

Лабораторная работа №1

Основы *Git* и *GitHub*

Цель работы:

Выполнение лабораторной 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.

Порядок работы:

1. Зарегистрировался в *GitHub*, как показано на рисунке 1.

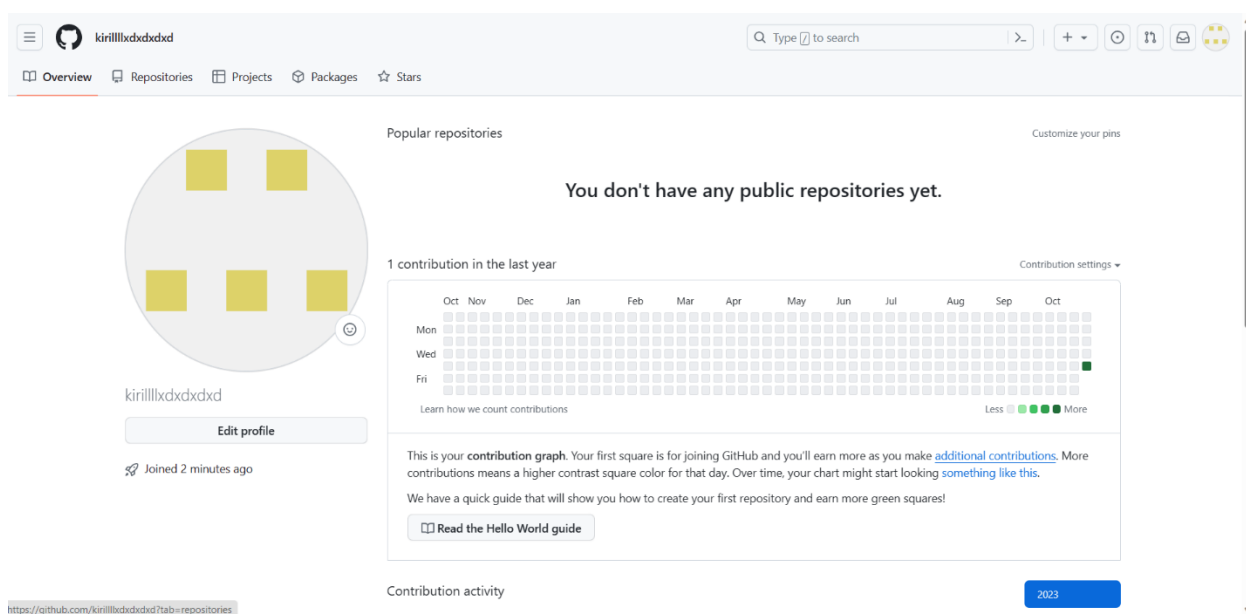


Рисунок 1 - Аккаунт *GitHub*

2. Создал новый репозиторий: задал имя репозитория, добавил описание, выбрал видимость репозитория «публичный», выбрал опцию «*Initialize this repository with a README*», затем отредактировал файл *README*, как показано на рисунке 2.

