

# Отчет: Работа с Git. Дополнительное задание

Выполнила: Чикомазова Алиса 504497

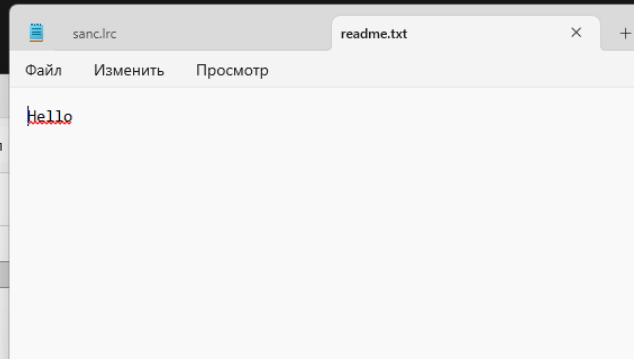
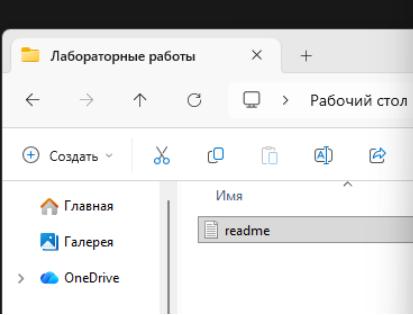
## 1. Базовые операции с файлами

Создание файла "Hello" > readme.txt` - Создан текстовый файл

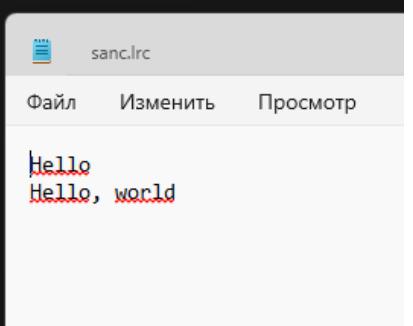
Добавление текста "Hello, world" >> readme.txt` - Текст добавлен в конец файла

Просмотр содержимого `cat readme.txt` - Проверка содержимого файла

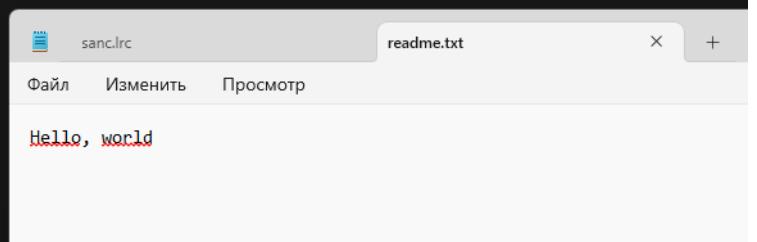
```
● PS C:\Users\user\Desktop\python\Лабораторные работы> echo "Hello" > readme.txt
○ PS C:\Users\user\Desktop\python\Лабораторные работы> 
```



```
● PS C:\Users\user\Desktop\python\Лабораторные работы> "Hello, world" >> .\readme.txt
○ PS C:\Users\user\Desktop\python\Лабораторные работы> 
```



```
● PS C:\Users\user\Desktop\python\Лабораторные работы> "Hello, world" > .\readme.txt
○ PS C:\Users\user\Desktop\python\Лабораторные работы> 
```



## **2. Работа с Git: Основной цикл**

Инициализация `git init` - Репозиторий создан  
Добавление файлов `git add readme.txt` - Файл добавлен в индекс  
Создание коммита `git commit -m "first file"` - Первый коммит создан  
Проверка статуса `git status` - Отслеживание изменений

```
● PS C:\Users\user\Desktop\python\Лабораторные работы> "Hello, world" > .\readme.txt
● PS C:\Users\user\Desktop\python\Лабораторные работы> git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
● PS C:\Users\user\Desktop\python\Лабораторные работы> git add readme.txt
● PS C:\Users\user\Desktop\python\Лабораторные работы> git status
On branch master

No commits yet

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

● PS C:\Users\user\Desktop\python\Лабораторные работы> git commit -m 'first file'
[master (root-commit) b7dafc4] first file
  1 file changed, 1 insertion(+)
  create mode 100644 README.txt
● PS C:\Users\user\Desktop\python\Лабораторные работы> git status
On branch master
nothing to commit, working tree clean
```

## **3. Множественные файлы и паттерны**

Добавление нескольких файлов `git add \*.txt` - Все `.txt` файлы добавлены  
Добавление всех изменений `git add .` - Все новые/измененные файлы  
Работа с папками `git add folder/` - Добавлена папка с содержимым

```

● PS C:\Users\user\Desktop\python\Лабораторные работы> git add readme.txt
● PS C:\Users\user\Desktop\python\Лабораторные работы> git status
On branch master

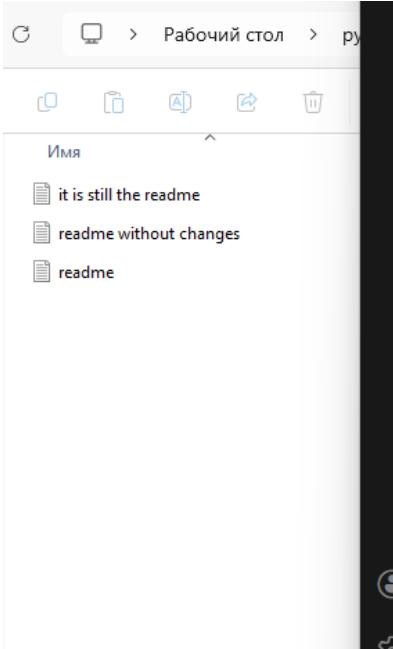
No commits yet

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

● PS C:\Users\user\Desktop\python\Лабораторные работы> git commit -m 'first file'
[master (root-commit) b7dafc4] first file
  1 file changed, 1 insertion(+)
  create mode 100644 readme.txt
● PS C:\Users\user\Desktop\python\Лабораторные работы> git status
On branch master
nothing to commit, working tree clean
● PS C:\Users\user\Desktop\python\Лабораторные работы> 'and everyone else' >> .\readme.txt
● PS C:\Users\user\Desktop\python\Лабораторные работы> 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.txt

no changes added to commit (use "git add" and/or "git commit -a")
● PS C:\Users\user\Desktop\python\Лабораторные работы> git add readme.txt
● PS C:\Users\user\Desktop\python\Лабораторные работы> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   readme.txt

