index  
  
user@ARIANA MINGW64 /c/D/tmp (master)  
$ git status  
On branch master  
Changes to be committed:  
  (use "git restore --staged <file>..." to unstage)  
        new file:   test.txt
```

12. Чтобы вернуться в состояние до изменений, следует использовать команду git checkout test.txt и git checkout README.md.


```
user@ARIANA MINGW64 /c/D/tmp (master)
$ git reset --hard HEAD
HEAD is now at e7da116 Изменения в README
```

13. Применим команду `git reset --hard` и проверяем полученный статус

```
user@ARIANA MINGW64 /c/D/tmp (master)
$ git status
On branch master
nothing to commit, working tree clean

user@ARIANA MINGW64 /c/D/tmp (master)
$
```

14. Всё вернулось на места к состоянию последнего коммита.

```
user@ARIANA MINGW64 /c/D/tmp (master)
$ git log
commit e7da1169e7c4138acb8c8b03fbfb8bbf4c4faae9 (HEAD -> master)
Author: = <=>
Date: Thu Jan 15 11:55:35 2026 +0300

    Изменения в README

commit cfe57708250317cd2000e5e34484016e6db9d601
Author: = <=>
Date: Thu Jan 15 11:38:35 2026 +0300

    Добавлен файл README
```

```
user@ARIANA MINGW64 /c/D/tmp (master)
$ git reset --hard e7da116
HEAD is now at e7da116 Изменения в README
```

15. Попробуем откатиться не на последний, а на предпоследний коммит. Для этого нужно указать его номер (несколько цифр, позволяющих идентифицировать его однозначно). Чтобы знать номера коммитов, сначала выведите лог. Затем выполните откат по номеру предпоследнего коммита.

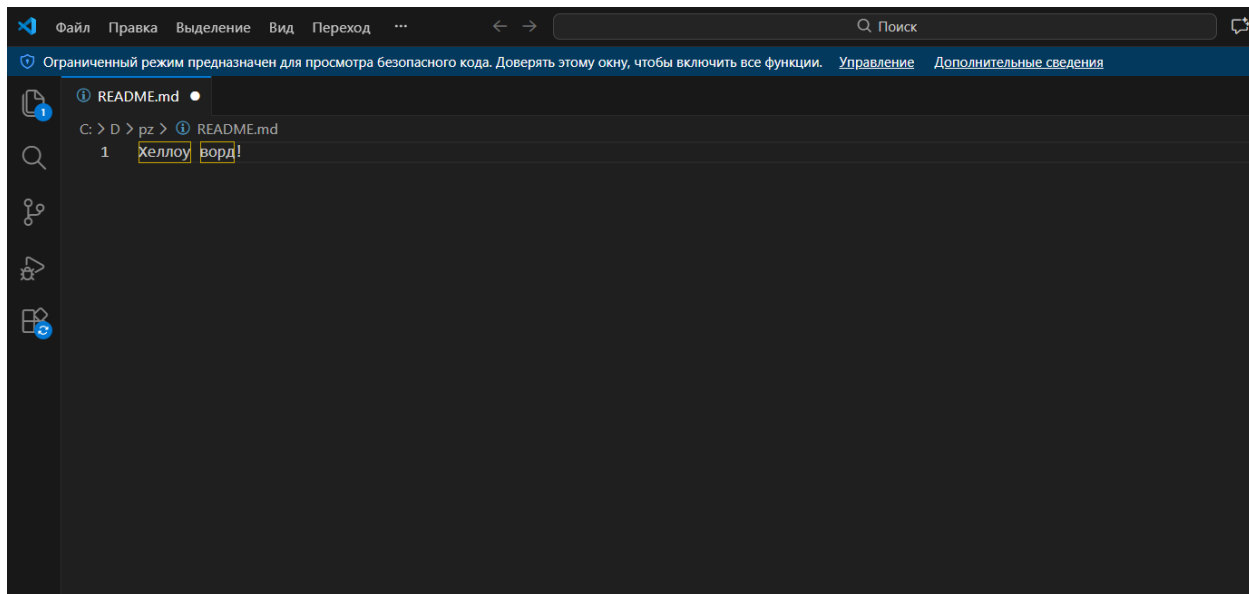
Самостоятельная работа.