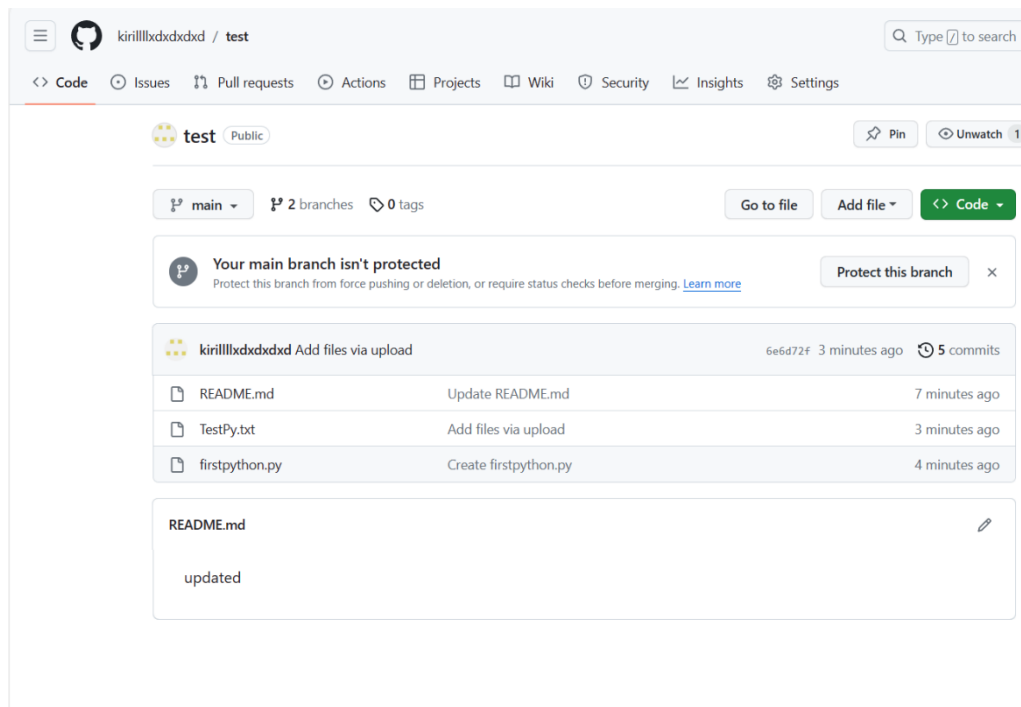
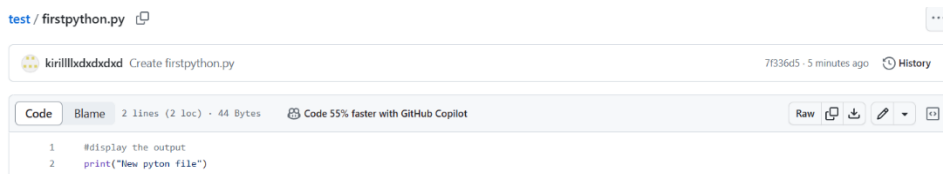


Рисунок 2 - Новый репозиторий

Создал файл *firstpython.py* с помощью встроенного веб-редактора *GitHub*.
Зафиксировал изменения в репозитории, как показано на рисунке 3.

Рисунок 3 - Создание файла



3. Создал новую ветку «*scndBrnch*», создал новый файл. Убедился, что файл, добавленный в дочернюю ветку, не добавляется автоматически в основную ветку, как показано на рисунке 4.

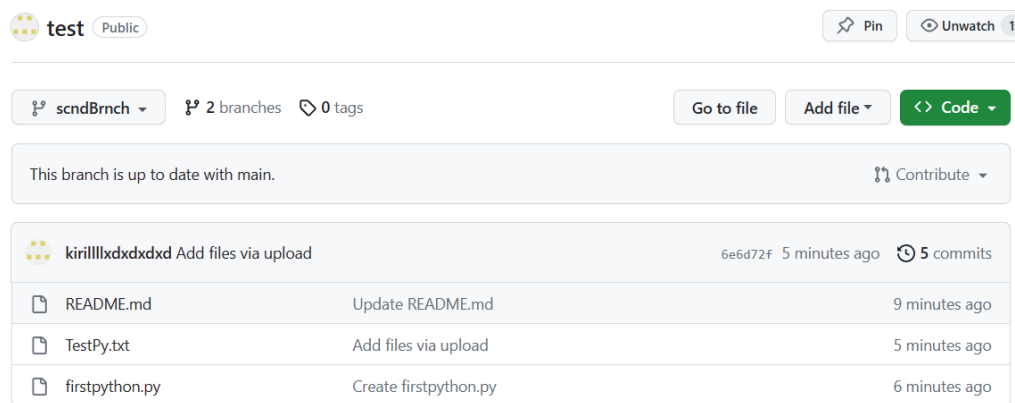


Рисунок 4 - Создание новой ветки

В *scndBrnch* нажал кнопку «*Compare & pull request*», убедился, что в списке указан история ветки, как показано на рисунке 5.