```



```

no changes added to commit (use "git add" and/or "git commit -a")
● PS C:\Users\user\Desktop\python\Лабораторные работы> git add readme.txt
● PS C:\Users\user\Desktop\python\Лабораторные работы> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   readme.txt

● PS C:\Users\user\Desktop\python\Лабораторные работы> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   readme.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    it is still the readme.txt
    readme without changes.txt

● PS C:\Users\user\Desktop\python\Лабораторные работы> git add "*.txt"
● PS C:\Users\user\Desktop\python\Лабораторные работы> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   it is still the readme.txt
    new file:   readme without changes.txt
    modified:   readme.txt

```

#### 4. Анализ и история

Просмотр истории `git log` - Список всех коммитов

Детальный лог `git log --summary` - Показывает созданные файлы Сравнение изменений `git diff` - Показывает непроиндексированные изменения

Сравнение с HEAD `git diff HEAD` - Все изменения с последнего коммита

Сравнение индекса `git diff --staged` - Изменения готовые к коммиту

```
● PS C:\Users\user\Desktop\python\Лабораторные работы> git commit -m 'all files'
[master 4c7182b] all files
 3 files changed, 5 insertions(+)
    create mode 100644 it is still the readme.txt
    create mode 100644 readme without changes.txt
● PS C:\Users\user\Desktop\python\Лабораторные работы> git status
On branch master
nothing to commit, working tree clean
● PS C:\Users\user\Desktop\python\Лабораторные работы> git log
commit 4c7182b167fcfdbf9c9ffac4fd5a24df5924b9bb (HEAD -> master)
Author: a14759226-glitch <a14759226@gmail.com>
Date:   Mon Dec 15 21:58:17 2025 +0300

    all files

commit b7dafc4761000cf14d1f90a06dcc0eff3f496f66
Author: a14759226-glitch <a14759226@gmail.com>
Date:   Mon Dec 15 21:50:54 2025 +0300

    first file
```

```
● PS C:\Users\user\Desktop\python\Лабораторные работы> git log --summary
commit 4c7182b167fcfdbf9c9ffac4fd5a24df5924b9bb (HEAD -> master)
Author: a14759226-glitch <a14759226@gmail.com>
Date:   Mon Dec 15 21:58:17 2025 +0300

    all files

    create mode 100644 it is still the readme.txt
    create mode 100644 readme without changes.txt

commit b7dafc4761000cf14d1f90a06dcc0eff3f496f66
Author: a14759226-glitch <a14759226@gmail.com>
Date:   Mon Dec 15 21:50:54 2025 +0300

    first file

    create mode 100644 readme.txt
○ PS C:\Users\user\Desktop\python\Лабораторные работы>
```

## 5. Работа с удаленными репозиториями

Просмотр удаленных `git remote -v` - Показаны привязанные репозитории  
Добавление remote `git remote add origin URL` - Привязка к GitHub  
Удаление remote `git remote remove origin` - Отвязка от репозитория  
Отправка на GitHub `git push -u origin master` - Успешная отправка файлов

```

PS C:\Users\user\Desktop\python\Лабораторные работы> git remote -v
origin https://github.com/ctel-master/demo-repo.git (fetch)
origin https://github.com/ctel-master/demo-repo.git (push)
PS C:\Users\user\Desktop\python\Лабораторные работы> git remote remove origin
PS C:\Users\user\Desktop\python\Лабораторные работы> git remote -v
PS C:\Users\user\Desktop\python\Лабораторные работы> git remote add origin https://github.com/a14759226-glitch/test_py.git
PS C:\Users\user\Desktop\python\Лабораторные работы> git push -u origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 503 bytes | 167.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/a14759226-glitch/test_py.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
PS C:\Users\user\Desktop\python\Лабораторные работы>

```

**test\_py** Public

master 1 Branch 0 Tags Go to file Add file Code

	a14759226-glitch all files	4c7182b · 29 minutes ago	2 Commits
	it is still the readme.txt	all files	29 minutes ago
	readme without changes.txt	all files	29 minutes ago
	readme.txt	all files	29 minutes ago

README

Hello, world  
and everyone else

test\_py / readme.txt in master

Edit Preview

```

1 Hello, world
2 and everyone else
3 it is really exciting
4

```

**Commit changes**

Commit message

Extended description

Commit directly to the master branch  
 Create a new branch for this commit and start a pull request [Learn more about pull requests](#)

**Cancel** **Commit changes**

```
● PS C:\Users\user\Desktop\python\Лабораторные работы> git diff HEAD
diff --git a/readme.txt b/readme.txt
index dc32be0..74bf8ba 100644
--- a/readme.txt
+++ b/readme.txt
@@ -1,3 +1,4 @@
-Hello, world
-and everyone else
-it is really exciting
+Hello, world ))
+and everyone else /
+it is really exciting !!
+
```

```
● PS C:\Users\user\Desktop\python\Лабораторные работы> git status
On branch master
Your branch is up to date with 'origin/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.txt

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

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

```

● PS C:\Users\user\Desktop\python\Лабораторные работы> git add 'folder/for test.txt'
● PS C:\Users\user\Desktop\python\Лабораторные работы> cd "C:\Users\user\Desktop\python\Лабораторные работы"
● PS C:\Users\user\Desktop\python\Лабораторные работы> git status
On branch master
Your branch is up to date with 'origin/master'.

