

Практическая работа №1

Основы Git и Github

Цель работы. Выполнение практической работы направлено на изучение:

1. наиболее распространенных практик в области контроля версий программного обеспечения, его использования в командной разработке ПО и DevOps;
2. концепции Git, основанной на понятиях репозитория и ветвления версий ПО;
3. порядка использования GitHub и его базовых операций.

Задание № 1.

Для начала создадим папку внутри которой будем работать

Откроем git bash внутри папки и создадим локальный репозиторий с помощью команды git init

```
Student@ws310-2M MINGW64 /z/3 курс/IS&T2024/090303b/210803177/MyGitRepo1
$ git init
Initialized empty Git repository in //IISTServer/students/3 курс/IS&T2024/090303b/210803177/MyGitRepo1/.git/
```

создадим некие пару начальных файлов, с помощью команды touch file initial

```
Student@ws310-2M MINGW64 /z/3 курс/IS&T2024/090303b/210803177/MyGitRepo1 (master)
$ touch file initial
```

Добавим его и закоммитим изменения

```
Student@ws310-2M MINGW64 /z/3 курс/IS&T2024/090303b/210803177/MyGitRepo1 (master)
$ git commit -m "first commit"
[master (root-commit) 13a3f62] first commit
  Committer: Student <Student@IIIST.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file
create mode 100644 initial
```

Теперь создадим новую ветку с помощью команды `git branch newbranch`

```
Student@ws310-2M MINGW64 /z/3 курс/IS&T2024/090303b/210803177/MyGitRepo1 (master)
$ git branch newbranch
```

перейдем в нее с помощью команды `git checkout newbranch`

```
Student@ws310-2M MINGW64 /z/3 курс/IS&T2024/090303b/210803177/MyGitRepo1 (master)
$ git checkout newbranch
Switched to branch 'newbranch'
```

создадим внутри ветки новый файл `newbranchfile`

```
Student@ws310-2M MINGW64 /z/3 курс/IS&T2024/090303b/210803177/MyGitRepo1 (newbranch)
$ touch newbranchfile
```

и добавим его, сохраним изменения

```
Student@ws310-2M MINGW64 /z/3 курс/IS&T2024/090303b/210803177/MyGitRepo1 (newbranch)
$ git add .
```

```

Student@ws310-2M MINGW64 /z/3 курс/IS&T2024/090303b/210803177/MyGitRepo1 (newbranch)
$ git commit -m "create new file"
[newbranch 474bd2f] create new file
Committer: Student <Student@IIST.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 newbranchfile

```

после чего отменим последний коммит с помощью команды `git revert HEAD --no-edit`

```

Student@ws310-2M MINGW64 /z/3 курс/IS&T2024/090303b/210803177/MyGitRepo1 (newbranch)
$ git revert HEAD --no-edit
[newbranch 04790ce] Revert "create new file"
Date: Sat Feb 17 15:31:04 2024 +0300
Committer: Student <Student@IIST.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 newbranchfile

```

создадим новый правильный файл за место прошлого

```

Student@ws310-2M MINGW64 /z/3 курс/IS&T2024/090303b/210803177/MyGitRepo1 (newbranch)
$ touch newgoodfile

```

добавим его и сохраним изменения

```

Student@ws310-2M MINGW64 /z/3 курс/IS&T2024/090303b/210803177/MyGitRepo1 (newbranch)
$ git add .

Student@ws310-2M MINGW64 /z/3 курс/IS&T2024/090303b/210803177/MyGitRepo1 (newbranch)
$ git commit -m "create new good file"
[newbranch ac6de07] create new good file
Committer: Student <Student@IIST.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 newgoodfile

```

наконец перейдем на основную ветку master

```

Student@ws310-2M MINGW64 /z/3 курс/IS&T2024/090303b/210803177/MyGitRepo1 (newbranch)
$ git checkout master
Switched to branch 'master'