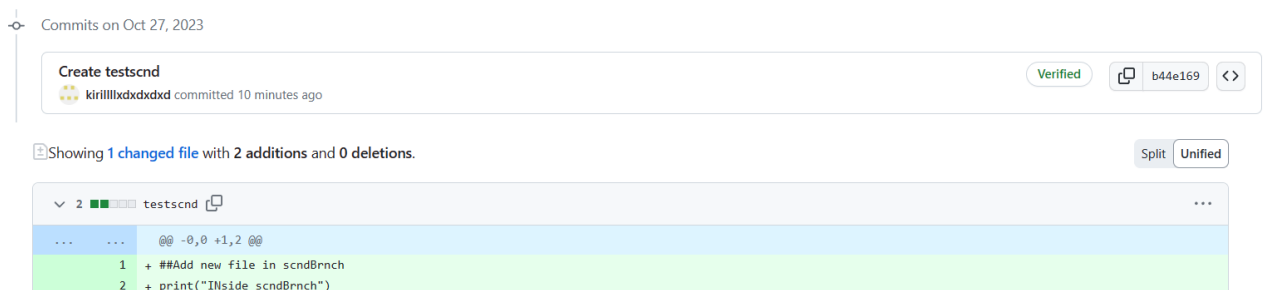


Рисунок 5 - Список измененных файлов

4. Чтобы объединить ветки по запросу *pull request* в проекте, открыл вкладку «*Pull requests*». Отображается список ожидающих запросов на включение. Перешел на нужный *pull request* и нажал «*Merge pull request*», чтобы принять запрос на включение и объединить обновления, как показано на рисунке 6.