```
user@ARIANA MINGW64 ~ (master)
$ cd C:\D\pz
bash: cd: C:Dpz: No such file or directory

user@ARIANA MINGW64 ~ (master)
$ cd C:/D/pz

user@ARIANA MINGW64 /c/D/pz
$ git init
Initialized empty Git repository in C:/D/pz/.git/

user@ARIANA MINGW64 /c/D/pz (master)
$ .....
```



```
user@ARIANA MINGW64 /c/D/pz (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@ARIANA MINGW64 /c/D/pz (master)
```

```

user@ARIANA MINGW64 /c/D/pz (master)
$ git *
git: 'README.md' is not a git command. See 'git --help'.

user@ARIANA MINGW64 /c/D/pz (master)
$ git add *

user@ARIANA MINGW64 /c/D/pz (master)
$ git status
On branch master

No commits yet

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

```

```

user@ARIANA MINGW64 /c/D/pz (master)
$ git add *

user@ARIANA MINGW64 /c/D/pz (master)
$ git commit -m "Добавление 3 файлов"
[master 44ffed6] Добавление 3 файлов
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 1.txt
 create mode 100644 2.txt
 create mode 100644 3.txt

```

```

user@ARIANA MINGW64 /c/D/pz (master)
$ git status
On branch master
nothing to commit, working tree clean

user@ARIANA MINGW64 /c/D/pz (master)
$ git add *

user@ARIANA MINGW64 /c/D/pz (master)
$ git status
bash: git: command not found

user@ARIANA MINGW64 /c/D/pz (master)
$ git status
On branch master

```

```

user@ARIANA MINGW64 /c/D/pz (master)
$ git log
commit 44ffed680dae9cb7a440518ce1e20fa5263f8c58 (HEAD -> master)
Author: = <=>
Date:   Thu Jan 29 09:45:20 2026 +0300

    Добавление 3 файлов

commit c6a15ec113fe4a769ba3779ab9cb4225420d31fc
Author: = <=>
Date:   Thu Jan 29 09:42:13 2026 +0300

    README

```

```

user@ARIANA MINGW64 /c/D/pz (master)
$

```

```

user@ARIANA MINGW64 /c/D/pz (master)
$ git add подпапка

user@ARIANA MINGW64 /c/D/pz (master)
$ git commit -m "Добавление файлов в подпапку"
[master f138394] 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\\1.txt"

```

```

user@ARIANA MINGW64 /c/D/pz (master)
$ rm подпапка/*

user@ARIANA MINGW64 /c/D/pz (master)
$ git add *

user@ARIANA MINGW64 /c/D/pz (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\1.txt"

```

```

user@ARIANA MINGW64 /c/D/pz (master)
$ rmdir подпапка

user@ARIANA MINGW64 /c/D/pz (master)
$ git add *

user@ARIANA MINGW64 /c/D/pz (master)
$ git log --pretty=format:"%h-%s"
f138394-Добавление файлов в подпапку
44ffed6-Добавление 3 файлов
c6a15ec-README

```

Ответы на вопросы:

1. Как отобразить текущий статус репозитория?

Чтобы посмотреть текущий статус репозитория, используется команда:

git status

Она показывает изменённые файлы, новые (untracked) файлы и файлы, готовые к коммиту.

2. Какие файлы называются Untracked files?

Untracked files — это файлы, которые находятся в папке репозитория, но ещё не добавлены под контроль Git.

Git их «видит», но не отслеживает изменения в них.

3. Как передать файл под контроль Git?

Чтобы начать отслеживать файл, используется команда:

git add имя_файла

Например:

git add index.html

Чтобы добавить все файлы сразу:

git add .

4. Как отобразить список изменений в репозитории?

Чтобы посмотреть историю коммитов, используется команда:

git log

Она показывает список всех коммитов с их хешами, автором, датой и сообщением.

5. Как отменить коммит и почему это делать не рекомендуется?

git reset --hard HEAD~1

Это удалит последний коммит вместе с изменениями.

Делать это не рекомендуется, если коммит уже отправлен в удалённый репозиторий, потому что это изменяет историю проекта и может вызвать проблемы у других участников.

6. Как Git идентифицирует все этапы и изменения?

Git использует уникальные хеши (SHA-1 или SHA-256) для каждого коммита.

Каждый коммит получает уникальный идентификатор (например: a3f5c7d...), по которому Git отслеживает историю изменений.

Лабораторная работа №4