```

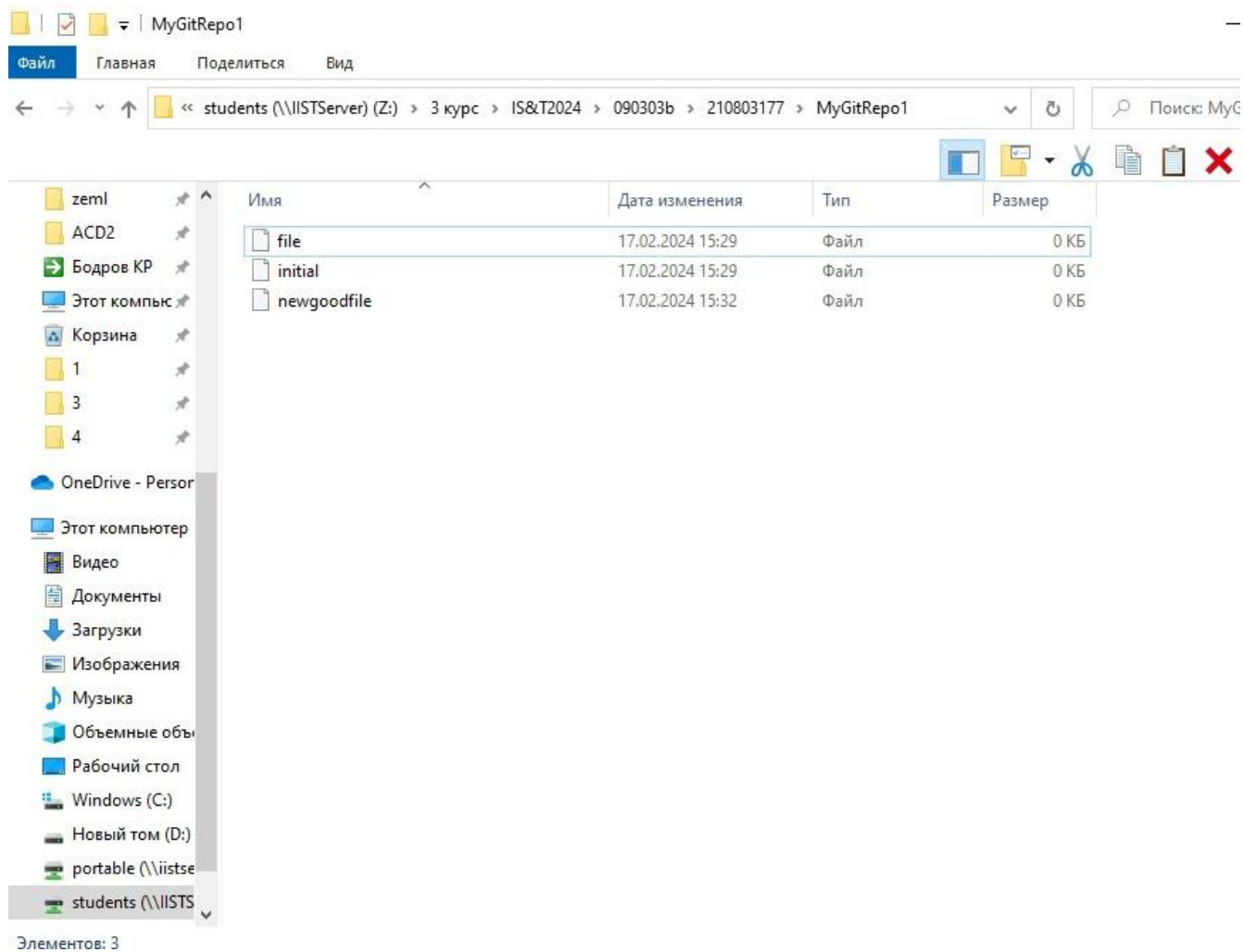
сольем две ветки в одну, master, командой git merge

```

Student@ws310-2M MINGW64 /z/3 курс/IS&T2024/090303b/210803177/MyGitRepo1 (master)
$ git merge newbranch
Updating 13a3f62..ac6de07
Fast-forward
 newgoodfile | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 newgoodfile

```

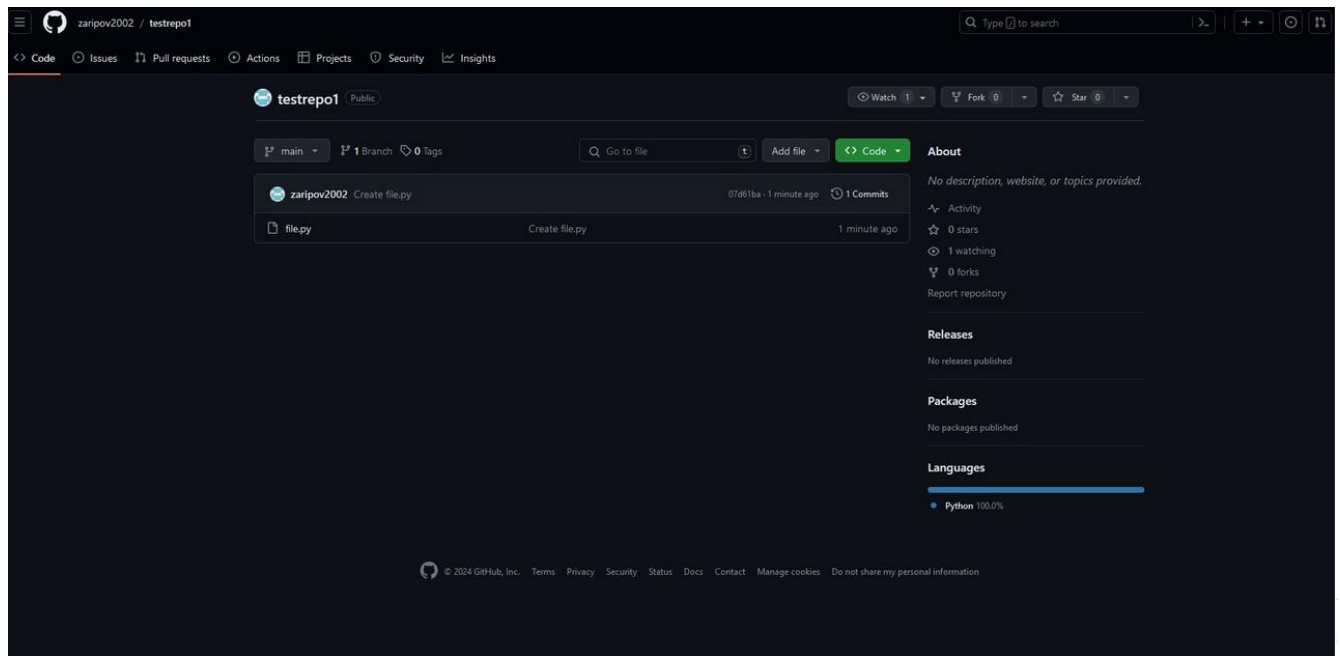
В результате всех проделанных изменений папка выглядит так как показано на рисунке