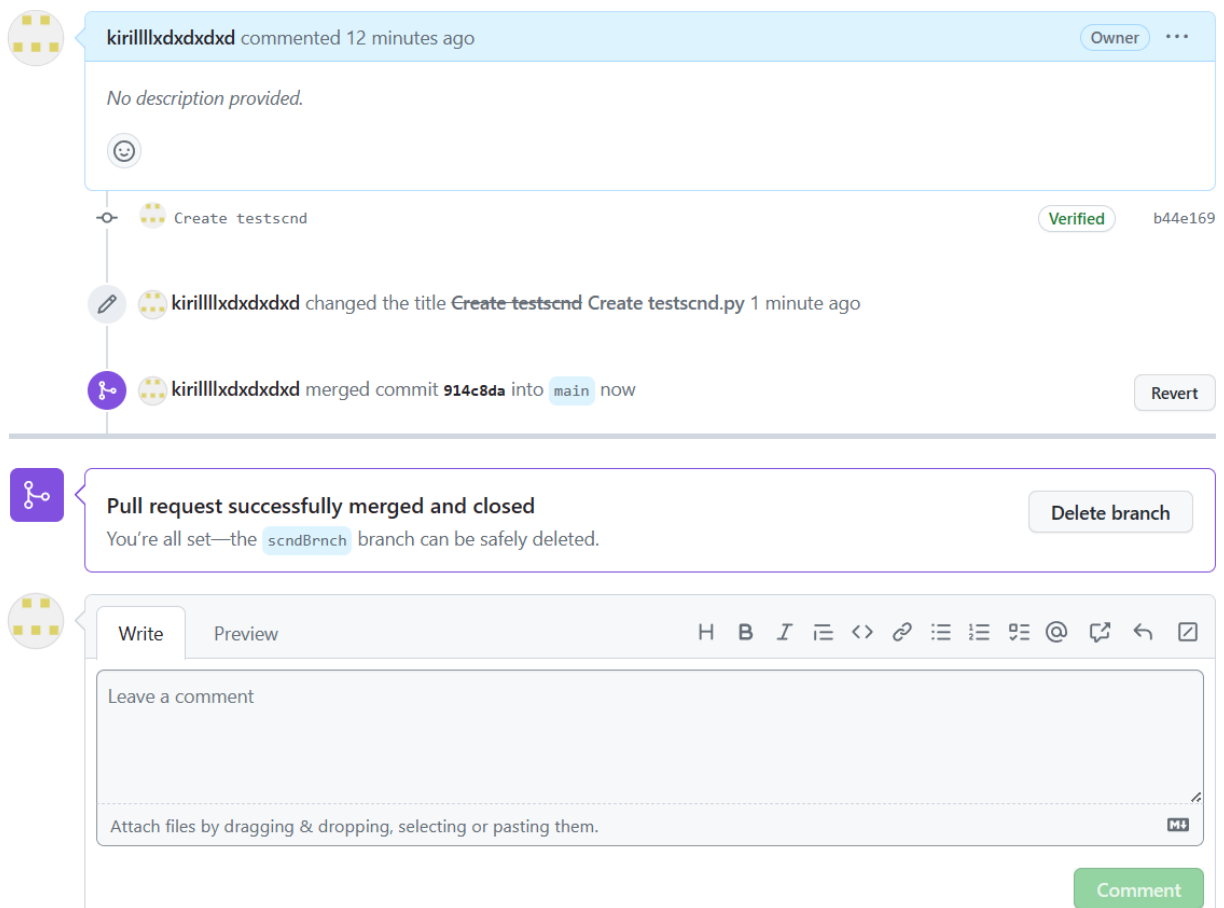


Рисунок 6 - Запрос на включение

5. Произвел работу с локальным репозиторием посредством командной строки *Windows PowerShell* через специальное приложение *Git*.

Создал каталог *myrepo* и перешел в созданный каталог, затем создал локальный репозиторий, вывел на экран содержимое подкаталога *.git*, в котором находится локальный репозиторий, как показано на рисунке 7.

```

deads@KOMPUHTERXD MINGW64 ~
$ mkdir myrepo

deads@KOMPUHTERXD MINGW64 ~
$ cd myrepo

deads@KOMPUHTERXD MINGW64 ~/myrepo
$ git init
Initialized empty Git repository in C:/Users/deads/myrepo/.git/

deads@KOMPUHTERXD MINGW64 ~/myrepo (main)
$ ls -la .git total 40
ls: cannot access 'total': No such file or directory
ls: cannot access '40': No such file or directory
.git:
total 11
drwxr-xr-x 1 deads 197609  0 Oct 27 00:50 ./
drwxr-xr-x 1 deads 197609  0 Oct 27 00:50 ../
-rw-r--r-- 1 deads 197609 21 Oct 27 00:50 HEAD
-rw-r--r-- 1 deads 197609 130 Oct 27 00:50 config
-rw-r--r-- 1 deads 197609 73 Oct 27 00:50 description
drwxr-xr-x 1 deads 197609  0 Oct 27 00:50 hooks/
drwxr-xr-x 1 deads 197609  0 Oct 27 00:50 info/
drwxr-xr-x 1 deads 197609  0 Oct 27 00:50 objects/
drwxr-xr-x 1 deads 197609  0 Oct 27 00:50 refs/

deads@KOMPUHTERXD MINGW64 ~/myrepo (main)
$ |

```

Рисунок 7 - Новый каталог

Создал пустой файл *newfile*, добавил его в репозиторий. Прежде чем зафиксировать изменения, сообщил *git* информацию пользователя. Появившийся в репозитории *newfile* зафиксировал с добавлением сообщения «*added new file*», как показано на рисунке 8.

```

deads@KOMPUHTERXD MINGW64 ~/myrepo (main)
$ git commit -m "added newfile"
[main (root-commit) ef0ab46] added newfile
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 newfile

deads@KOMPUHTERXD MINGW64 ~/myrepo (main)
$ |

```

Рисунок 8 - Новый файл

Создал новую ветку под названием *my1stbranch*, убедился в наличии двух веток в репозитории и переключился с ветки *master* на ветку *my1stbranch*, как показано на рисунке 9.

```
deads@KOMPUHTERXD MINGW64 ~/myrepo (my1stbranch)
$ git branch
main
* my1stbranch
```

