



МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технологический университет «СТАНКИН»
(ФГБОУ ВО «МГТУ «СТАНКИН»)

Институт цифровых
интеллектуальных систем

Кафедра
компьютерных систем управления

Дисциплина «Основы системного программного обеспечения»

Отчет по лабораторной работе № 1
Работа Git Hub

Выполнил
студент гр. АДБ-20-06:

Афонин Е.К.

(дата)

(подпись)

Проверил
к.т.н., доцент

Ковалев И.А.

(дата)

(подпись)

Москва 2023 г.

Содержание

- 1. Цель**
- 2. Задачи работы**
- 3. Описание и выполнение тестовых и индивидуальных заданий**

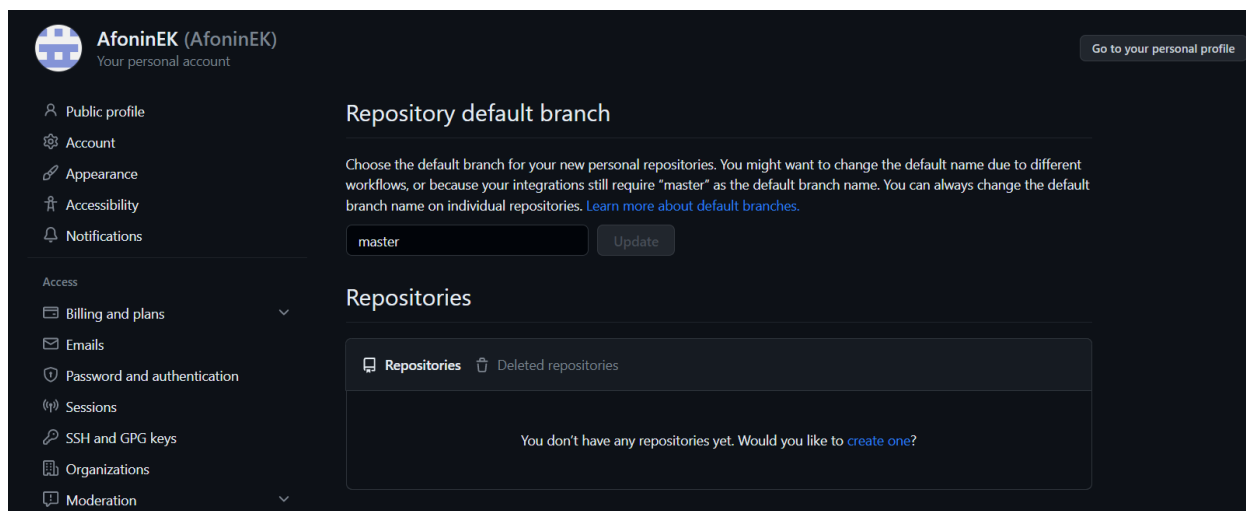
1.Цель: научиться работать с системой контроля версий Git Hub.

2. Задачи работы:

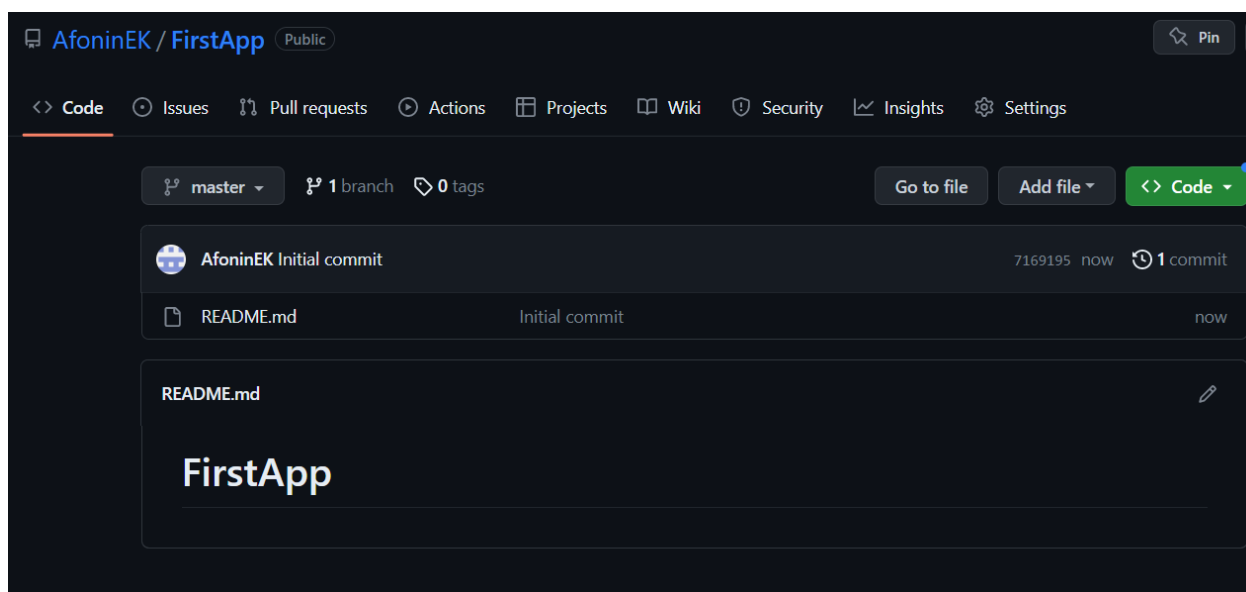
- 1) Создать учетную запись на Git Hub.
- 2) Создать репозиторий и научиться добавлять туда файлы.
- 3) Научиться отправлять изменения на сервер.

3. Описание и выполнение тестовых и индивидуальных заданий

Создадим новый репозиторий FirstApp.

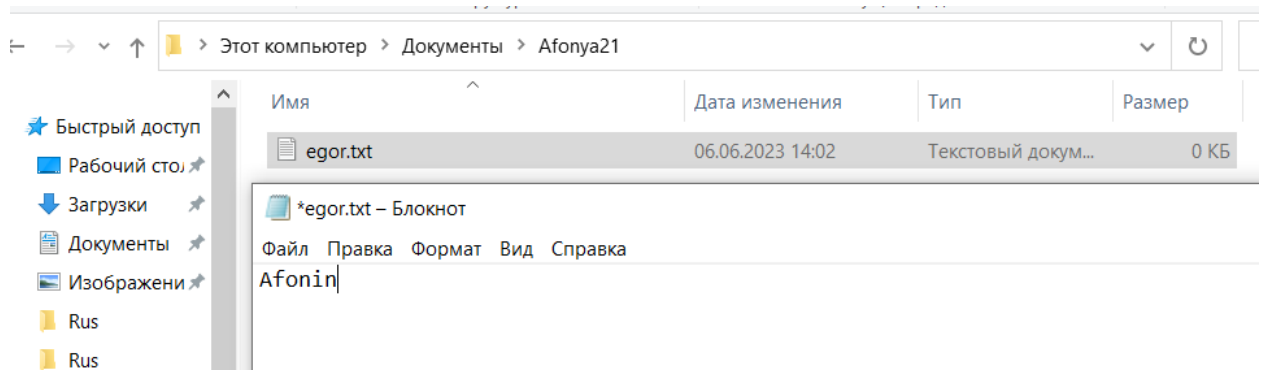


Увидим, что наш репозиторий создан.



Создадим теперь локальную папку, которая будет связана с нашим облачным git хранилищем.

В ней создадим файл egor.txt с фамилией.