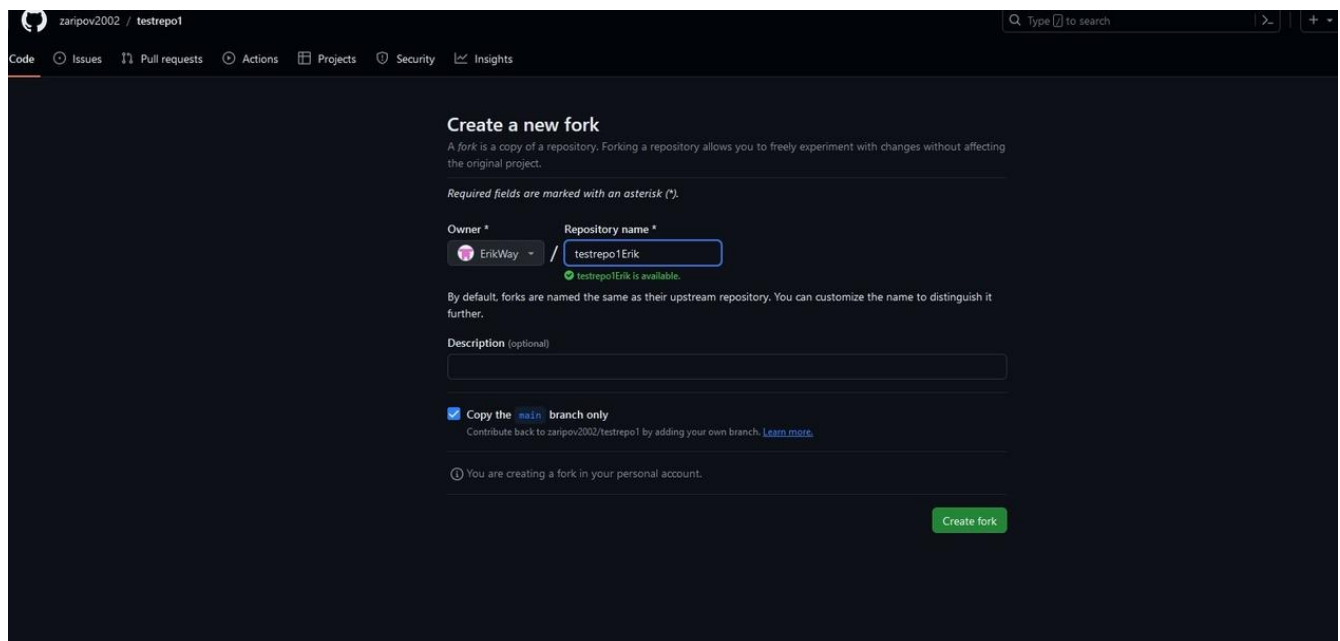
Задание № 2.

Для начала возьмем репозиторий другого студента, который доступен по ссылке <https://github.com/zaripov2002/testrepo1>