Рисунок 9 - Новая ветка

Внес изменения в *newfile* путем добавления текста, убедился, что текст добавлен, как показано на рисунке 10.

```
deads@KOMPUHTERXD MINGW64 ~/myrepo (my1stbranch)
$ echo 'Here is some text in my newfile.' >>newfile

deads@KOMPUHTERXD MINGW64 ~/myrepo (my1stbranch)
$ cat newfile
Here is some text in my newfile.
```

Рисунок 10 - Запись и чтение из файла

Создал новый файл *readme.md* и добавил в репозиторий, проверил изменения в текущей ветке *my1stbranch*, затем добавил *newfile* явно, как показано на рисунке 11.

```
deads@KOMPUHTERXD MINGW64 ~/myrepo (my1stbranch)
$ git status
On branch my1stbranch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   readme.md

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:   newfile

deads@KOMPUHTERXD MINGW64 ~/myrepo (my1stbranch)
$ git add *
warning: in the working copy of 'newfile', LF will be replaced by CRLF the next
time Git touches it

deads@KOMPUHTERXD MINGW64 ~/myrepo (my1stbranch)
$ git status
On branch my1stbranch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   newfile
    new file:   readme.md
```

Рисунок 11 - Добавление файла

Сохранил изменения в ветку, прикрепив сообщение «*added readme.md modified newfile*», далее получил историю последних коммитов — последний коммит в *my1stbranch*, а также предыдущий коммит в *master*, как показано на рисунке 12.

```

deads@KOMPUHTERXD MINGW64 ~/myrepo (my1stbranch)
$ git commit -m "added readme.md modified newfile"
[my1stbranch 87aueb5] added readme.md modified newfile
2 files changed, 1 insertion(+)
create mode 100644 readme.md

deads@KOMPUHTERXD MINGW64 ~/myrepo (my1stbranch)
$ git log
commit 87aueb5b7f35805fc6f31aef9adbc13ae84f74b1 (HEAD -> my1stbranch)
Author: peron olkio <kirill111xd@yandex.ru>
Date:   Fri Oct 27 01:10:42 2023 +0300

    added readme.md modified newfile

commit ef0ab462e8339e956a0bfaf161c7dda6f45db9b2 (main)
Author: peron olkio <kirill111xd@yandex.ru>
Date:   Fri Oct 27 00:58:09 2023 +0300

    added newfile

```

Рисунок 12 - Сохранение изменений

Произвел отмену изменений, используя ярлык *HEAD* для отката последнего коммита, как показано на рисунке 13.

```

deads@KOMPUHTERXD MINGW64 ~/myrepo (my1stbranch)
$ git revert HEAD --no-edit
[my1stbranch fea0944] Revert "added readme.md modified newfile"
Date: Fri Oct 27 01:12:58 2023 +0300
2 files changed, 1 deletion(-)
delete mode 100644 readme.md

```

Рисунок 13 - Отмена изменений

Создал новый *goodfile* и убедился, что файл зафиксирован в *my1stbranch*, как показано на рисунке 14.

```

deads@KOMPUHTERXD MINGW64 ~/myrepo (my1stbranch)
$ touch goodfile

deads@KOMPUHTERXD MINGW64 ~/myrepo (my1stbranch)
$ git add goodfile

deads@KOMPUHTERXD MINGW64 ~/myrepo (my1stbranch)
$ git commit -m "added goodfile"
[my1stbranch 66841d8] added goodfile
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 goodfile

deads@KOMPUHTERXD MINGW64 ~/myrepo (my1stbranch)
$ git log
commit 66841d83dd58a2246ae299d84306187d1ce32070 (HEAD -> my1stbranch)
Author: peron olkio <kirill111xd@yandex.ru>
Date:   Fri Oct 27 01:13:57 2023 +0300

    added goodfile

commit fea0944d1c01ba9d4bfa60ce58ffbf1e12455747
Author: peron olkio <kirill111xd@yandex.ru>
Date:   Fri Oct 27 01:12:58 2023 +0300

    Revert "added readme.md modified newfile"

    This reverts commit 87aueb5b7f35805fc6f31aef9adbc13ae84f74b1.

commit 87aueb5b7f35805fc6f31aef9adbc13ae84f74b1
Author: peron olkio <kirill111xd@yandex.ru>
Date:   Fri Oct 27 01:10:42 2023 +0300

    added readme.md modified newfile

commit ef0ab462e8339e956a0bfaf161c7dda6f45db9b2 (main)
Author: peron olkio <kirill111xd@yandex.ru>
Date:   Fri Oct 27 00:58:09 2023 +0300

    added newfile