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

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.txt

● PS C:\Users\user\Desktop\python\Лабораторные работы> git diff --staged
diff --git a/folder/for test.txt b/folder/for test.txt
new file mode 100644
index 0000000..e69de29
● PS C:\Users\user\Desktop\python\Лабораторные работы> git reset folder/for test.txt
fatal: ambiguous argument 'folder/for': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'
● PS C:\Users\user\Desktop\python\Лабораторные работы> git reset 'folder/for test.txt'
Unstaged changes after reset:
 M      readme.txt
● PS C:\Users\user\Desktop\python\Лабораторные работы> git diff --staged
● PS C:\Users\user\Desktop\python\Лабораторные работы> git diff
diff --git a/readme.txt b/readme.txt
index dc32be0..74bf8ba 100644
--- a/readme.txt
+++ b/readme.txt
@@ -1,3 +1,4 @@
-Hello, world
-and everyone else
-it is really exciting
+Hello, world ))
+and everyone else /
+it is really exciting !!
+

```

## 6. Отмена и восстановление

Отмена индексации `git reset файл.txt` - Файл убран из индекса

Отмена изменений `git checkout -- файл.txt` - Восстановлена версия из репозитория

Удаление из индекса `git rm --cached` - Файл удален из отслеживания

- PS C:\Users\user\Desktop\python\Лабораторные работы> **git checkout -- .\readme.txt**
- PS C:\Users\user\Desktop\python\Лабораторные работы> **git status**  
On branch master  
Your branch is up to date with 'origin/master'.  
  
Untracked files:  
(use "git add <file>..." to include in what will be committed)  
  **folder/**  
  
nothing added to commit but untracked files present (use "git add" to track)
- PS C:\Users\user\Desktop\python\Лабораторные работы> **cat .\readme.txt**  
Hello, world  
and everyone else  
it is really exciting

- PS C:\Users\user\Desktop\python\Лабораторные работы> **git checkout clean**  
Switched to branch 'clean'
- PS C:\Users\user\Desktop\python\Лабораторные работы> **git branch**  
\* **clean**  
  master
- PS C:\Users\user\Desktop\python\Лабораторные работы> **ls**  
  
Directory: C:\Users\user\Desktop\python\Лабораторные работы

Mode	LastWriteTime	Length	Name
----	-----	-----	-----
d---	15.12.2025	22:35	<b>folder</b>
-a---	15.12.2025	21:53	33 it is still the readme.txt
-a---	15.12.2025	21:53	33 readme without changes.txt
-a---	15.12.2025	22:42	56 readme.txt

- PS C:\Users\user\Desktop\python\Лабораторные работы> **rm -r .\folder\**
- PS C:\Users\user\Desktop\python\Лабораторные работы> **git commit -m 'deleted folder and files'**  
On branch clean  
nothing to commit, working tree clean
- PS C:\Users\user\Desktop\python\Лабораторные работы> **git rm 'it is still readme'**  
fatal: pathspec 'it is still readme' did not match any files
- PS C:\Users\user\Desktop\python\Лабораторные работы> **git rm "C:\Users\user\Desktop\python\Лабораторные работы\it is still the readme.txt"**  
rm 'it is still the readme.txt'
- PS C:\Users\user\Desktop\python\Лабораторные работы> **git rm "C:\Users\user\Desktop\python\Лабораторные работы\readme without changes.txt"**  
rm 'readme without changes.txt'
- PS C:\Users\user\Desktop\python\Лабораторные работы> **git status**  
On branch clean  
Changes to be committed:  
(use "git restore --staged <file>..." to unstage)  
  **deleted: it is still the readme.txt**  
  **deleted: readme without changes.txt**
- PS C:\Users\user\Desktop\python\Лабораторные работы> **git commit -m 'deleted folder and files'**  
[clean 7737f6f] deleted folder and files  
2 files changed, 4 deletions(-)  
  delete mode 100644 it is still the readme.txt  
  delete mode 100644 readme without changes.txt
- PS C:\Users\user\Desktop\python\Лабораторные работы> **git status**  
On branch clean  
nothing to commit, working tree clean