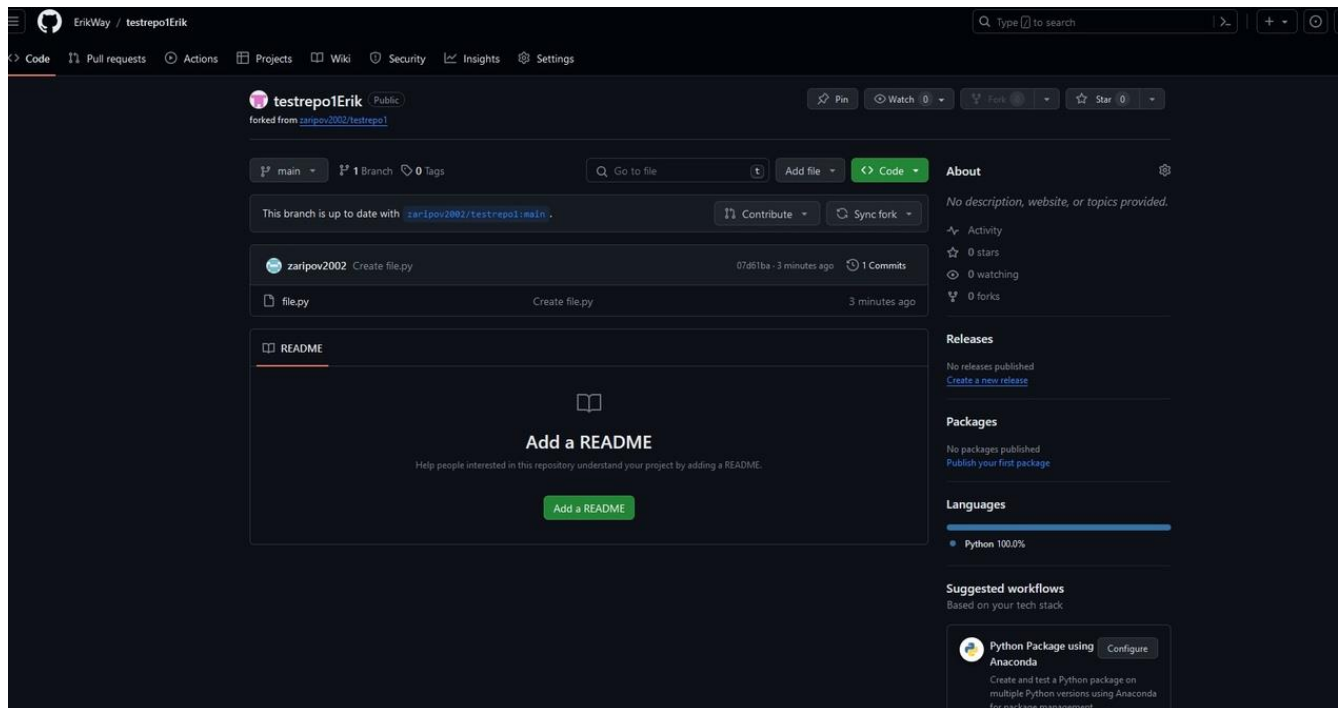
Изначально репозиторий содержит лишь один файл



Создадим форк origin проекта, нажав на кнопку Fork, после чего попадаем на меню задания названия новому репозиторию, назовем его testrepo1Erik



После чего получаем репозиторий аналогичный главному



Далее перейдем к работе в локальном репозитории, для этого создадим папку, и откроем внутри нее git bash, после чего инициализируем внутри git, прописав команду `git init`, после чего сделаем клон нашего форк-репозитория, с помощью команды `git clone https://github.com/ErikWay/testrepo1Erik` и перейдем в корневую папку прописав команду `cd testrepo1Erik`

```
erik@WIN-4FVL10UTBTL MINGW64 ~/Desktop/rolle
$ git init
Initialized empty Git repository in C:/Users/erik/Desktop/rolle/.git/

erik@WIN-4FVL10UTBTL MINGW64 ~/Desktop/rolle (main)
$ git clone https://github.com/ErikWay/testrepo1Erik
Cloning into 'testrepo1Erik'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

erik@WIN-4FVL10UTBTL MINGW64 ~/Desktop/rolle (main)
$ cd testrepo1Erik
```

Теперь создадим некий новый пустой файл `newErikFile`, добавим его и сохраним изменения, как показано на рисунке

```

erik@WIN-4FVL10UTBTL MINGW64 ~/Desktop/rolle/testrepo1Erik (main)
$ touch newErikFile

erik@WIN-4FVL10UTBTL MINGW64 ~/Desktop/rolle/testrepo1Erik (main)
$ git add .

erik@WIN-4FVL10UTBTL MINGW64 ~/Desktop/rolle/testrepo1Erik (main)
$ git commit -m "create ErikFile"
[main 6f9e7bf] create ErikFile
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 newErikFile

```

После чего, нам необходимо выгрузить изменения в гитхаб, используя команду `git push`. В ходе выполнения этого пункта, опираясь на методические рекомендации, в которых было указано использовать команду `git push origin main`, вылетала ошибка, указывающая на отсутствие заданного адреса на переменной `origin`, что можно исправить задав этот адрес командой `git remote add origin <URL>`, однако указывать `origin` и `main` в нашем случае нет необходимости, так как подключение к удаленному репозиторию уже было создано, во время клона проекта и так как ветка у нас всего одна, `main`.

Так же, в тех же методических рекомендациях было указано использовать токен, для передачи данных на гитхаб, однако, вместо токена, была произведена авторизация в гитхаб со всплывающего окна.