```
user@ARIANA MINGW64 /c/D/tmp (master)
$ git clone https://github.com/Rinvvi/GitTestPigoreva.git test
Cloning into 'test'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
```

```
user@ARIANA MINGW64 /c/D/tmp/test (main)
$ git remote -v
origin https://github.com/Rinvvi/GitTestPigoreva.git (fetch)
origin https://github.com/Rinvvi/GitTestPigoreva.git (push)
```

```
user@ARIANA MINGW64 /c/D/tmp/test (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:   hello.txt
```





```
user@ARIANA MINGW64 /c/D/tmp/test (main)
$ git commit -m "hello.txt"
[main eb61aaa] hello.txt
1 file changed, 1 insertion(+)
create mode 100644 hello.txt

user@ARIANA MINGW64 /c/D/tmp/test (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

```
user@ARIANA MINGW64 /c/D/tmp/test (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 255 bytes | 127.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Rinvvi/GitTestPigoreva.git
   af120c1..eb61aaa  main -> main
```

Выполняем команду **gitpush**. Эта команда берет имеющиеся локальные изменения и отправляет их в удалённый репозиторий. В ответ на запрос введите своё имя на Github и пароль.

 = hello.txt		eb61aaa · 1 minute ago	 2 Commits
 README.md	Initial commit	22 minutes ago	
 hello.txt	hello.txt	1 minute ago	

Откроем репозиторий на сайте Github, убедитесь, что файл добавился.

```

user@ARIANA MINGW64 /c/D/tmp/test (main)
$ git pull
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 (from 0)
Unpacking objects: 100% (3/3), 994 bytes | 24.00 KiB/s, done.
From https://github.com/Rinvvi/GitTestPigoreva
   eb61aaa..74bd7b0  main       -> origin/main
Updating eb61aaa..74bd7b0
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)

user@ARIANA MINGW64 /c/D/tmp/test (main)
$ git log
commit 74bd7b0ba830ae5dadadb7ae4fc8adb80f22da0 (HEAD -> main, origin/main, origin/HEAD)
Author: Rinvvi <ariannapigoreva@gmail.com>
Date: Thu Jan 22 11:52:02 2026 +0300

    Update README.md

commit eb61aaa8c7779a72661f7c8e4fd29046cdae8f68
Author: = <=>
Date: Thu Jan 22 11:35:37 2026 +0300

    hello.txt

commit af120c1f04cd3a3c698b20309b6c43f6b1129494
Author: Rinvvi <ariannapigoreva@gmail.com>
Date: Thu Jan 22 11:14:08 2026 +0300

    Initial commit

user@ARIANA MINGW64 /c/D/tmp/test (main)
$

```

Введем команду `git pull`. Эта команда загрузит изменения из удалённого репозитория. Вызываем `git log` и убедимся, что коммитов стало три.

```

user@ARIANA MINGW64 /c/D/tmp/test (main)
$ cat README.md
# GitTestPigoreva
Тестовый репозиторий
Дополнительный тест

user@ARIANA MINGW64 /c/D/tmp/test (main)
$ |

```

Чтобы посмотреть содержимое обновленного файла, введем команду `cat README.md`

Самостоятельная работа.

```

user@ARIANA MINGW64 ~ (master)
$ cd C:/D/lol

user@ARIANA MINGW64 /c/D/lol
$ git init
Initialized empty Git repository in C:/D/lol/.git/

```

```
user@ARIANA MINGW64 /c/D/loI (master)
$ git remote add origin https://github.com/Rinvvi/GitTestPigoreva.git

user@ARIANA MINGW64 /c/D/loI (master)
$ git remote -v
origin https://github.com/Rinvvi/GitTestPigoreva.git (fetch)
origin https://github.com/Rinvvi/GitTestPigoreva.git (push)
```

```
user@ARIANA MINGW64 /c/D/loI (master)
$ git fetch origin
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 9 (delta 1), reused 3 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (9/9), 2.00 KiB | 35.00 KiB/s, done.
From https://github.com/Rinvvi/GitTestPigoreva
* [new branch]      main      -> origin/main
```

Ответы на вопросы:

1. Как отправить изменения на удалённый репозиторий?