```
● PS C:\Users\user\Desktop\python\Лабораторные работы> git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
● PS C:\Users\user\Desktop\python\Лабораторные работы> git merge clean
Updating e184c67..7737f6f
Fast-forward
  it is still the readme.txt | 2 --
  readme without changes.txt | 2 --
  2 files changed, 4 deletions(-)
  delete mode 100644 it is still the readme.txt
  delete mode 100644 readme without changes.txt
● PS C:\Users\user\Desktop\python\Лабораторные работы> git branch -d clean
Deleted branch clean (was 7737f6f).
● PS C:\Users\user\Desktop\python\Лабораторные работы> git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (1/1), done.
Writing objects: 100% (2/2), 241 bytes | 241.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/a14759226-glitch/test_py.git
  e184c67..7737f6f  master -> master
```

Освоенные концепции:

1. Git workflow (add → commit → push)
2. Работа с локальным репозиторием
3. Подключение к GitHub
4. Работа с историей и сравнениями
5. Основные операции отмены
6. Работа с ветками

**GitHub Desktop** — приложение от GitHub для работы с Git. Предназначено для упрощения работы с Git, особенно для начинающих пользователей.

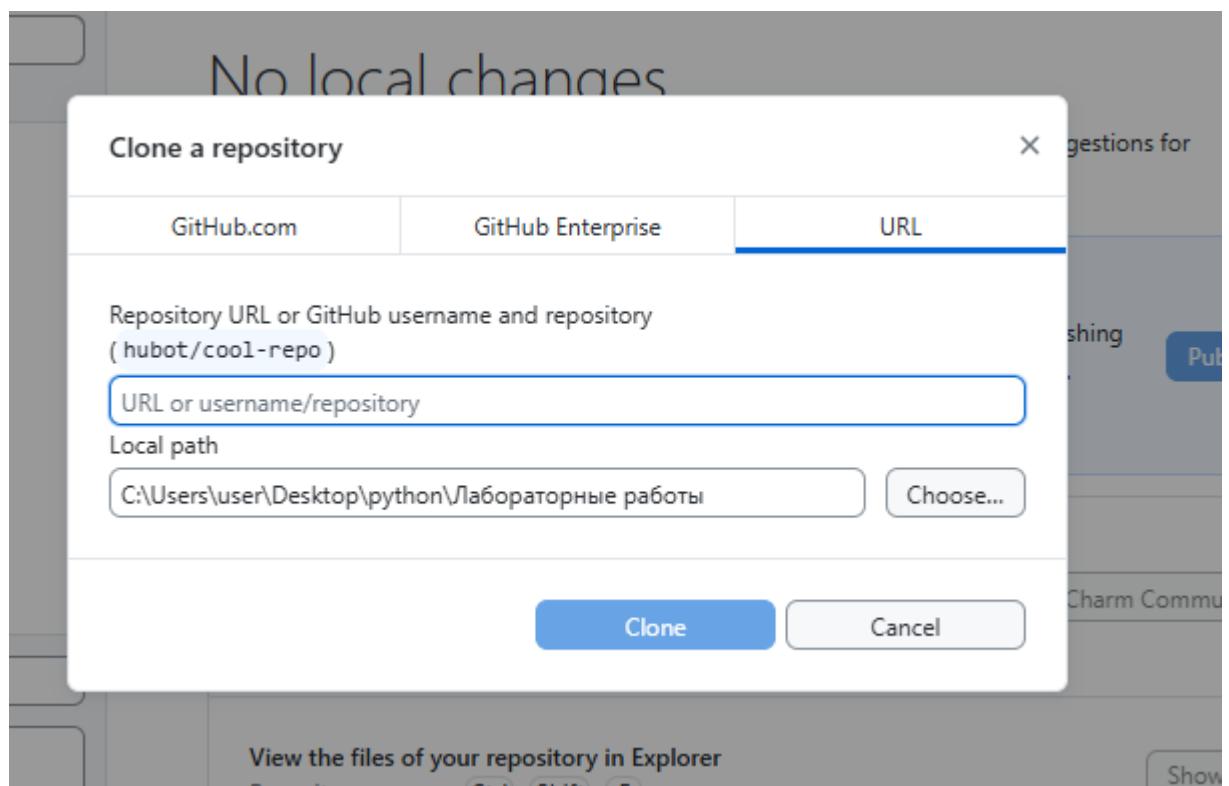
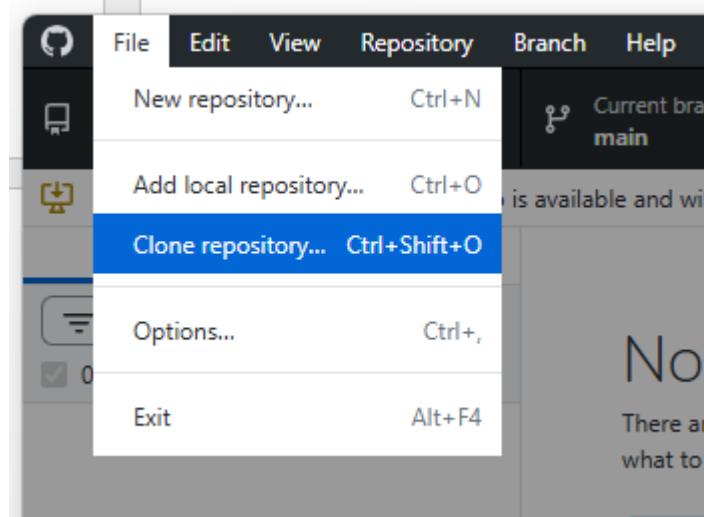
## *1. Клонирование репозитория*

git clone <https://github.com/user/repo.git>

В GitHub Desktop

1. Меню File → Clone Repository
2. Вкладка GitHub (или URL)
3. Выбор репозитория из списка
4. Выбор локальной папки
5. Кнопка "Clone"

Преимущество: Визуальный интерфейс показывает все репозитории из вашего аккаунта GitHub.



## 2. Создание нового репозитория

git init

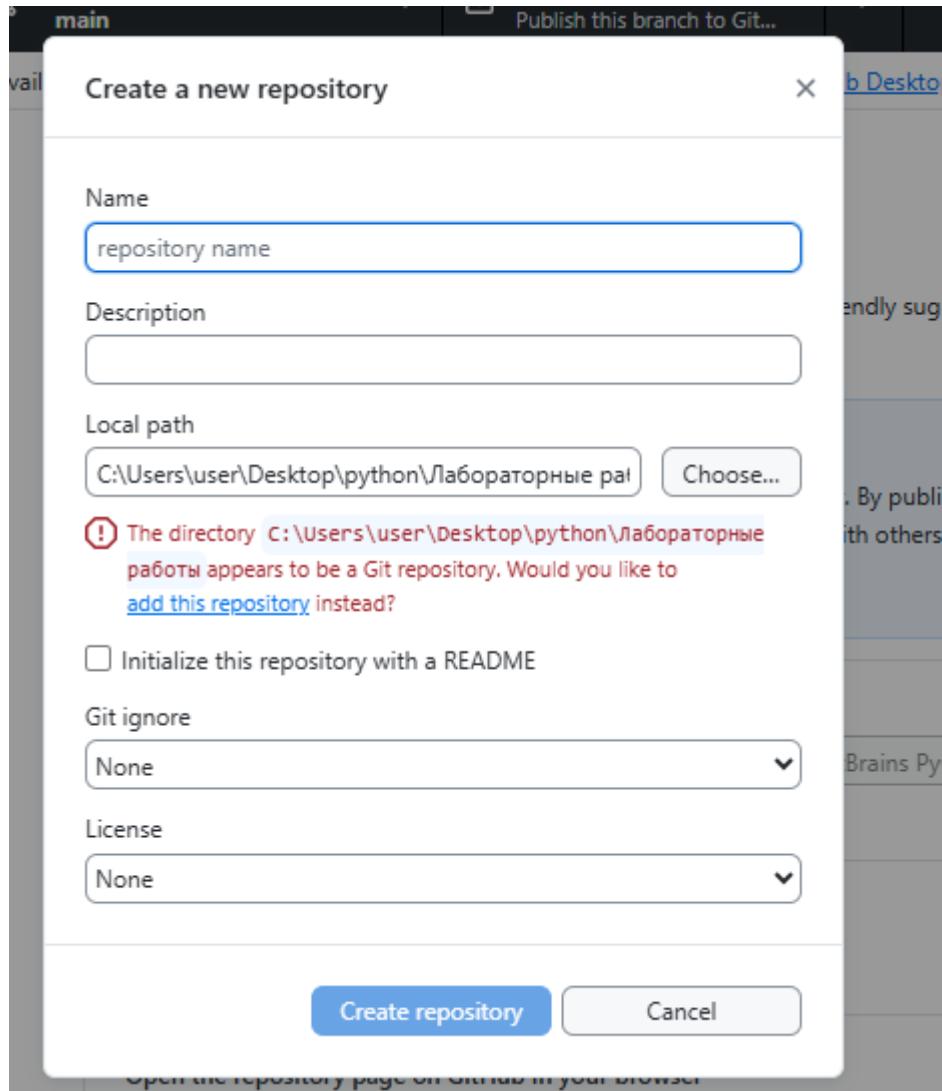
File → New Repository

Заполнение полей:

- Name: название репозитория
- Description: описание
- Local Path: путь на компьютере

- Initialize with README: создание README.md
- Git ignore: выбор шаблона
- License: выбор лицензии

Кнопка "Create Repository"



### 3. Добавление файлов

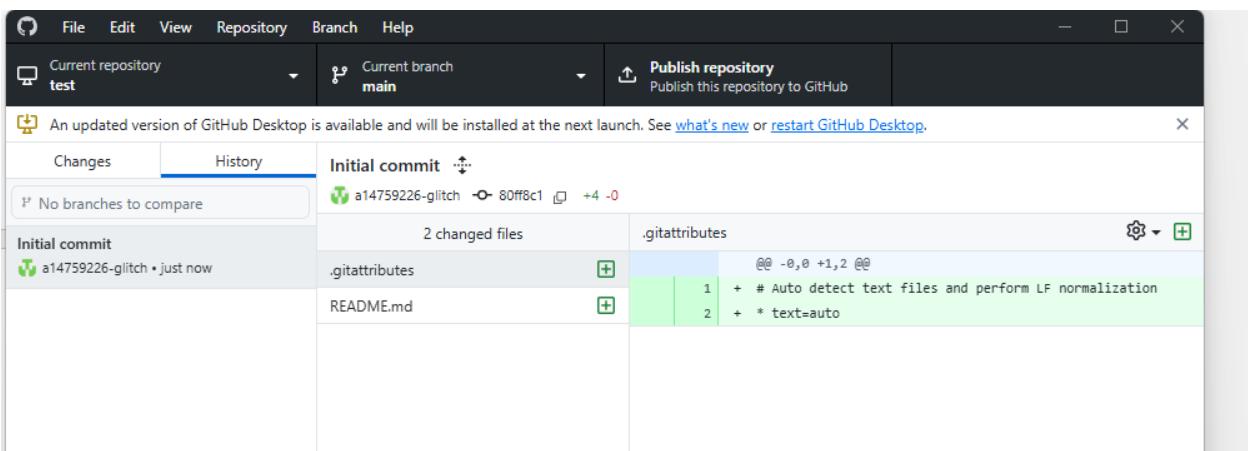
```
git add file.txt  
