

Лабораторная работа №9

4)

```
vadim@DESKTOP-IMNOJB5 MINGW64 ~  
$ git version  
git version 2.47.1.windows.1  
  
vadim@DESKTOP-IMNOJB5 MINGW64 ~  
$ |
```

5) Я воспользовался командой “git help”

```
vadim@DESKTOP-IMNOJB5 MINGW64 ~  
$ git help  
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]  
        [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]  
        [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--no-lazy  
-fetch]  
        [--no-optional-locks] [--no-advice] [--bare] [--git-dir=<path>]  
        [--work-tree=<path>] [--namespace=<name>] [--config-env=<name>=<envva  
r>]  
        <command> [<args>]  
  
These are common Git commands used in various situations:  
  
start a working area (see also: git help tutorial)  
  clone      Clone a repository into a new directory  
  init       Create an empty Git repository or reinitialize an existing one  
  
work on the current change (see also: git help everyday)  
  add        Add file contents to the index  
  mv         Move or rename a file, a directory, or a symlink  
  restore    Restore working tree files  
  rm         Remove files from the working tree and from the index  
  
examine the history and state (see also: git help revisions)  
  bisect     Use binary search to find the commit that introduced a bug  
  diff       Show changes between commits, commit and working tree, etc  
  grep       Print lines matching a pattern  
  log        Show commit logs  
  show       Show various types of objects  
  status     Show the working tree status  
  
grow, mark and tweak your common history  
  backfill   Download missing objects in a partial clone  
  branch     List, create, or delete branches  
  commit     Record changes to the repository  
  merge      Join two or more development histories together  
  rebase     Reapply commits on top of another base tip  
  reset      Reset current HEAD to the specified state  
  switch     Switch branches  
  tag        Create, list, delete or verify a tag object signed with GPG  
  
collaborate (see also: git help workflows)  
  fetch      Download objects and refs from another repository  
  pull       Fetch from and integrate with another repository or a local branch  
  push       Update remote refs along with associated objects  
  
'git help -a' and 'git help -g' list available subcommands and some  
concept guides. See 'git help <command>' or 'git help <concept>'  
to read about a specific subcommand or concept.  
See 'git help git' for an overview of the system.
```

Эта команда выводит общее руководство по Git, где можно найти информацию о всех доступных командах.

6) Перешёл в проводнике в рабочую папку и запустил Git Bash с помощью контекстного меню. Инициализация репозитория в выбранной папке выполняется командой “git init”

```
vadim@DESKTOP-IMNOJB5 MINGW64 /Repository
$ git init
Initialized empty Git repository in C:/Git/Repository/.git/
```

7)-8)

```
vadim@DESKTOP-IMNOJB5 MINGW64 /
$ git config --global user.name "Потребко Вадим"

vadim@DESKTOP-IMNOJB5 MINGW64 /
$ git config --global user.email vadim26129@gmail.com

vadim@DESKTOP-IMNOJB5 MINGW64 /
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
init.defaultbranch=master
user.name=Потребко Вадим
user.email=vadim26129@gmail.com
```

9)-12) С помощью команды “git add Hello.txt” я проиндексировал данный файл. Также с помощью команды “git status” я могу узнать текущее состояние репозитория.

```

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (master)
$ git status
On branch master

No commits yet

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

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

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (master)
$ git add Hello.txt

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (master)
$ git status
On branch master

No commits yet

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

```

13)-14) Я зафиксировал изменения в репозитории с помощью команды "added Hello.txt to the repo". Ключ -m позволяет добавить комментарий, описывающий, что именно было изменено в коммите ("added Hello.txt to the repo"). Git информирует об успешном создании нового коммита.

```

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (master)
$ git commit -m "added Hello.txt to the repo"
[master (root-commit) 9cc00f6] added Hello.txt to the repo
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Hello.txt

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (master)
$ git log
commit 9cc00f66663f1435e895c1c0394659eb7e595912 (HEAD -> master)
Author: Потребко Вадим <vadim26129@gmail.com>
Date:   Mon Dec 2 03:06:55 2024 +0300

    added Hello.txt to the repo

```

Лабораторная №10

3) Для создания новой ветки test и переключения в нее использовала команды "git branch test" "git checkout test". Чтобы объединить обе команды в одну, можно использовать "git checkout -b test".

```

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (master)
$ git branch test

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (master)
$ git chteckout test
git: 'chteckout' is not a git command. See 'git --help'.

The most similar command is
    checkout

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (master)
$ git branch
* master
  test

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (master)
$ git checkout test
Switched to branch 'test'

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (test)
$ git branch
  master
* test

```

4) В результате выполнения команды указатель HEAD сдвигается на ветку test.

```

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (test)
$ git status
On branch test
nothing to commit, working tree clean

```

HEAD указывает на ветку test:

```

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (test)
$ git log
commit 9cc00f66663f1435e895c1c0394659eb7e595912 (HEAD -> test, master)
Author: Потребко Вадим <vadim26129@gmail.com>
Date:   Mon Dec 2 03:06:55 2024 +0300

    added Hello.txt to the repo

```

5) Чтобы увидеть содержимое всех файлов в папке проекта, воспользовалась командой "ls"

```

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (test)
$ ls
Hello.txt