Чтобы отправить изменения на удалённый репозиторий (например, на GitHub), используется команда:

```
git push origin имя_ветки
```

Например:

```
git push origin main
```

2. Как влить изменения в локальную ветку из удалённого репозитория?

Чтобы получить изменения из удалённого репозитория и объединить их с локальной веткой, используется команда:

```
git pull
```

Или конкретно:

```
git pull origin main
```

3. Какой командой можно зафиксировать изменения в репозитории?

Чтобы сохранить изменения (создать коммит), используется команда:

```
git commit -m "Сообщение коммита"
```

4. Как используется GitHub в реальной работе?

GitHub используется для:

- хранения кода в удалённом репозитории;
- совместной работы над проектами;
- отслеживания изменений;
- управления задачами (Issues);
- проверки кода (Pull Request);
- автоматического тестирования и развертывания (GitHub Actions).

5. Опишите принцип коллективной работы с GitHub

1. Создаётся общий репозиторий.
2. Каждый разработчик создаёт свою ветку.
3. В своей ветке он вносит изменения и делает коммиты.
4. Затем создаётся Pull Request (запрос на слияние).
5. Другие участники проверяют код.
6. После проверки изменения сливаются в основную ветку (main).

Лабораторная работа №5

```
user@ARIANA MINGW64 ~ (master)
$ cd C:/D/tmp2

user@ARIANA MINGW64 /c/D/tmp2
$ git init
Initialized empty Git repository in C:/D/tmp2/.git/

user@ARIANA MINGW64 /c/D/tmp2 (master)
$ cd C:/D/Repo2

user@ARIANA MINGW64 /c/D/Repo2
$ git init
Initialized empty Git repository in C:/D/Repo2/.git/

user@ARIANA MINGW64 /c/D/Repo2 (master)
$ git add .

user@ARIANA MINGW64 /c/D/Repo2 (master)
$ git status
On branch master

No commits yet

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

user@ARIANA MINGW64 /c/D/Repo2 (master)
$ git commit -m "Init"
[master (root-commit) 5d97474] Init
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.txt

user@ARIANA MINGW64 /c/D/Repo2 (master)
$
```

```
user@ARIANA MINGW64 ~ (master)
$ cd C:/D/tmp2

user@ARIANA MINGW64 /c/D/tmp2 (master)
$ git remote add super2 c/D/Repo2

user@ARIANA MINGW64 /c/D/tmp2 (master)
$ git remote
super2

user@ARIANA MINGW64 /c/D/tmp2 (master)
$
```

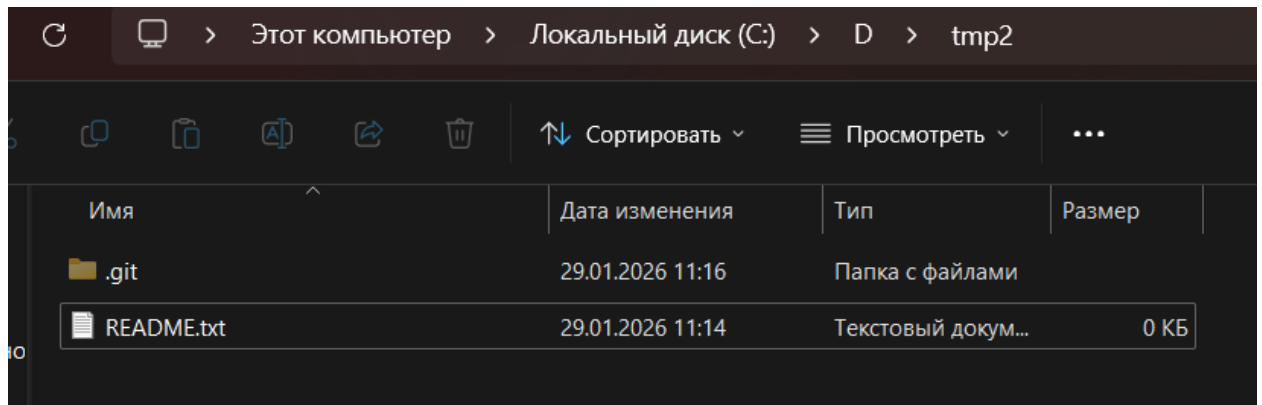
```
user@ARIANA MINGW64 /c/D/tmp2 (master)
$ git remote -v
super2 c/D/Repo2 (fetch)
super2 c/D/Repo2 (push)

user@ARIANA MINGW64 /c/D/tmp2 (master)
$
```

```
user@ARIANA MINGW64 /c/D/tmp2 (master)
$ git fetch super
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), 165 bytes | 11.00 KiB/s, done.
From C:/D/Repo2
* [new branch]      master      -> super/master
```

```
user@ARIANA MINGW64 /c/D/tmp2 (master)
$ git checkout --track super/master
branch 'master' set up to track 'super/master'.
Already on 'master'
```

```
user@ARIANA MINGW64 /c/D/tmp2 (master)
$ git pull
Already up to date.
```