Настроим наше имя и адрес электронной почты. А также проинициализируем эту папку как git репозиторий.

```
Microsoft Windows [Version 10.0.19045.2965]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\egor1\OneDrive\Документы\Аfonya21>git config --global user.name "АфонинЕК"

C:\Users\egor1\OneDrive\Документы\Аfonya21>git config --global user.email afonin_zhora@bk.ru

C:\Users\egor1\OneDrive\Документы\Аfonya21>git init
Initialized empty Git repository in C:/Users/egor1/OneDrive/Документы/Аfonya21/.git/

C:\Users\egor1\OneDrive\Документы\Аfonya21>
```

Командная строка вернула нам сообщение, что проинициализирована пустая git директория. Теперь используя проводник windows создадим в своей папке любой текстовый файл и пропишем туда наше имя. Введем команду:

```
C:\Users\egor1\OneDrive\Документы\Аfonya21>git status
On branch master

No commits yet

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

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

C:\Users\egor1\OneDrive\Документы\Аfonya21>
```

Нам отобразилось сообщение, что есть новый файл, но он не отслеживается. Это значит, что система еще не знает, нужно следить за изменениями в файле или его можно игнорировать. Для этого нужно добавить файлы командой add.

```
C:\Users\egor1\OneDrive\Документы\Afonya21>git add .

C:\Users\egor1\OneDrive\Документы\Afonya21>git status
On branch master

No commits yet

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

C:\Users\egor1\OneDrive\Документы\Afonya21>_
```

Выполнив эту команду посмотрим на статус репозитория и видим, что файл готов к коммиту.

Теперь закоммитим наш файл.

Чтобы зафиксировать изменения, нам нужно хотя бы одно изменение в области подготовки.

```
C:\Users\egor1\OneDrive\Документы\Afonya21>git commit -m "first commit"
[master (root-commit) 605de44] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 egor.txt

C:\Users\egor1\OneDrive\Документы\Afonya21>git status
On branch master
nothing to commit, working tree clean

C:\Users\egor1\OneDrive\Документы\Afonya21>
```

Коммит создан. И после вызова `git status` видно, что файлов для коммитов больше нет.

Теперь свяжем наш локальный репозиторий с репозиторием на GitHub, для этого выполним следующее.

```
C:\Users\egor1\OneDrive\Документы\Afonya21>git remote add origin https://github.com/AfoninEK/FirstApp.git

C:\Users\egor1\OneDrive\Документы\Afonya21>git remote -v
origin  https://github.com/AfoninEK/FirstApp.git (fetch)
origin  https://github.com/AfoninEK/FirstApp.git (push)
```

Мы указываем, что подключаемся к репозиторию по URI, так как проект может иметь несколько репозиторий одновременно нужно давать им разные имена, а также указать `origin` у главного репозитория.

Затем посмотрим с помощью команды `git remote -v` к какому проекту мы подключены.

`git pull origin master --allow-unrelated-histories` — команда нужная, если у нас указано несколько репозиториях и у нас образовалось запутывание веток. В нашем случае таких проблем не было. Так же можно удалить не правильный репозиторий командой `git remote rm origin`.

В нашем случае у нас есть файл и на сервере и локальном репозитории, поэтому нам вначале надо получить все данные с нашего репозитория, а потом уже их заливать на сервер нашими локальными данными.





Мы сделали изменения в нашем репозитории, теперь другие пользователи могут скачать изменения при помощи команды `pull`.

`git pull origin master`









Теперь перешлем локальный коммит на сервер. Для этого существует команда `push`. Она принимает два параметра: имя удаленного репозитория (мы назвали нашу ветку `origin`) и ветку, в которую необходимо внести изменения (`master` — это ветка по умолчанию для всех репозиториях).

```
C:\Users\egor1\OneDrive\Документы\Afonya21>git push origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 519 bytes | 259.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/AfoninEK/FirstApp.git
  7169195..91526a6  master -> master
C:\Users\egor1\OneDrive\Документы\Afonya21>_
```

Видим, что изменения отобразились и на нашем сервере.