git add .
```

В левой панели отображаются изменения

Файлы разделены на:

- Changes (измененные файлы)
- Untracked files (новые файлы)

Визуализация: Показывает diff для каждого файла прямо в интерфейсе.



#### 4. Создание коммита

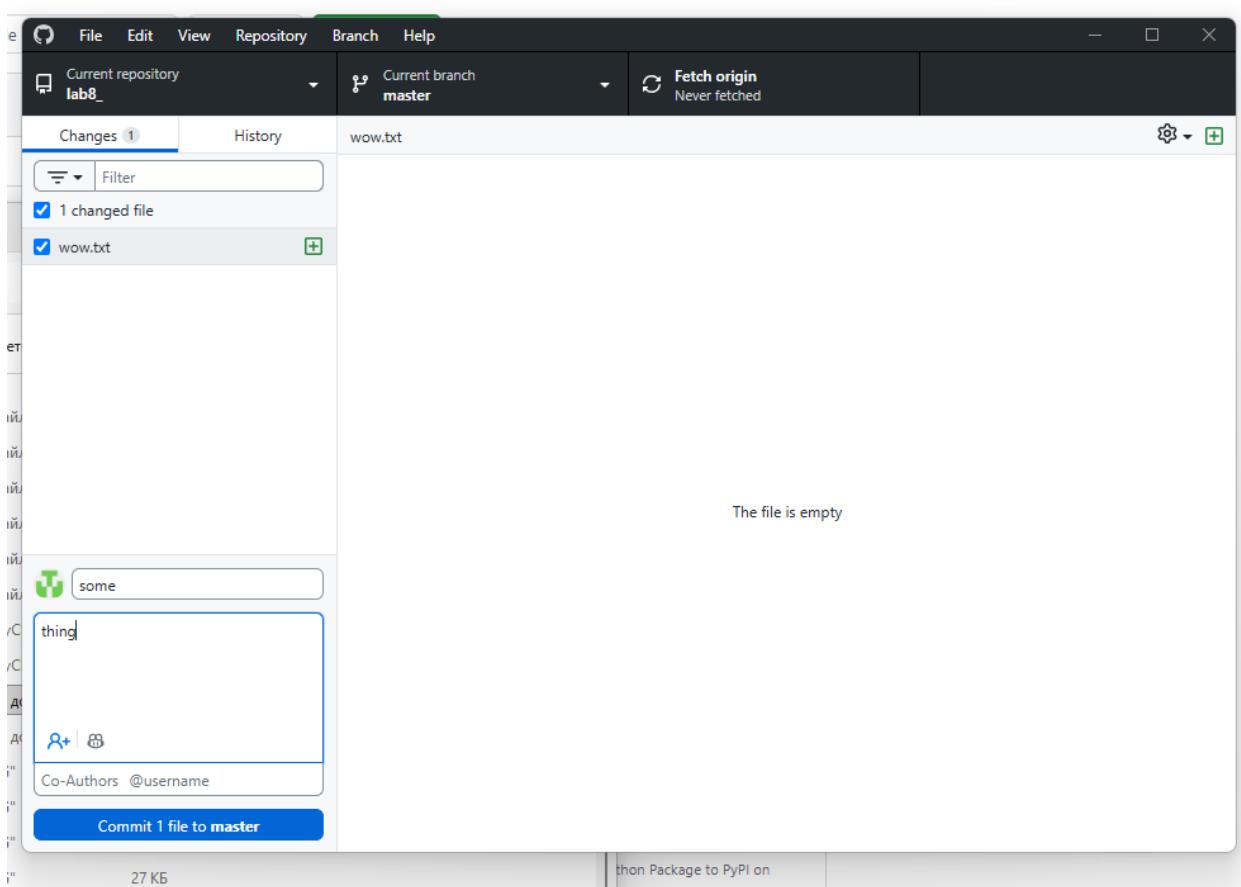
git commit -m "Сообщение коммита"

В нижней левой части окна:

- Поле "Summary"
- Поле "Description"

Кнопка "Commit to main"

Можно выбрать "Add Co-Authors"

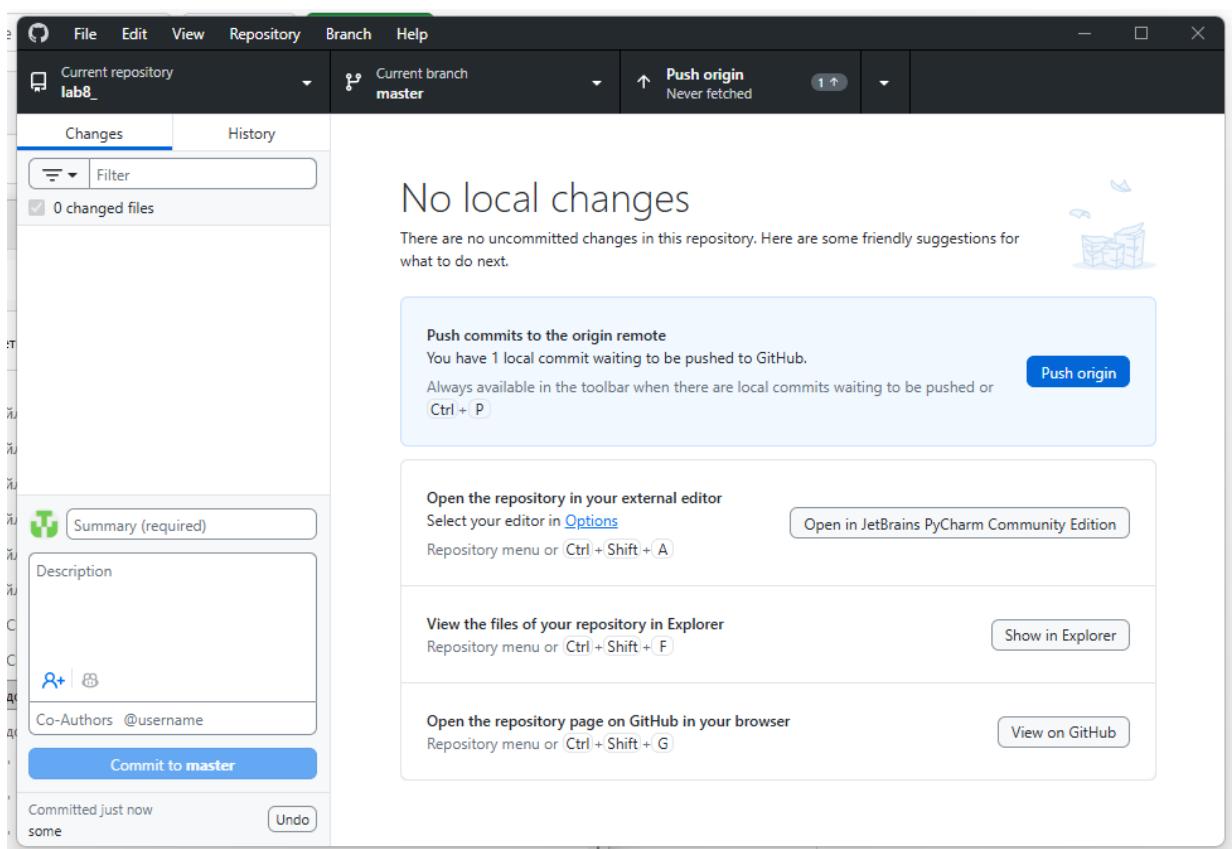


## **5. Отправка на сервер**

`git push origin main`

После коммита появляется кнопка "Push origin"  
Отображает количество неотправленных коммитов  
Прогресс отправки в реальном времени

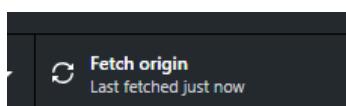
Визуализация: Показывает историю коммитов в виде графа.



## **6. Получение изменений**

`git pull origin main`

Кнопка "Fetch origin" проверяет изменения  
Если есть изменения, появляется кнопка "Pull origin"  
Можно настроить автоматический fetch



## *Работа с ветками*

### *1. Создание ветки*