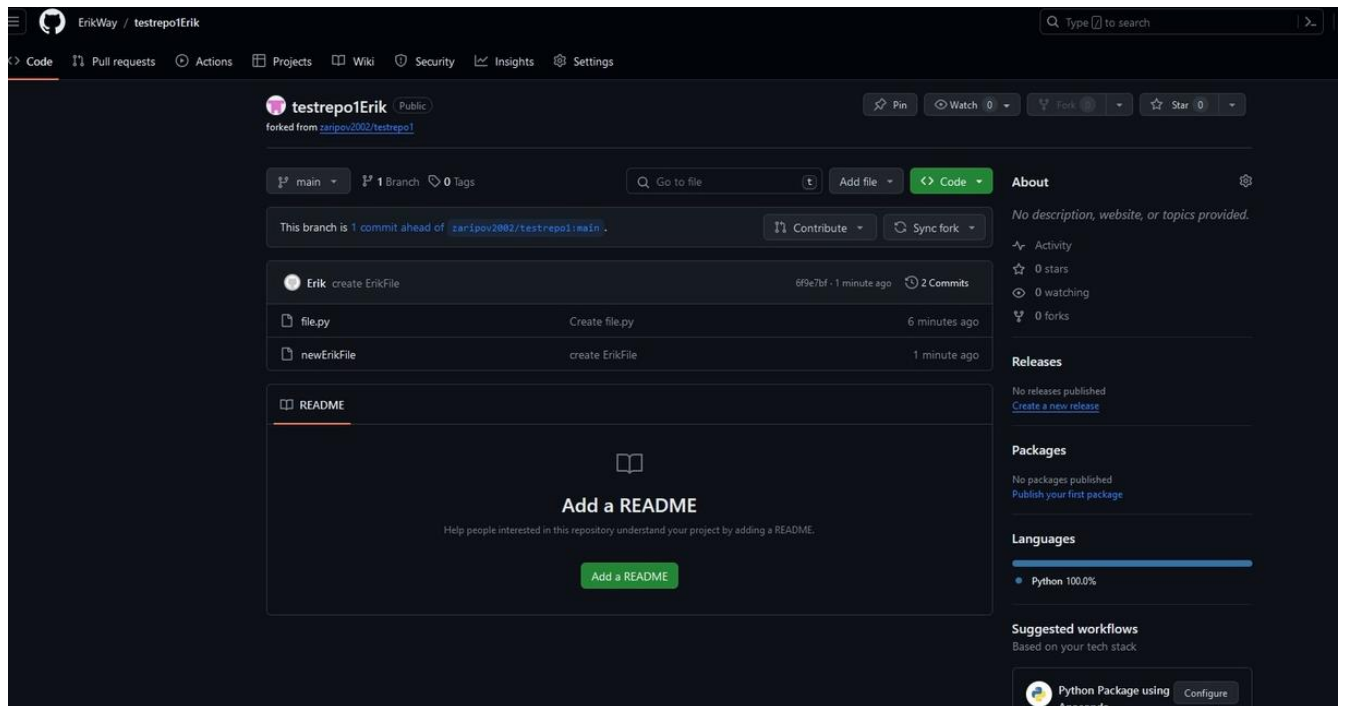
```

erik@WIN-4FVL10UTBTL MINGW64 ~/Desktop/rolle/testrepo1Erik (main)
$ git push
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), 272 bytes | 136.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ErikWay/testrepo1Erik
07d61ba..6f9e7bf main -> main

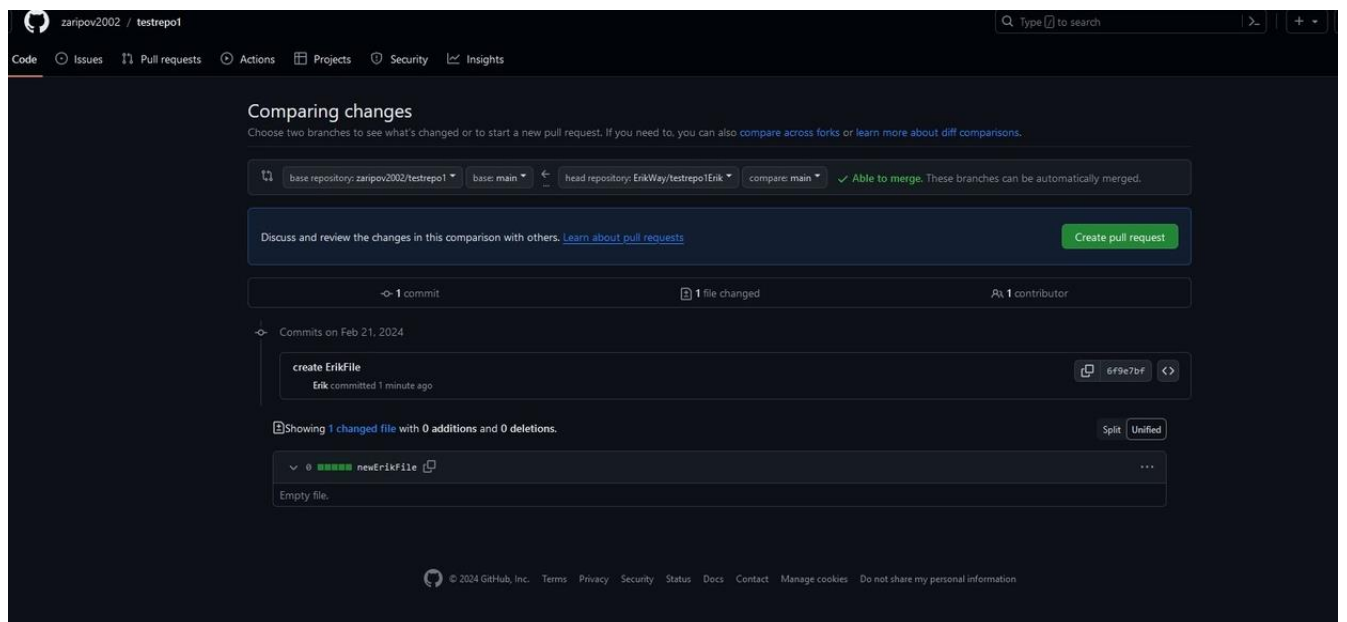
erik@WIN-4FVL10UTBTL MINGW64 ~/Desktop/rolle/testrepo1Erik (main)
$ |