	AfoniaEK Merge branch 'master' of https://github.com/AfoninEK/FirstApp	91526a6 2 minutes ago	 3 commits
	README.md	Initial commit	18 minutes ago
	egor.txt	first commit	6 minutes ago

Если нажмем на «commit», то увидим все наши изменения.

Commits			
 master			
 Commits on Jun 6, 2023			
Merge branch 'master' of https://github.com/AfoninEK/FirstApp		91526a6	
AfoniaEK committed 2 minutes ago			
first commit		605de44	
AfoniaEK committed 7 minutes ago			
Initial commit		7169195	
AfoniaEK committed 18 minutes ago			
	Verified		

Теперь создадим новую ветку.

Ветка master создается сама, а дополнительные, мы создаем по своему желанию, с помощью команды git branch.

Теперь переключимся на другую ветку. Это мы делаем с помощью команды checkout

```
C:\Users\egor1\OneDrive\Документы\Afonya21>git branch second

C:\Users\egor1\OneDrive\Документы\Afonya21>git branch
* master
  second

C:\Users\egor1\OneDrive\Документы\Afonya21>git chekout second
git: 'chekout' is not a git command. See 'git --help'.

The most similar command is
    checkout

C:\Users\egor1\OneDrive\Документы\Afonya21>git checkout second
Switched to branch 'second'

C:\Users\egor1\OneDrive\Документы\Afonya21>_
```

Создаем в нашем локальном репозитории test2.txt и напишем там свою фамилию, затем добавляем в области подготовленных файлов, закомитем и отправим на сервер.

```
C:\Users\egor1\OneDrive\Документы\Afonya21>git add .

C:\Users\egor1\OneDrive\Документы\Afonya21>git status
On branch second
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   egor2.txt

C:\Users\egor1\OneDrive\Документы\Afonya21>
```

```

C:\Users\egor1\OneDrive\Документы\Afonya21>git commit -m "two"
[second 4e3a626] two
1 file changed, 1 insertion(+)
create mode 100644 egor2.txt

C:\Users\egor1\OneDrive\Документы\Afonya21>git push origin second
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), 303 bytes | 303.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'second' on GitHub by visiting:
remote:   https://github.com/AfoninEK/FirstApp/pull/new/second
remote:
To https://github.com/AfoninEK/FirstApp.git
 * [new branch]      second -> second

C:\Users\egor1\OneDrive\Документы\Afonya21>

```

Зайдем на Git Hub и посмотрим, что в ветке master нет нашего файла, а в ветке second он присутствует.

This branch is 1 commit ahead of master. [Contribute](#)

AfoninEK two 4e3a626 1 minute ago 4 commits

README.md	Initial commit	28 minutes ago
egor.txt	first commit	17 minutes ago
egor2.txt	two	1 minute ago

master 2 branches 0 tags [Go to file](#) [Add file](#) [Code](#)

Your master branch isn't protected
Protect this branch from force pushing or deletion, or require status checks before merging. [Learn more](#) [Protect this branch](#)

AfoninEK Merge branch 'master' of https://github.com/AfoninEK/FirstApp 91526a6 13 minutes ago 3 commits

README.md	Initial commit	29 minutes ago
egor.txt	first commit	17 minutes ago

Переключимся снова на ветку master. Теперь если через проводник windows зайти в папку с проектом, то файла test2.txt, там не будет видно.

```

C:\Users\egor1\OneDrive\Документы\Afonya21>git checkout master
Switched to branch 'master'

C:\Users\egor1\OneDrive\Документы\Afonya21>

```


Этот компьютер > Документы > Afonya21					▼	🔄
Имя	Дата изменения	Тип	Размер			
.git	06.06.2023 14:34	Папка с файлами				
egor.txt	06.06.2023 14:03	Текстовый докум...	1 КБ			
README.md	06.06.2023 14:17	Файл "MD"	1 КБ			