```
git branch feature-branch  
git checkout feature-branch  
# или  
git checkout -b feature-branch
```

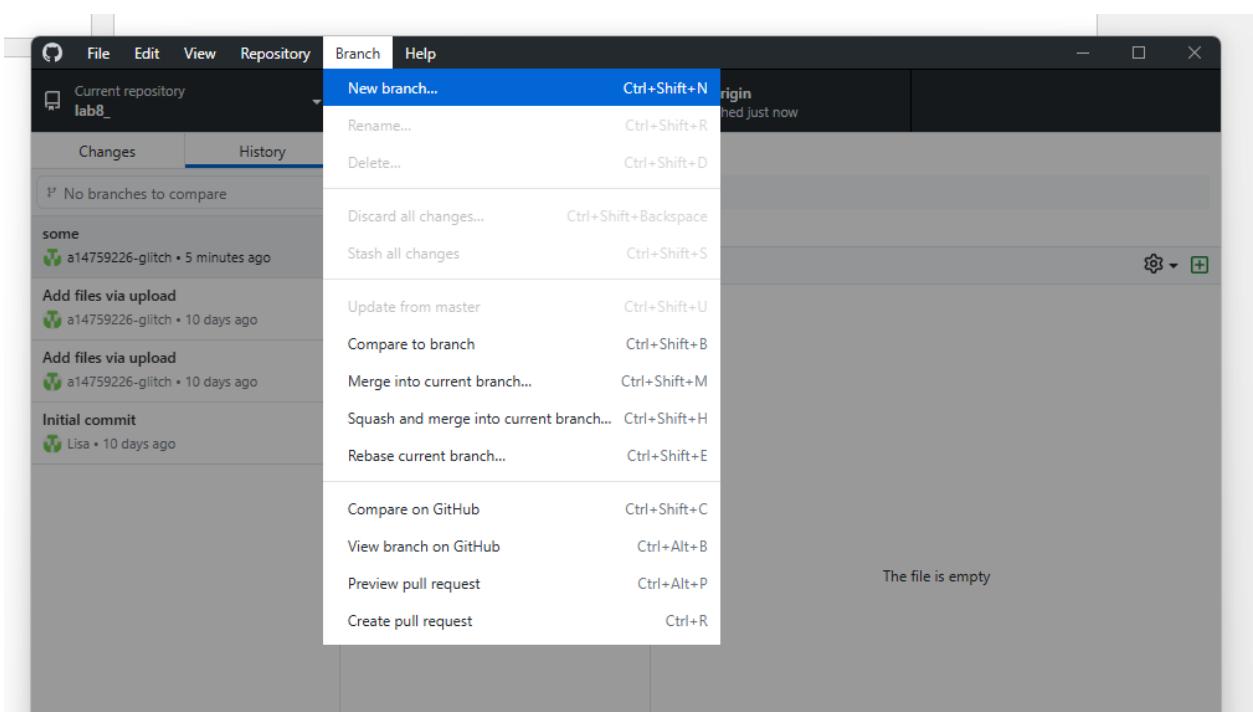
Текущая ветка отображается вверху окна

Нажатие → "New branch"

Ввод имени ветки

Выбор, от какой ветки создать

Кнопка "Create branch"



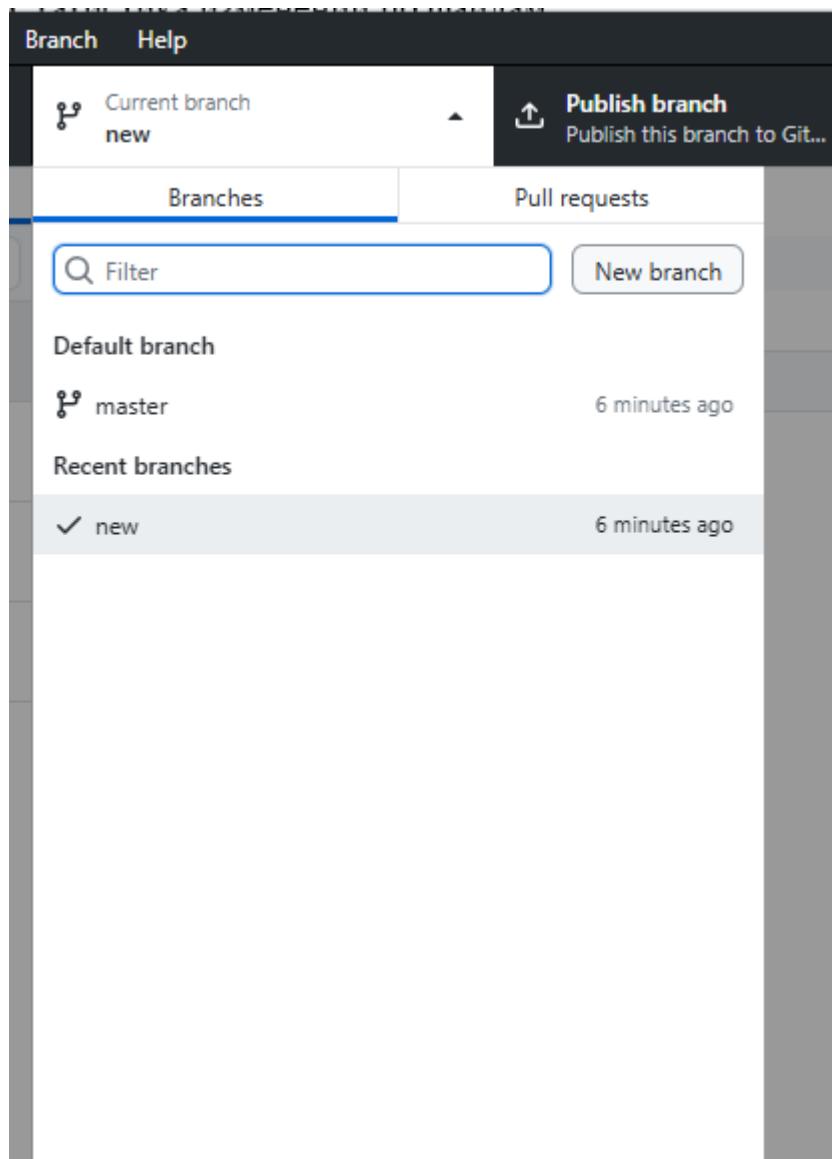
### *2. Переключение между ветками*