```

Рисунок 14 - Новый файл

Далее объединил содержимое *my1stbranch* с основной веткой, для этого сначала сделал ветку *master* активной. Произвел слияние веток и вывел на экран журнал. После того, как слияние успешно завершилось, удалил ветку *my1stbranch*, как показано на рисунке 15.

```

deads@KOMPUHTERXD MINGW64 ~/myrepo (my1stbranch)
$ git merge my1stbranch
Already up to date.

deads@KOMPUHTERXD MINGW64 ~/myrepo (my1stbranch)
$ git log
commit 66841d83dd58a2246ae299d84306187d1ce32070 (HEAD -> my1stbranch)
Author: peron olkio <kirill11xd@yandex.ru>
Date:   Fri Oct 27 01:13:57 2023 +0300

    added goodfile

```

Рисунок 15 - Слияние веток

Задание №1:

git checkout -b newbranch //Создал новую ветку *newbranch*

ni newbranchfile //Создал пустой файл *newbranchfile*

git add newbranchfile //Добавил файл в свою ветку

git commit -m "добавлен newbranchfile" //Зафиксировал изменения в новой ветке

git revert HEAD --no-edit //Отменил последние зафиксированные изменения

ni newgoodfile //Создал новый файл с именем *newgoodfile*

git add newgoodfile //Добавил последний файл в новую ветку

git commit -m "добавлен newgoodfile" //Зафиксировал изменения

git checkout master //Переключился на основную ветку

git merge newbranch //Объединил изменения в новой ветке с основной

Результат выполнения представлен на рисунках 16 и 17.


```
deads@KOMPUHTERXD MINGW64 ~/myrepo (my1stbranch)
$ git checkout -b newbranch
Switched to a new branch 'newbranch'

deads@KOMPUHTERXD MINGW64 ~/myrepo (newbranch)
$ git checkout newbranch
Already on 'newbranch'

deads@KOMPUHTERXD MINGW64 ~/myrepo (newbranch)
$ touch newbranchfile

deads@KOMPUHTERXD MINGW64 ~/myrepo (newbranch)
$ git add newbranch
fatal: pathspec 'newbranch' did not match any files

deads@KOMPUHTERXD MINGW64 ~/myrepo (newbranch)
$ git add newbranchfile

deads@KOMPUHTERXD MINGW64 ~/myrepo (newbranch)
$ git commit -m "added newbranchfile"
[newbranch e8e2307] added newbranchfile
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 newbranchfile

deads@KOMPUHTERXD MINGW64 ~/myrepo (newbranch)
$ git revert HEAD --no-edit
[newbranch d8adafe] Revert "added newbranchfile"
Date: Fri Oct 27 01:22:15 2023 +0300
1 file changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 newbranchfile

deads@KOMPUHTERXD MINGW64 ~/myrepo (newbranch)
$ touch newgoodfile

deads@KOMPUHTERXD MINGW64 ~/myrepo (newbranch)
$ git add newgoodfile

deads@KOMPUHTERXD MINGW64 ~/myrepo (newbranch)
$ git commit -m "added newgoodfile"
[newbranch 317c951] added newgoodfile
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 newgoodfile

deads@KOMPUHTERXD MINGW64 ~/myrepo (newbranch)
$ git status
On branch newbranch
nothing to commit, working tree clean

deads@KOMPUHTERXD MINGW64 ~/myrepo (newbranch)
$ git log
commit 317c9514886a4c88fa6dccbfd745bd985f34e3a9 (HEAD -> newbranch)
Author: peron olkio <kirill111xd@yandex.ru>
Date: Fri Oct 27 01:23:07 2023 +0300

    added newgoodfile
```