Чтобы он появился, нужно воспользоваться merge для объединения веток.

```
C:\Users\egor1\OneDrive\Документы\Afonya21>git merge second
Updating 91526a6..4e3a626
Fast-forward
 egor2.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 egor2.txt
```

Теперь удалим ветку second.

```
C:\Users\egor1\OneDrive\Документы\Afonya21>git branch -d second
Deleted branch second (was 4e3a626).

C:\Users\egor1\OneDrive\Документы\Afonya21>
```

Создадим ветку, в которой будут одинаковые файлы.

```
C:\Users\egor1\OneDrive\Документы\Afonya21>git branch newdev

C:\Users\egor1\OneDrive\Документы\Afonya21>git branch
* master
  newdev
```

Переключимся на неё

```
C:\Users\egor1\OneDrive\Документы\Afonya21>git checkout newdev
Switched to branch 'newdev'
```

Добавим в файл изменения, а, то есть имя и отчество и закомитим их.

```
C:\Users\egor1\OneDrive\Документы\Afonya21>git add .

C:\Users\egor1\OneDrive\Документы\Afonya21>git status
On branch newdev
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   egor.txt

C:\Users\egor1\OneDrive\Документы\Afonya21>git commit -m "three commit"
[newdev 7cd791a] three commit
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\egor1\OneDrive\Документы\Afonya21>
```

Переключимся на ветку master и посмотрим файл, в нем только имя отчество в другой ветке.

Теперь объединим ветки

Откроем файл снова и увидим изменения

```
C:\Users\egor1\OneDrive\Документы\Afonya21>git merge newdev
Updating 4e3a626..7cd791a
Fast-forward
 egor.txt | 2 + -
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Откроем файл снова, там есть и имя, и отчество

Переключимся на ветку newdev и удалим в отчестве несколько букв

Зафиксируем, закоммитим

Переключимся на ветку master, добавим к отчеству несколько букв

Зафиксируем, закоммитим

Объединим ветки

```
C:\Users\egor1\OneDrive\Документы\Afonya21>git checkout newdev
Switched to branch 'newdev'

C:\Users\egor1\OneDrive\Документы\Afonya21>git add .

C:\Users\egor1\OneDrive\Документы\Afonya21>git status
On branch newdev
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   egor.txt

C:\Users\egor1\OneDrive\Документы\Afonya21>git commit -m "four commit"
[newdev 56e9ea2] four commit
1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\egor1\OneDrive\Документы\Afonya21>git checkout master
Switched to branch 'master'

C:\Users\egor1\OneDrive\Документы\Afonya21>git add .

C:\Users\egor1\OneDrive\Документы\Afonya21>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   egor.txt

C:\Users\egor1\OneDrive\Документы\Afonya21>git commit -m "five commit"
[master 7c641ba] five commit
1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\egor1\OneDrive\Документы\Afonya21>git merge newdev
Auto-merging egor.txt
CONFLICT (content): Merge conflict in egor.txt
Automatic merge failed; fix conflicts and then commit the result.

C:\Users\egor1\OneDrive\Документы\Afonya21>
```

Теперь ничего не получится, так как изменения есть в обеих ветках.
Разрешим конфликт.

```

C:\Users\egor1\OneDrive\Документы\Afonya21>git diff
diff --cc egor.txt
index 5ecc598,ae16ca1..0000000
--- a/egor.txt
+++ b/egor.txt
@@@ -1,1 -1,1 +1,5 @@@
- Afonin Egorka
- Afonin Eg
++<<<<<< HEAD
++Afonin Egorka
++=====
++Afonin Eg
++>>>>>> newdev

```

Исправим ошибку вручную, удалив разделители и выбрав нужный нам вариант. Сохраняем файл, фиксируем изменения, коммитим, отправляем на сервер. И удаляем ветку newdev.

```

C:\Users\egor1\OneDrive\Документы\Afonya21>git add .