Ответы на вопрос:

1. Что такое ветка?

Ветка (branch) — это отдельная линия разработки в Git.

Она позволяет работать над новыми функциями или исправлениями независимо от основной версии проекта. По умолчанию основная ветка обычно называется main (или master).

2. На какие типы делятся репозитории Git?

Локальные — находятся на компьютере пользователя.

Удалённые — размещены на сервере (например, GitHub, GitLab, Bitbucket).

3. Как добавить ссылку на удалённый репозиторий?

Чтобы подключить удалённый репозиторий, используется команда:

```
git remote add origin ссылка_на_репозиторий
```

4. Как вывести подробный список прикрепленных репозиториев?

```
git remote -v
```

5. Что выполняет команда git remote show?

Команда

```
git remote show origin
```

показывает подробную информацию об удалённом репозитории:

- URL-адрес,
- отслеживаемые ветки,
- информацию о push и pull,
- состояние синхронизации веток.

Лабораторная работа №6

```
user@ARIANA MINGW64 ~ (master)
$ cd C:/D/branches

user@ARIANA MINGW64 /c/D/branches
$ git init
Initialized empty Git repository in C:/D/branches/.git/

user@ARIANA MINGW64 /c/D/branches (master)
$ git tag

user@ARIANA MINGW64 /c/D/branches (master)
$ git status
On branch master

No commits yet

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

user@ARIANA MINGW64 /c/D/branches (master)
$ git commit -m "init"
On branch master

Initial commit

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

user@ARIANA MINGW64 /c/D/branches (master)
$ git add .

user@ARIANA MINGW64 /c/D/branches (master)
$ git commit -m "init"
[master (root-commit) 1180110] init
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt

user@ARIANA MINGW64 /c/D/branches (master)
$
```

```
user@ARIANA MINGW64 /c/D/branches (master)
$ git tag 1.0

user@ARIANA MINGW64 /c/D/branches (master)
$ git show
commit 1180110cf5f9106d4bf40d1c32dded2e2fcc72b2 (HEAD -> master, tag: 1.0)
Author: = <=>
Date: Thu Jan 29 11:45:12 2026 +0300

    init

diff --git a/1.txt b/1.txt
new file mode 100644
index 0000000..e69de29

user@ARIANA MINGW64 /c/D/branches (master)
$ |
```

```
user@ARIANA MINGW64 /c/D/branches (master)
$ git tag -l "1.*"
1.0
```

Ответы на вопросы:

1. Что такое метка?

Метка (tag) — это специальная ссылка в Git на определённый коммит.

Обычно используется для обозначения версий проекта (например, v1.0, v2.1).

2. Какие типы меток существуют?

Существуют два типа меток:

- **Лёгкие (lightweight)** — просто указатель на коммит.
- **Аннотированные (annotated)** — содержат дополнительную информацию (автор, дата, сообщение) и сохраняются как отдельный объект в Git.

3. Как добавляется метка к репозиторию?

Лёгкая метка:

```
git tag v1.0
```

Аннотированная метка:

```
git tag -a v1.0 -m "Версия 1.0"
```

4. Как вывести информацию по метке?

Чтобы посмотреть информацию о метке, используется команда:

```
git show v1.0
```

Также можно вывести список всех меток:

```
git tag
```

5. Для чего нужен сабмодуль?

Сабмодуль (submodule) — это способ подключить один Git-репозиторий внутри другого.