```

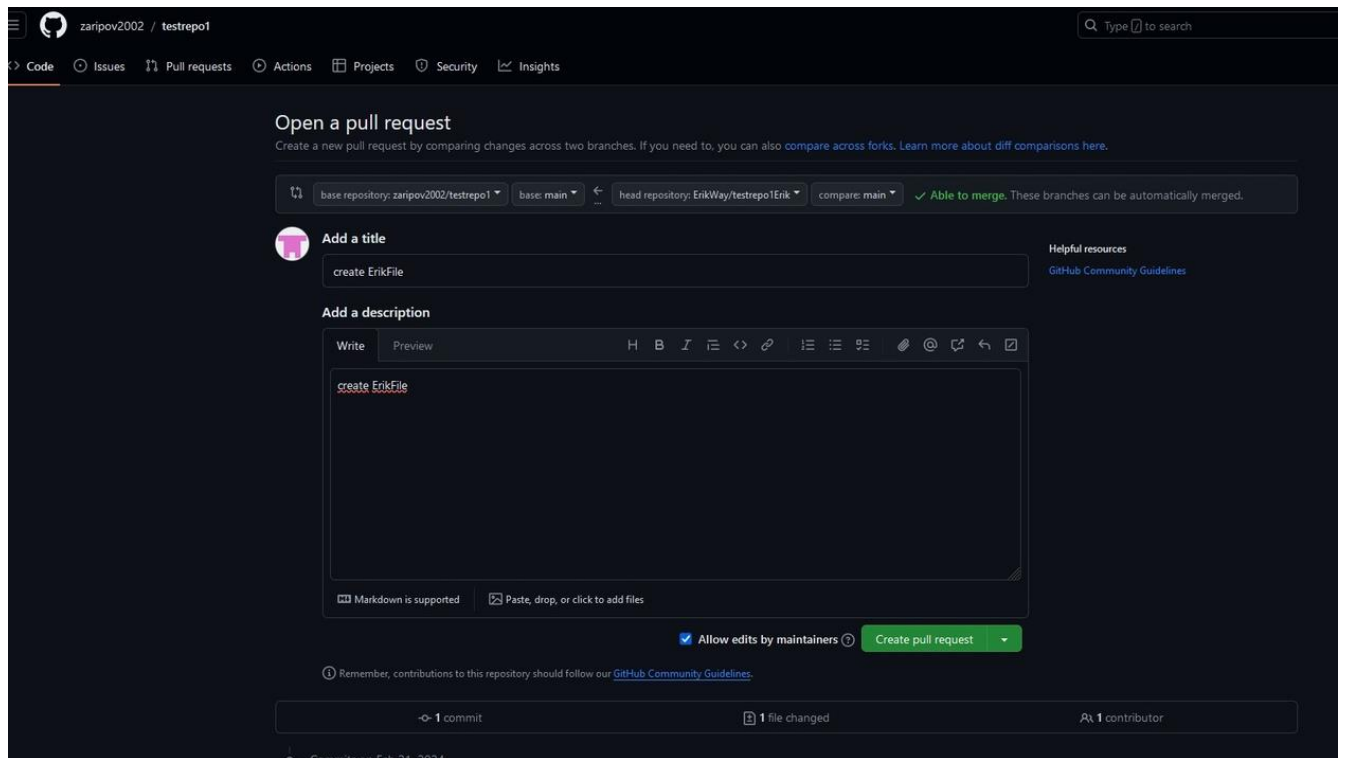
Зайдем на гитхаб, и удостоверимся, что файл был добавлен



