

# Отчет по практической работе №1

## Основы Git и GitHub

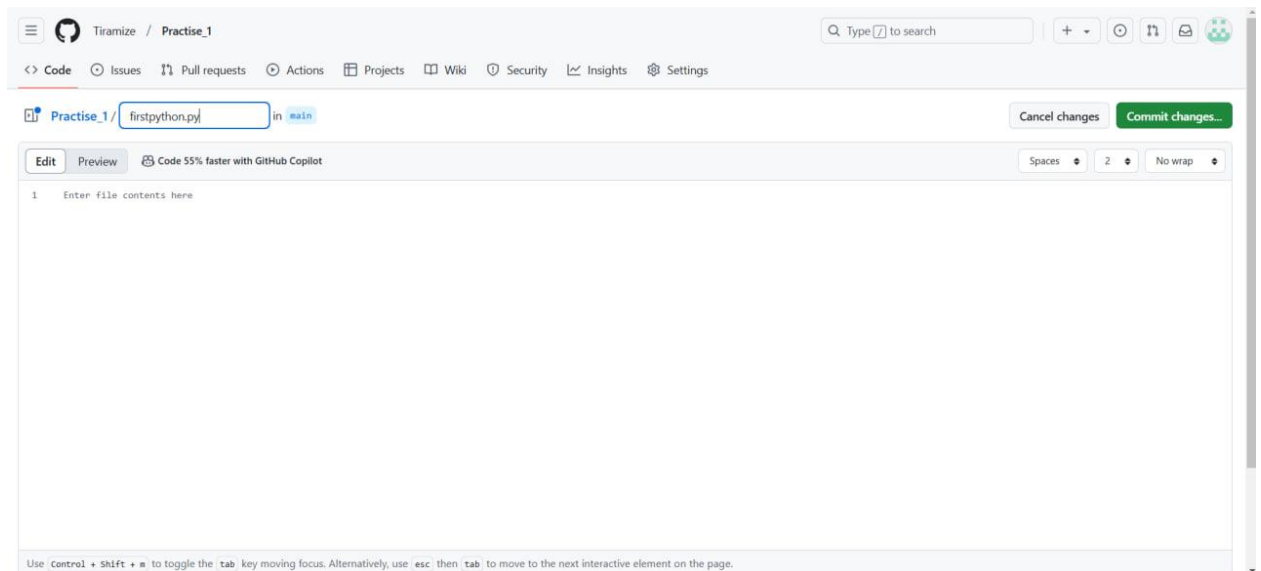
### Цель работы

Изучение основных принципов работы с системами контроля версий Git и GitHub, включая создание репозиториев, управление ветками, коммитами, слиянием изменений и совместной разработкой.

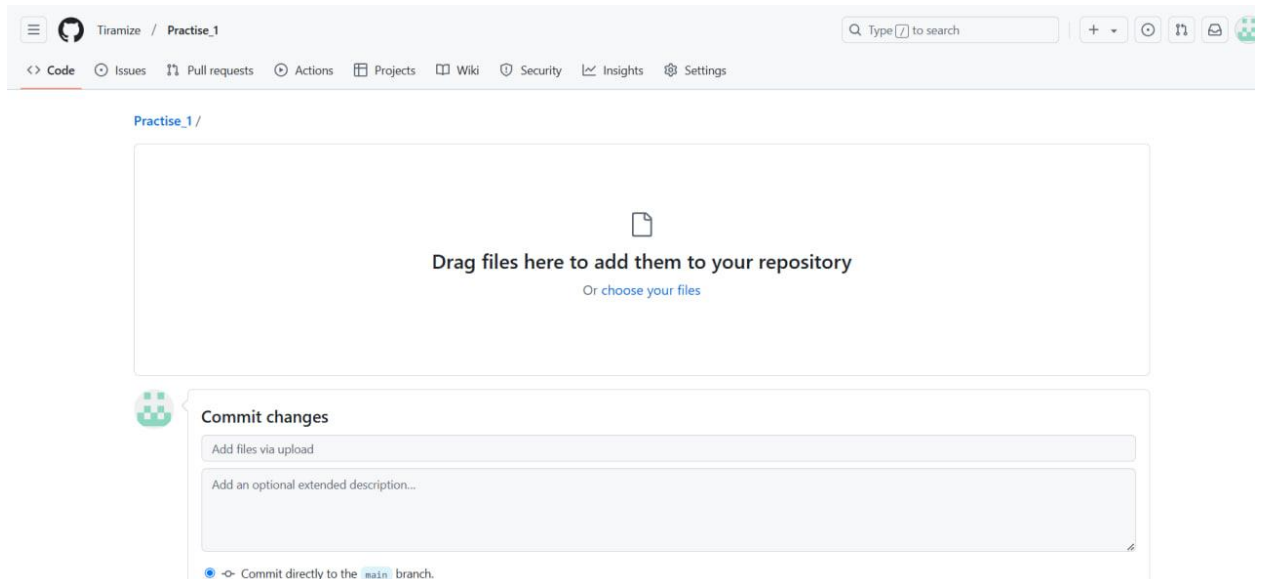
### Выполненные задания:

#### Задание 1:

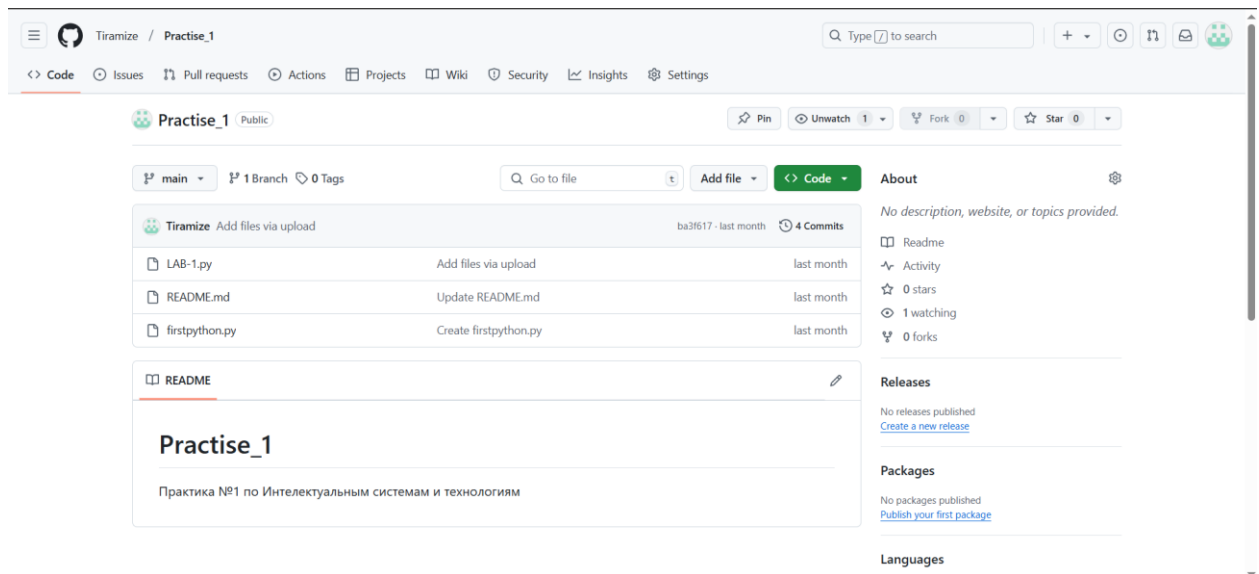
Регистрируемся в GitHub, создаем репозиторий в GitHub



Корневая папка вашего репозитория указана по умолчанию и содержит только один файл README.md.