Он используется, когда нужно включить внешний проект или библиотеку в свой репозиторий, сохраняя связь с его оригинальной версией.

6. Как добавить сабмодуль?

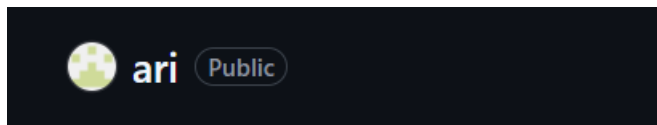
Чтобы добавить сабмодуль, используется команда:

```
git submodule add ссылка_на_репозиторий
```

Лабораторная работа №7

Самостоятельная работа.

Создание репозитория на GitHub



Инициализация репозитория локально

Перейди в папку, где будет проект:

```
user@ARIANA MINGW64 ~ (master)
$ cd Desktop

user@ARIANA MINGW64 ~/Desktop (master)
$ mkdir profile-page

user@ARIANA MINGW64 ~/Desktop (master)
$ cd profile-page
```

Инициализируй репозиторий:

```
user@ARIANA MINGW64 ~/Desktop/profile-page (master)
$ git init
Initialized empty Git repository in C:/Users/user/Desktop/profile-page/.git/
```

Подключи удалённый репозиторий:

```
user@ARIANA MINGW64 ~/Desktop/profile-page (master)
$ git remote add origin https://github.com/Rinvvi/ari.git
```

Создание файла index.html и первый коммит

```
user@ARIANA MINGW64 ~/Desktop/profile-page (master)
$ touch index.html
```

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <meta charset="UTF-8">
5  |   <title>Мой профиль</title>
6  </head>
7  <body>
8  |   <h1>Обо мне</h1>
9  |   <p>Меня зовут Пигорева Арианна Я начинающий разработчик.</p>
10 |   <p>Интересуюсь веб-разработкой и Git.</p>
11 </body>
12 </html>
13 |
```

Фиксация изменений

```
user@ARIANA MINGW64 ~/Desktop/profile-page (master)
$ git status
On branch master

No commits yet

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

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

user@ARIANA MINGW64 ~/Desktop/profile-page (master)
$ git add index.html

user@ARIANA MINGW64 ~/Desktop/profile-page (master)
$ git commit -m "Добавлен файл index.html с профилем"
[master (root-commit) 3ef8f1e] добавлен файл index.html с профилем
 1 file changed, 12 insertions(+)
 create mode 100644 index.html
```

Отправь в GitHub:

```
user@ARIANA MINGW64 ~/Desktop/profile-page (master)
$ git branch -M master

user@ARIANA MINGW64 ~/Desktop/profile-page (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 479 bytes | 239.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Rinnvi/ari.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

кнопка «Подписаться на новости»


```
user@ARIANA MINGW64 ~/Desktop/profile-page (master)
$ git checkout -b add-subscribe-button
Switched to a new branch 'add-subscribe-button'
```

Добавление кнопки

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="UTF-8">
5      <title>Мой профиль</title>
6  </head>
7  <body>
8      <h1>Обо мне</h1>
9      <p>Меня зовут Пигорева Арианна Я начинающий разработчик.</p>
10     <p>Интересуюсь веб-разработкой и Git.</p>
11     <button>Подписаться на новости</button>
12
13 </body>
14 </html>
15
```

Коммит и отправка ветки

```
user@ARIANA MINGW64 ~/Desktop/profile-page (add-subscribe-button)
$ git add index.html

user@ARIANA MINGW64 ~/Desktop/profile-page (add-subscribe-button)
$ git commit -m "Добавлен кнопка подписки"
[add-subscribe-button 3c958d9] добавлен кнопка подписки
1 file changed, 2 insertions(+)

user@ARIANA MINGW64 ~/Desktop/profile-page (add-subscribe-button)
$ git push -u origin add-subscribe-button
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 358 bytes | 358.00 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/Rinnvi/ari/pull/new/add-subscribe-button
remote:
To https://github.com/Rinnvi/ari.git
 * [new branch]      add-subscribe-button -> add-subscribe-button
branch 'add-subscribe-button' set up to track 'origin/add-subscribe-button'.
```

Слияние с master