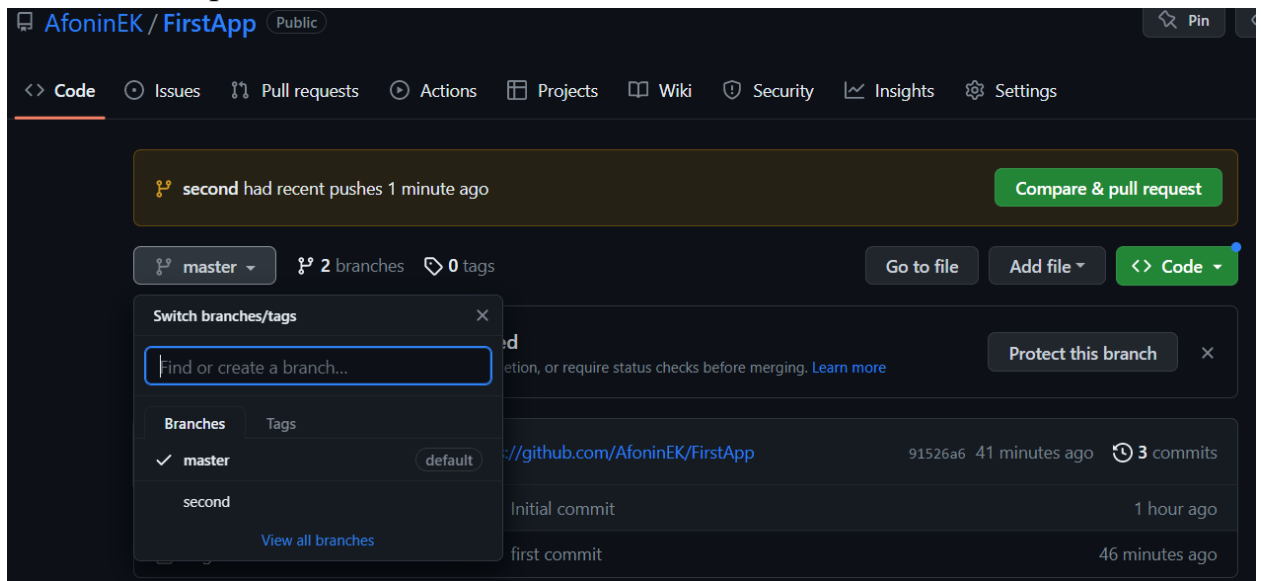
C:\Users\egor1\OneDrive\Документы\Afonya21>git commit -m "test"
[master 1bf05e1] test

C:\Users\egor1\OneDrive\Документы\Afonya21>git push origin master
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (10/10), 925 bytes | 462.00 KiB/s, done.
Total 10 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/AfoninEK/FirstApp.git
  91526a6..1bf05e1  master -> master

C:\Users\egor1\OneDrive\Документы\Afonya21>git branch -d newdev
Deleted branch newdev (was 56e9ea2).

```

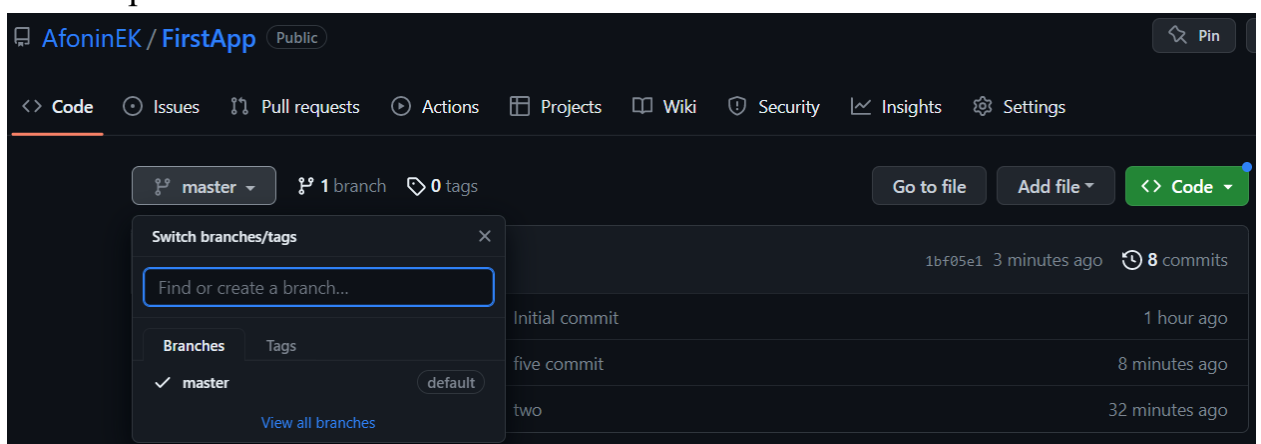
Если посмотрим на Git Hub, то наши ветки там остались.



Удалим их.