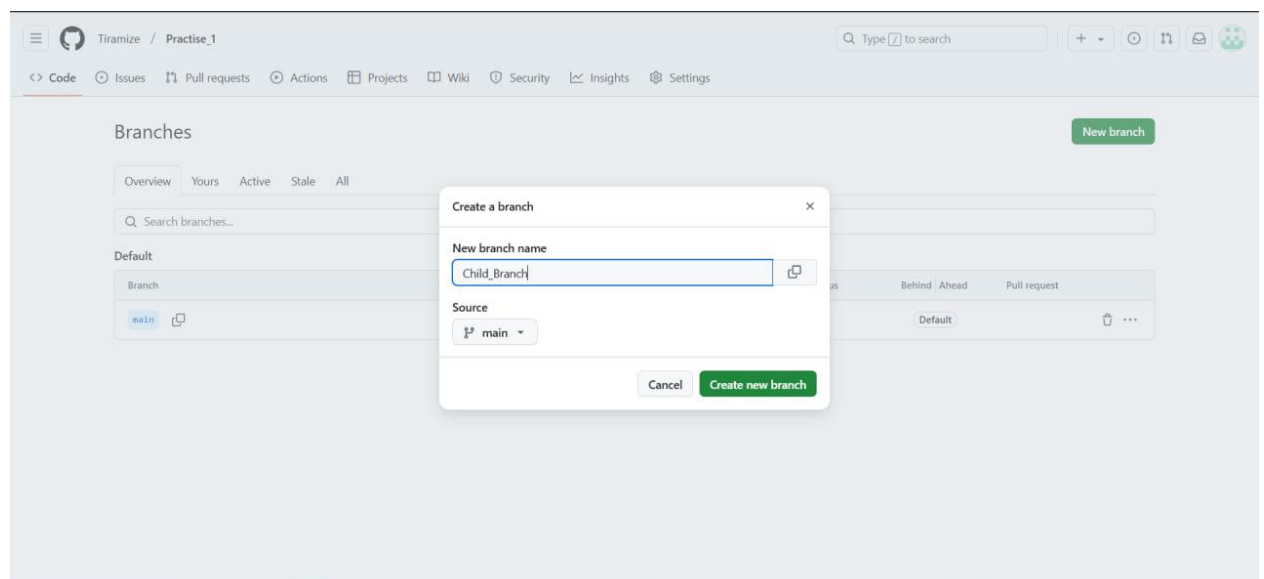


В репозитории теперь отображаются загруженные файлы.



Создаем branch.

- Перейду на главную страницу вашего репозитория. В верхней части списка файлов найдите раскрывающееся меню «branch».



В вашем репозитории теперь есть две ветки: master и Child\_Branch.

## Branches

[New branch](#)

Overview Yours Active Stale All

Q Search branches...

### Default

Branch	Updated	Check status	Behind	Ahead	Pull request
main	last month			Default	...

### Your branches

Branch	Updated	Check status	Behind	Ahead	Pull request
Child_Branch	now		0	0	...

Создал файл в репозитории.

Введите имя и расширение файла `testchild.py` и добавил следующие строки в тело нового файла:

Tiramize Create testchild.py ...

Code

Blame

1 lines (1 loc) · 30 Bytes

Code 55% faster with GitHub Copilot

```
1 print ("inside Child branch")
```

Операция «Merge» на базе «pull request»

Чтобы объединить ветки по запросу `pull request` в проекте, я выполнил следующие действия:

### Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#) or [learn more about diff comparisons](#).

base: main@1day ← compare: main

1 commit

1 file changed

1 contributor

Commits on Mar 11, 2025

Create testchild.py ...

Tiramize authored 12 minutes ago

Verified

a825359

<>

Showing 1 changed file with 1 addition and 0 deletions.

Split Unified

1 testchild.py

```
... @@ -0,0 +1 @@  
1 + print ("inside Child branch")
```

## Задание 2:

Создаем новую директорию для локального репозитория:

1. Создаем каталог myrepo.

```
PS C:\Users\User> mkdir myrepo
```

2. Перейдем в каталог myrepo.

```
PS C:\Users\User> cd myrepo  
PS C:\Users\User\myrepo> █
```

3. В этом каталоге myrepo создаем локальный репозиторий git.

```
PS C:\Users\User\myrepo> git init  
Initialized empty Git repository in C:/Users/User/myrepo/.git/  
PS C:\Users\User\myrepo> █
```

4. Создадим пустой файл

```
User@DESKTOP-QQSBC9L MINGW64 ~/mypero/newbranch (master)  
$ touch newbranchfile  
  
User@DESKTOP-QQSBC9L MINGW64 ~/mypero/newbranch (master)  
$ █
```

5. Добавляем этот файл в репозиторий и зафиксируем изменения

```
User@DESKTOP-QQSBC9L MINGW64 ~/mypero/newbranch (master)  
$ git add newbranchfile  
  
User@DESKTOP-QQSBC9L MINGW64 ~/mypero/newbranch (master)  
$ git commit -m "added newbranchfile"  
[master d79519d] added newbranchfile  
1 file changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 newbranch/newbranchfile  
  
User@DESKTOP-QQSBC9L MINGW64 ~/mypero/newbranch (master)  
$ █
```

6. Создадим еще один файл с именем readme.md и проверим статус

```
User@DESKTOP-QQSBC9L MINGW64 ~/mypero (master)  
$ git status  
On branch master  
Changes to be committed:  
  (use "git restore --staged <file>..." to unstage)  
    modified:   newfile  
    new file:   readme.md
```