```
user@ARIANA MINGW64 ~/Desktop/profile-page (add-subscribe-button)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

user@ARIANA MINGW64 ~/Desktop/profile-page (master)
$ git merge add-subscribe-button
Updating 3ef8f1e..3c958d9
Fast-forward
 index.html | 2 ++
1 file changed, 2 insertions(+)
```

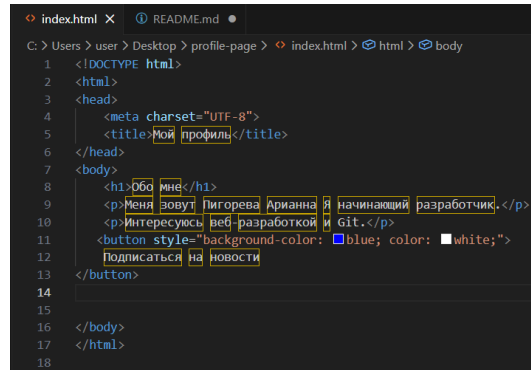
Отправка изменений

```
user@ARIANA MINGW64 ~/Desktop/profile-page (master)
$ git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Rinnvi/ari.git
 3ef8f1e..3c958d9 master -> master
```

Ветка для изменения цвета кнопки

```
user@ARIANA MINGW64 ~/Desktop/profile-page (master)
$ git checkout -b change-button-color
Switched to a new branch 'change-button-color'
```

Измени кнопку:



```
index.html X  README.md
C: > Users > user > Desktop > profile-page > index.html > html > body
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <meta charset="UTF-8">
5    <title>Мой профиль</title>
6  </head>
7  <body>
8    <h1>Обо мне</h1>
9    <p>Меня зовут Пигорева Арианна, я начинающий разработчик.</p>
10   <p>Интересуюсь веб-разработкой и Git.</p>
11   <button style="background-color: blue; color: white;">
12     Подписаться на новости
13   </button>
14
15
16 </body>
17 </html>
18
```

```
user@ARIANA MINGW64 ~/Desktop/profile-page (change-button-color)
$ git add index.html

user@ARIANA MINGW64 ~/Desktop/profile-page (change-button-color)
$ git commit -m "Изменен цвет кнопки"
[change-button-color 3f7af1e] Изменен цвет кнопки
1 file changed, 4 insertions(+), 1 deletion(-)

user@ARIANA MINGW64 ~/Desktop/profile-page (change-button-color)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

user@ARIANA MINGW64 ~/Desktop/profile-page (master)
$ git merge change-button-color
Updating 3c958d9..3f7af1e
Fast-forward
 index.html | 5 ++++-
1 file changed, 4 insertions(+), 1 deletion(-)

user@ARIANA MINGW64 ~/Desktop/profile-page (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 361 bytes | 361.00 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.
To https://github.com/Rinvvi/ari.git
 3c958d9..3f7af1e master -> master
```

Ветка для формы входа через соцсети

```
user@ARIANA MINGW64 ~/Desktop/profile-page (master)
$ git checkout -b add-social-login
Switched to a new branch 'add-social-login'
```

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4    <meta charset="UTF-8">
5    <title>Мой профиль</title>
6  </head>
7  <body>
8    <h1>Обо мне</h1>
9    <p>Меня зовут Лигорева Арианна Я начинающий разработчик.</p>
10   <p>Интересуюсь веб-разработкой и Git.</p>
11   <button style="background-color: blue; color: white;"
12     Подписаться на новости
13 </button>
14
15 <form>
16   <h2>Вход через соцсети</h2>
17   <button>Войти через VK</button>
18   <button>Войти через Google</button>
19 </form>
20
21 </body>
22 </html>
23

```

```

user@ARIANA MINGW64 ~/Desktop/profile-page (add-social-login)
$ git add index.html

user@ARIANA MINGW64 ~/Desktop/profile-page (add-social-login)
$ git commit -m "Добавлена форма входа через соцсети"
[add-social-login 76a5a1a] Добавлена форма входа через соцсети
1 file changed, 5 insertions(+)

user@ARIANA MINGW64 ~/Desktop/profile-page (add-social-login)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

user@ARIANA MINGW64 ~/Desktop/profile-page (master)
$ git merge add-social-login
Updating 3f7af1e..76a5a1a
Fast-forward
 index.html | 5 +++++
1 file changed, 5 insertions(+)

user@ARIANA MINGW64 ~/Desktop/profile-page (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 415 bytes | 415.00 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.
To https://github.com/Rinvvi/ari.git
 3f7af1e..76a5a1a master -> master