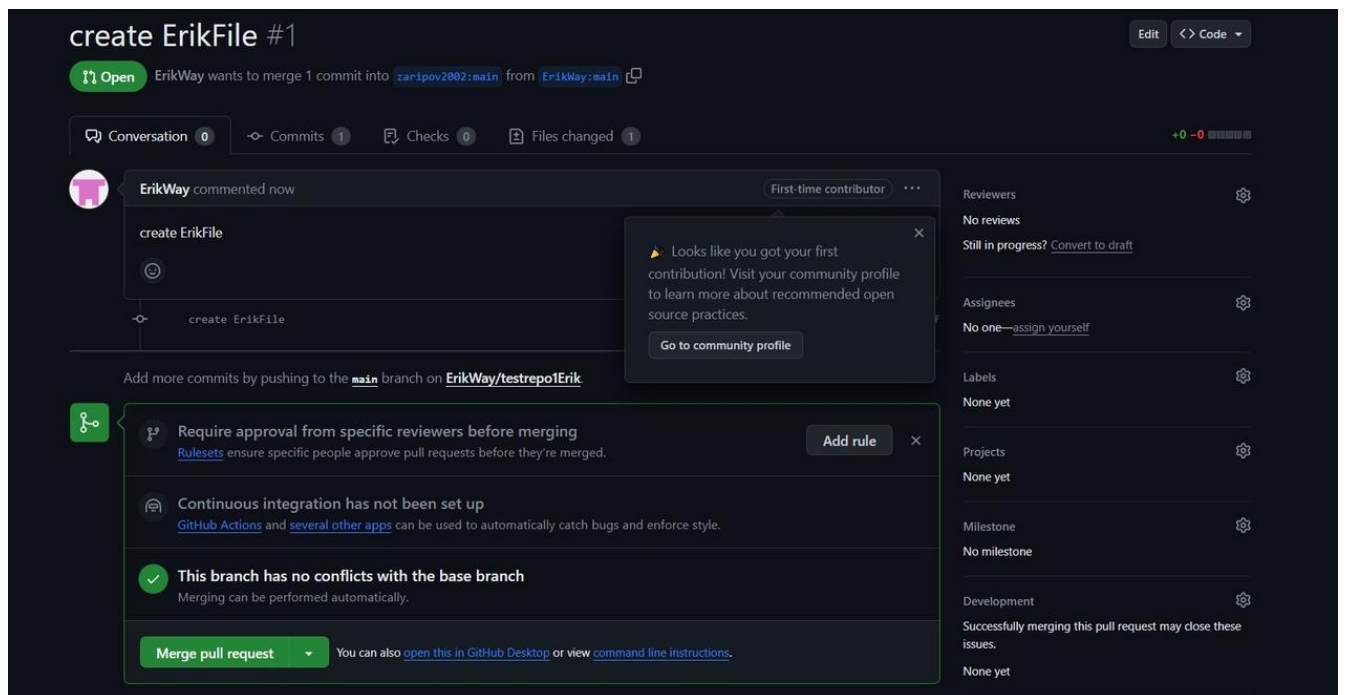
Синхронизируем изменения с origin репозиториум, нажав на кнопку sync fork



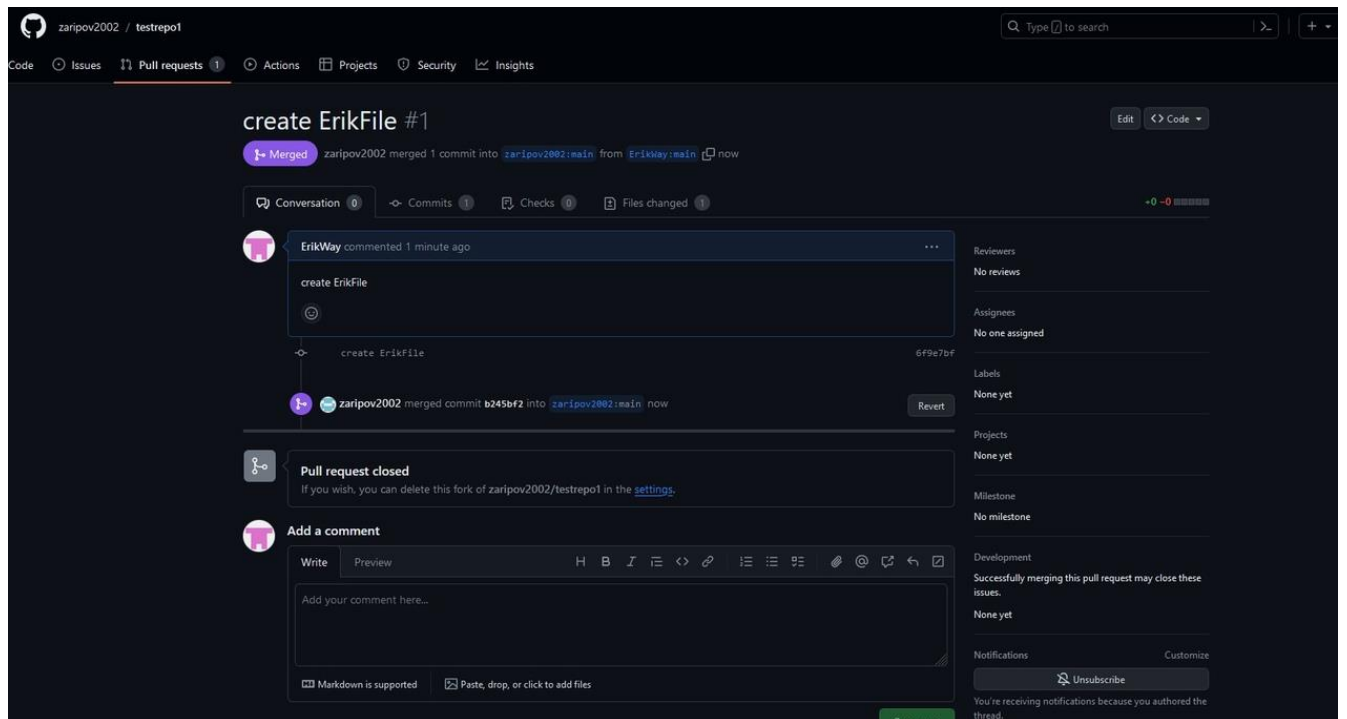
Нажмем кнопку Create pull request, создав запрос на слияние с главным репозиториум второго студента, после чего заполняем необходимые поля