Рисунок 16 - Процесс выполнения задания

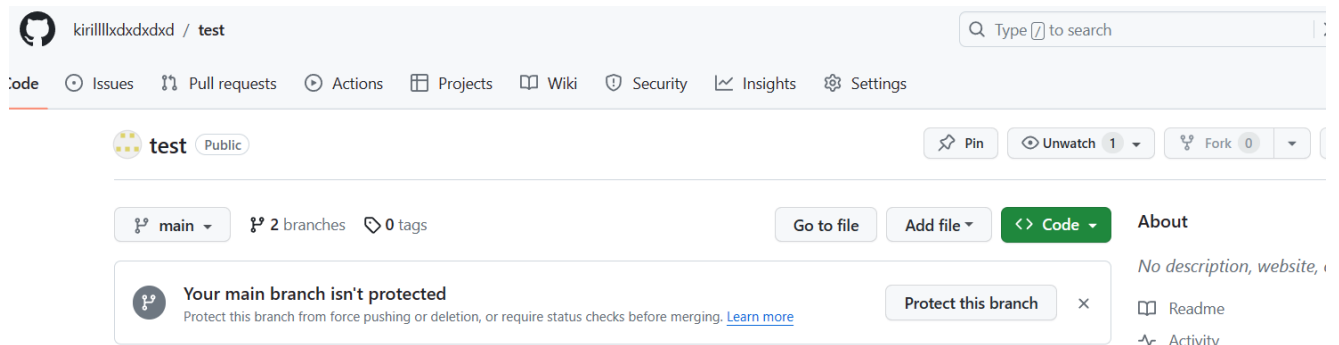


Рисунок 18 - *Fork* проекта второго студента

Выполнил *clone* проекта в локальный репозиторий с помощью *Windows PowerShell*, как показано на рисунке 19.

```
PS C:\Windows\system32> cd C:\1pr
PS C:\1pr> mkdir fork

Каталог: C:\1pr

Mode                LastWriteTime         Length Name
----                -
d-----         27.10.2023          2:47         fork
```

Рисунок 19 - *Clone* проекта

Добавил новый файл в репозиторий, зафиксировал изменения, как показано на рисунке 20.

```
PS C:\1pr> git clone https://github.com/kirilllllxdxdxdxd/test.git
Cloning into 'test'...
remote: Enumerating objects: 19, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 19 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (19/19), 4.38 KiB | 4.38 MiB/s, done.
Resolving deltas: 100% (1/1), done.
PS C:\1pr> cd test
PS C:\1pr\test> ni windowshell.txt

Каталог: C:\1pr\test

Mode                LastWriteTime         Length Name
----                -
-a-----         27.10.2023   2:47             0 windowshell.txt
```

Рисунок 20 - Новый файл

Выполнил синхронизацию с *fork*-репозиторием, как показано на рисунке 21.

```
PS C:\1pr\test> echo 'git test' >> windowshell.txt
PS C:\1pr\test> git add *
PS C:\1pr\test> git commit -m "добавлен файл"
[main 2933b7b] добавлен файл
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 windowshell.txt
PS C:\1pr\test> git push origin main
info: please complete authentication in your browser...
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 321 bytes | 321.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/kirilllllxdxdxdxd/test.git
 914c8da..2933b7b  main -> main
PS C:\1pr\test> █
```

Рисунок 21 - Синхронизация с *fork*-репозиторием

Вывод: изучили концепцию *Git*, основанную на понятиях репозитория и ветвления версий ПО, изучили порядок использования *GitHub* и его базовых операций, научились работать с онлайн-хостингом *GitHub*.