```

Создай две ветки из master

```

user@ARIANA MINGW64 ~/Desktop/profile-page (master)
$ git checkout -b conflict-branch-1
Switched to a new branch 'conflict-branch-1'

```

```
user@ARIANA MINGW64 ~/Desktop/profile-page (conflict-branch-1)
$ git commit -am "Изменение текста профиля (ветка 1)"
[conflict-branch-1 c97b912] Изменение текста профиля (ветка 1)
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
user@ARIANA MINGW64 ~/Desktop/profile-page (conflict-branch-1)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

user@ARIANA MINGW64 ~/Desktop/profile-page (master)
$ git checkout -b conflict-branch-2
Switched to a new branch 'conflict-branch-2'
```

```
user@ARIANA MINGW64 ~/Desktop/profile-page (conflict-branch-2)
$ git commit -am "Изменение текста профиля (ветка 2)"
[conflict-branch-2 9091281] Изменение текста профиля (ветка 2)
1 file changed, 1 insertion(+), 1 deletion(-)
```

Слияние с конфликтом

```
user@ARIANA MINGW64 ~/Desktop/profile-page (conflict-branch-2)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

user@ARIANA MINGW64 ~/Desktop/profile-page (master)
$ git merge conflict-branch-1
Updating 76a5a1a..c97b912
Fast-forward
 index.html | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

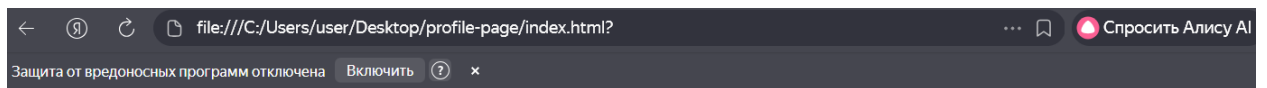
user@ARIANA MINGW64 ~/Desktop/profile-page (master)
$ git merge conflict-branch-1
Already up to date.
```

Решение конфликта

```
user@ARIANA MINGW64 ~/Desktop/profile-page (master)
$ git add index.html

user@ARIANA MINGW64 ~/Desktop/profile-page (master)
$ git commit -m "Разрешен конфликт"
[master f15ee1d] Разрешен конфликт
1 file changed, 1 insertion(+), 1 deletion(-)

user@ARIANA MINGW64 ~/Desktop/profile-page (master)
$ git push
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 586 bytes | 195.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/Rinvvi/ari.git
76a5a1a..f15ee1d master -> master
```



Обо мне

Меня зовут Пигорева Арианна Я начинающий разработчик, учусь в колледже на 2 курсе.

Интересуюсь веб-разработкой и Git.

[Подписаться на новости](#)

Вход через соцсети

[Войти через VK](#)

[Войти через Google](#)

Ответы на вопросы:

1. Какая ветка создаётся сразу после создания репозитория?

По умолчанию создаётся основная ветка — **main**
(в старых версиях Git она называлась master).

2. Как задать новую ветку?

Создать новую ветку можно командой:

```
git branch имя_ветки
```

Например:

```
git branch feature1
```

Создать и сразу перейти на неё:

```
git checkout -b feature1
```

(или в новой версии Git:)

```
git switch -c feature1
```

3. Что такое HEAD и какая взаимосвязь между git checkout?

HEAD — это указатель на текущий коммит и текущую ветку, в которой вы работаете.

Когда выполняется команда:

```
git checkout имя_ветки
```

HEAD перемещается на выбранную ветку.
То есть `git checkout` меняет положение HEAD.

4. Как произвести слияние веток?

Сначала нужно перейти в ветку, в которую будут вливаться изменения:

```
git checkout main
```

Затем выполнить слияние:

```
git merge имя_ветки
```

5. Что значит fast-forward?

Fast-forward — это тип слияния, при котором Git просто «перемещает» указатель ветки вперёд, если не было параллельных изменений.

То есть история остаётся линейной, без создания дополнительного коммита слияния.