```

6)-7) Создала новый текстовый файл test.html. Просмотрел все ветки.

```
vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (test)
$ git add test.html

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (test)
$ git commit -m "added test.html to the repo"
[test 436a1b6] added test.html to the repo
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 test.html

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (test)
$ git status
On branch test
nothing to commit, working tree clean

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (test)
$ git branch
  master
* test
```

8) Переключился обратно в ветку master и выполнила слияние с веткой test с помощью команды "git merge test"/

```
vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (test)
$ git checkout master
Switched to branch 'master'

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (master)
$ git merge test
Updating 9cc00f6..436a1b6
Fast-forward
 test.html | 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 test.html
```

9) В ветке master внес изменения в файл test.html, сохранив его. Затем зашел в ветку test, внес изменения в тот же файл и выполнил "git add test.html" и "git commit -m". Теперь, когда я попытался выполнить слияние ветки test обратно в master, возник конфликт. Git уведомил о конфликте. Я открыла файл test.html, разрешила конфликт вручную, устранив конфликтующие строки, затем сохранил файл.

```

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (master)
$ git branch
* master
  test

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (master)
$ git checkout test
Switched to branch 'test'

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (test)
$ git add test
fatal: pathspec 'test' did not match any files

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (test)
$ git add test.html

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (test)
$ git commit -m "Изменен файл test.html в ветке test"
[test 586e344] Изменен файл test.html в ветке test
1 file changed, 11 insertions(+)

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (test)
$ git checkout master
Switched to branch 'master'

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (master)
$ git merge test
Updating 436a1b6..586e344
Fast-forward
 test.html | 11 ++++++++
1 file changed, 11 insertions(+)

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (master)
$ git add test.html

vadim@DESKTOP-IMNOJB5 MINGW64 /Repository (master)
$ git commit -m "Конфликт в test.html разрешен"
[master 9ae00bb] Конфликт в test.html разрешен
1 file changed, 1 deletion(-)

```

Лабораторная №11

3)-9) Я создал аккаунт. Создал репозиторий. Создал в нем файл и добавил нужные данные. Также изучил его.

10) Выполнил клонирование удаленного репозитория в Git.

```
vadim@DESKTOP-IMNOJB5 MINGW64 ~  
$ git clone https://github.com/TaLiCmAn4IK/OPI_lab11  
Cloning into 'OPI_lab11'...  
remote: Enumerating objects: 6, done.  
remote: Counting objects: 100% (6/6), done.  
remote: Compressing objects: 100% (3/3), done.  
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
Receiving objects: 100% (6/6), done.
```

11) Конфигурация репозитория

```
core.ignorecase=true  
remote.origin.url=https://github.com/TaLiCmAn4IK/OPI_lab_11  
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*  
branch.main.remote=origin
```

12) Внес изменения в файл README.md и зафиксировал изменения в Git.

```
vadim@DESKTOP-IMNOJB5 MINGW64 ~/OPI_lab_11 (main)  
$ cat README.MD  
# OPI_lab_11  
Потребко  
Вадим  
Русланович  
Программная инженерия  
ОПИ  
курс 1, группа 6, подгруппа 1
```

13) Просмотрел список удаленных репозиторияев.

```
vadim@DESKTOP-IMNOJB5 MINGW64 ~/OPI_lab_11 (main)  
$ git remote  
origin  
  
vadim@DESKTOP-IMNOJB5 MINGW64 ~/OPI_lab_11 (main)  
$ git log  
commit 2fd547fa777a1be4f81972a6db8ef4ad0ab7b631 (HEAD -> main)  
Author: Потребко Вадим <vadim26129@gmail.com>  
Date: Mon Dec 9 04:32:41 2024 +0300  
  
    Updated from local repo  
  
commit 775292cc02c6ed0137f6d31cf8c926d6edb542eb (origin/main, origin/HEAD)  
Author: TaLiCmAn4IK <vadim26129@gmail.com>  
Date: Mon Dec 9 04:03:02 2024 +0300  
  
    Update README.md  
  
commit 8e54852a94f62f34a318ccf98f7f21ec2c29b1f5  
Author: TaLiCmAn4IK <vadim26129@gmail.com>  
Date: Mon Dec 9 04:00:34 2024 +0300  
  
    Initial commit
```

```
vadim@DESKTOP-IMNOJB5 MINGW64 ~/OPI_lab_11 (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)
    OPI_lab_11/

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

vadim@DESKTOP-IMNOJB5 MINGW64 ~/OPI_lab_11 (main)
$ git add .
warning: adding embedded git repository: OPI_lab_11
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:
hint:   git submodule add <url> OPI_lab_11
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint:   git rm --cached OPI_lab_11
hint:
hint: See "git help submodule" for more information.
hint: Disable this message with "git config advice.addEmbeddedRepo false"

vadim@DESKTOP-IMNOJB5 MINGW64 ~/OPI_lab_11 (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:   OPI_lab_11
```

```
vadim@DESKTOP-IMNOJB5 MINGW64 ~/OPI_lab_11 (main)
$ git commit -m "Updated from local repo"
[main 2fd547f] Updated from local repo
 1 file changed, 1 insertion(+)
 create mode 160000 OPI_lab_11
```