7. Внесем еще одно изменение в наш текущий активный my1stbranch

```
User@DESKTOP-QQSBC9L MINGW64 ~/mypero (master)  
$ git merge my1stbranch  
Already up to date.
```

8. Вывод журнала показывает, что недавно добавленный goodfile был зафиксирован в my1stbranch

```

User@DESKTOP-QQSBC9L MINGW64 ~/mypero (master)
$ touch goodfile

User@DESKTOP-QQSBC9L MINGW64 ~/mypero (master)
$ git add goodfile

User@DESKTOP-QQSBC9L MINGW64 ~/mypero (master)
$ git commit -m "added goodfile"
[master efeae44] added goodfile
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 goodfile

User@DESKTOP-QQSBC9L MINGW64 ~/mypero (master)
$ git log
commit efeae4478ec8a6fad43f6853eba5746b326da167 (HEAD -> master)
Author: papapa <papapa@papa.com>
Date:   Wed Mar 5 14:15:32 2025 +0300

    added goodfile

commit b3f49cea876c41f0e778c0c04c7d2c1620353024
Author: papapa <papapa@papa.com>
Date:   Wed Mar 5 14:13:13 2025 +0300

    added readme.md modified newfile

commit 3039b621e8224832706fad45aad08ffd8635e85b (my1stbranch)
Author: papapa <papapa@papa.com>
Date:   Wed Mar 5 14:06:06 2025 +0300

    added newfile

User@DESKTOP-QQSBC9L MINGW64 ~/mypero (master)

```

9. Теперь, когда изменения были объединены в главную ветку, my1stbranch можно удалить с помощью следующей команды git branch с опцией -d

```

User@DESKTOP-QQSBC9L MINGW64 ~/mypero (master)
$ git branch -d my1stbranch
Deleted branch my1stbranch (was 3039b62).

User@DESKTOP-QQSBC9L MINGW64 ~/mypero (master)

```

### Дополнительное задание:

- Создал новую ветку под названием newbranch, сделал новую ветку активной

```

User@DESKTOP-QQSBC9L MINGW64 ~/mypero (master)
$ mkdir newbranch

User@DESKTOP-QQSBC9L MINGW64 ~/mypero (master)
$ cd newbranch

User@DESKTOP-QQSBC9L MINGW64 ~/mypero/newbranch (master)

```



- Создал пустой файл с именем newbranchfile.

```
User@DESKTOP-QQSBC9L MINGW64 ~/mypero/newbranch (master)
$ touch newbranchfile

User@DESKTOP-QQSBC9L MINGW64 ~/mypero/newbranch (master)
$
```

- Добавил вновь созданный файл в свою ветку, зафиксировал изменения в новой ветке.

```
User@DESKTOP-QQSBC9L MINGW64 ~/mypero/newbranch (master)
$ git add newbranchfile

User@DESKTOP-QQSBC9L MINGW64 ~/mypero/newbranch (master)
$ git commit -m "added newbranchfile"
[master d79519d] added newbranchfile
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 newbranch/newbranchfile

User@DESKTOP-QQSBC9L MINGW64 ~/mypero/newbranch (master)
$
```

- Отменил последние зафиксированные изменения.

```
PS C:\Users\User\mypero\newbranch> git reset
```

- Создал новый файл с именем newgoodfile.

```
User@DESKTOP-QQSBC9L MINGW64 ~/mypero/newbranch (master)
$ mkdir newgoodfile
```

- Добавил последний файл в новую ветку.

```
User@DESKTOP-QQSBC9L MINGW64 ~/mypero/newbranch (master)
$ git add newgoodfile
```

- Зафиксировал изменения.

```
User@DESKTOP-QQSBC9L MINGW64 ~/mypero/newbranch (master)
$ git commit -m "added newgoodfile"
```

- Объединил изменения в новой ветке с основной.

```
User@DESKTOP-QQSBC9L MINGW64 ~/mypero/newbranch (master)
$ git checkout master
Already on 'master'

User@DESKTOP-QQSBC9L MINGW64 ~/mypero/newbranch (master)
$ git merge my1stbranch
```

## Вывод

В ходе работы были освоены ключевые навыки работы с Git и GitHub:

- Создание репозитория и управление ветками.
- Фиксация изменений и отмена коммитов.

- Синхронизация локальных и удаленных репозиторий.
- Совместная разработка через Fork, Clone и Pull Request.