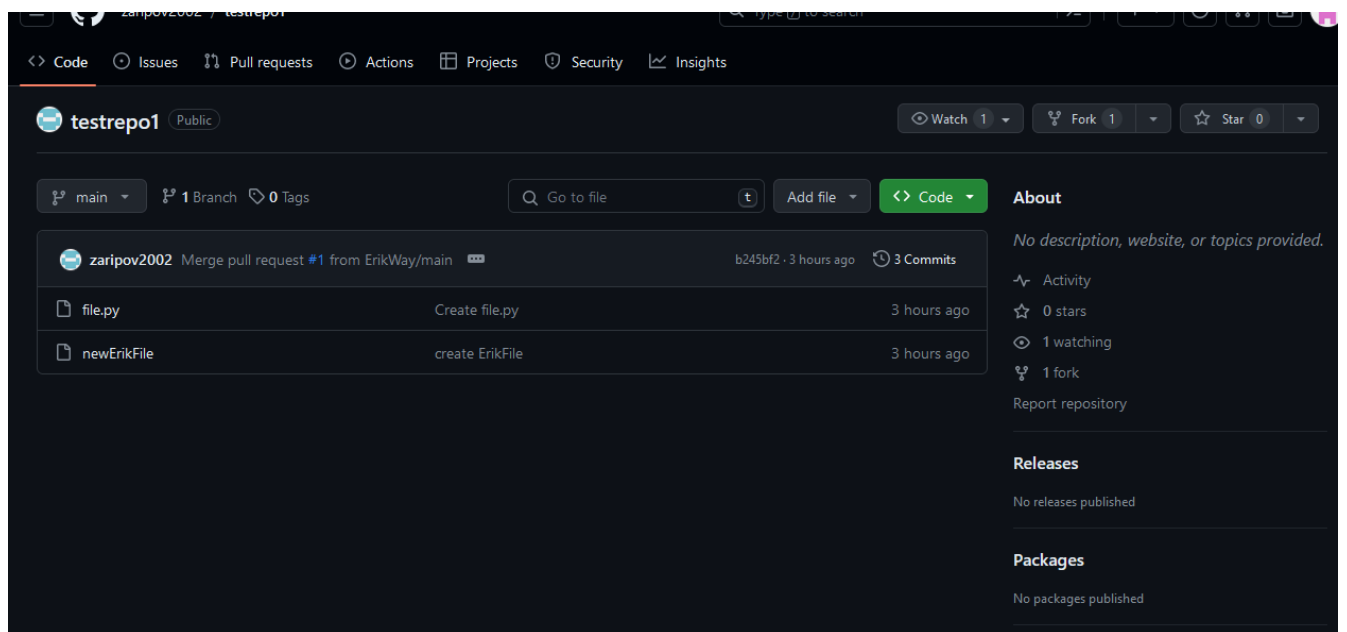
После Нажатия кнопки Create pull request, второй студент заходит на свой репозиторий, переходит во вкладку запросов, после чего попадает на вкладку



После принятия изменения, слияния веток мы получаем



Теперь в главном репозитории будет добавлен новый файл ErikFile



Вывод: в результате проделанной работы, были изучены концепции системы контроля версий Git, были изучены основные команды. В ходе работы возникли некоторые сложности при работе с выгрузкой изменений с помощью команды `git push origin main`, но в результате экспериментов с различными вариациями команд,

было выяснено наиболее результативное использования возможностей выгрузки изменений на гитхаб и ее тонкостей, которые описаны в ходе работы. Так же в ходе работы были получены навыки работы не только с программой Git, а так же порядок использования GitHub, использования внутри него fork-репозитория и выполнения pull request, что углубило мое понимание возможностей совместной разработки на одном проекте большой командой разработчиков.

Выполнил	Стригунов Э.С.
Проверил	Шайхутдинов Д.В.