```
C:\Users\egor1\OneDrive\Документы\Afonya21>git push origin --delete second
To https://github.com/AfoninEK/FirstApp.git
- [deleted]          second
```

Посмотрим изменения.



Также Git позволяет вернуть выбранный файл к состоянию на момент определенного коммита. Это делается командой checkout. Для того, чтобы просмотреть все наши коммиты используем команду git log

```
C:\Users\egor1\OneDrive\Документы\Afonya21>git log
commit 1bf05e1d25b9c49dd11a5c8f9b2926867b54a489 (HEAD -> master, origin/master)
Merge: 7c641ba 56e9ea2
Author: AfoninEK <afonin_zhora@bk.ru>
Date: Tue Jun 6 14:58:06 2023 +0300

    test

commit 7c641ba89ac453e9091115d7e177f0e878653354
Author: AfoninEK <afonin_zhora@bk.ru>
Date: Tue Jun 6 14:53:08 2023 +0300

    five commit

commit 56e9ea23c98f81a210d52c99e4c038fcde79d29f
Author: AfoninEK <afonin_zhora@bk.ru>
Date: Tue Jun 6 14:50:39 2023 +0300

    four commit

commit 7cd791a14812340e2c72bf9abf0bfa184f794935
Author: AfoninEK <afonin_zhora@bk.ru>
Date: Tue Jun 6 14:43:20 2023 +0300

    three commit

commit 4e3a6268386e35c8d35d5c7c5ac92f9af7c5bea9
Author: AfoninEK <afonin_zhora@bk.ru>
Date: Tue Jun 6 14:28:26 2023 +0300

    two

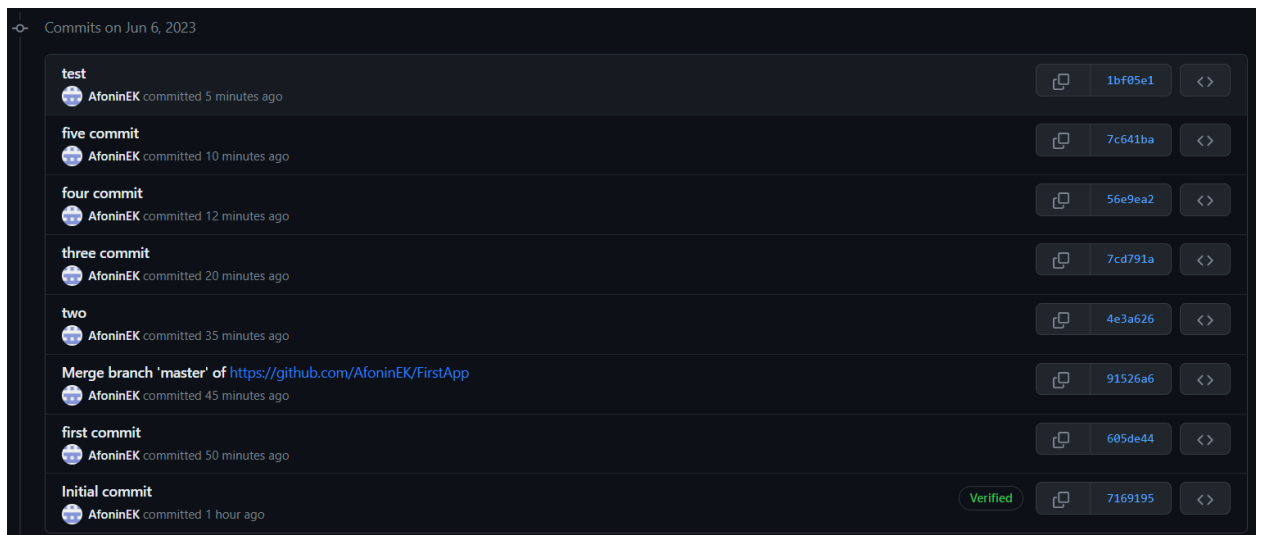
commit 91526a69a28157fcc5d677188083a060bc045fd3
Merge: 605de44 7169195
Author: AfoninEK <afonin_zhora@bk.ru>
Date: Tue Jun 6 14:17:33 2023 +0300