```
git checkout main
```

Клик по названию текущей ветки вверху

Выбор нужной ветки из списка

Автоматический stash изменений (можно отключить)



### 3. Слияние веток

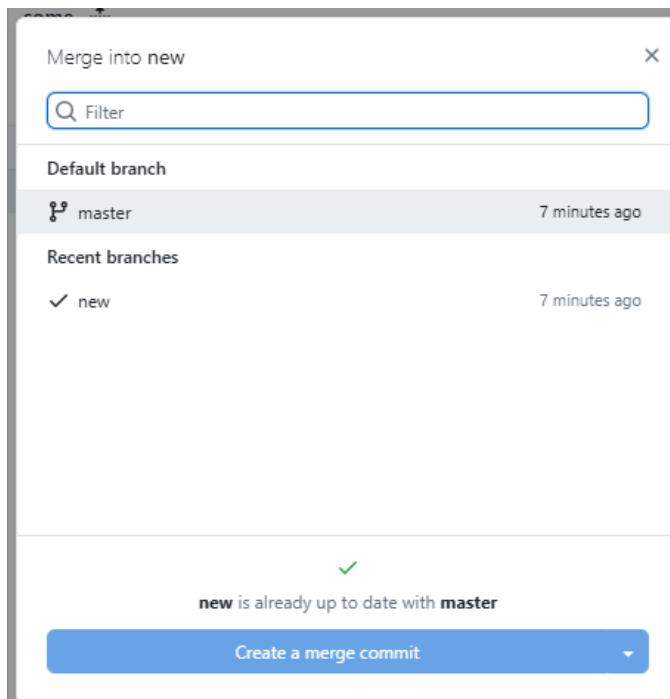
git merge feature-branch

Переключиться на целевую ветку

Branch → Merge into current branch

Выбор ветки для слияния

Кнопка "Merge feature-branch into main"



### Уникальные особенности GitHub Desktop

- Прямая авторизация
- Быстрое создание репозиториев на GitHub
- Синхронизация списка репозиториев
  
- Граф истории с ветвлением
- Статистика изменений по файлам
- Предпросмотр перед коммитом
  
- Автоматический stash при переключении веток
- Открытие в редакторе/терминале

**Итоговая оценка:** GitHub Desktop значительно упрощает начало работы с Git, но для сложных операций и полного контроля необходимы консольные команды. Идеально подходит для образовательных целей и командной работы.