    Merge branch 'master' of https://github.com/AfoninEK/FirstApp

commit 605de44f78d8281fc31cb35f2849b314bada2d5c
Author: AfoninEK <afonin_zhora@bk.ru>
Date: Tue Jun 6 14:12:39 2023 +0300

    first commit
```

Или можно использовать github.



Выберем любой коммит, на который хотим откатиться, указав его первые несколько символов.

```
C:\Users\egor1\OneDrive\Документы\Afonya21>git checkout 91526
Note: switching to '91526'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 91526a6 Merge branch 'master' of https://github.com/AfoninEK/FirstApp
C:\Users\egor1\OneDrive\Документы\Afonya21>git branch
* (HEAD detached at 91526a6)
  master
C:\Users\egor1\OneDrive\Документы\Afonya21>git checkout master
Previous HEAD position was 91526a6 Merge branch 'master' of https://github.com/AfoninEK/FirstApp
Switched to branch 'master'
```

Если посмотреть файлы теперь, то видно, что они поменялись – мы вернулись назад.

Чтобы это произошло создалась псевдо-ветка начинающаяся на этом коммите. Дальнейший правильный вариант заключается в том, что нужно продолжить работать на новой ветке, а затем смириваться.

Рассмотрим ситуацию, если мы закрепим файлы, но еще не комитили и хотим убрать их из области закрепления.

```
C:\Users\egor1\OneDrive\Документы\Afonya21>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:   egor.txt

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

C:\Users\egor1\OneDrive\Документы\Afonya21>git add /
fatal: /: '/' is outside repository at 'C:/Users/egor1/OneDrive/Документы/Afonya21'

C:\Users\egor1\OneDrive\Документы\Afonya21>git add .

C:\Users\egor1\OneDrive\Документы\Afonya21>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   egor.txt

C:\Users\egor1\OneDrive\Документы\Afonya21>git reset HEAD
Unstaged changes after reset:
M       egor.txt

C:\Users\egor1\OneDrive\Документы\Afonya21>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:   egor.txt

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

C:\Users\egor1\OneDrive\Документы\Afonya21>
```

Теперь снова добавим в область закрепления и закомитим, но еще не отправляя на сервер, можно вызвать ту же команду и тогда все, что осталось незакомиченным будет удалено.


```

C:\Users\egor1\OneDrive\Документы\Afonya21>git add .

C:\Users\egor1\OneDrive\Документы\Afonya21>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   egor.txt

C:\Users\egor1\OneDrive\Документы\Afonya21>git commit -m "head_commit"
[master 4044aa3] head_commit
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\egor1\OneDrive\Документы\Afonya21>git reset HEAD

C:\Users\egor1\OneDrive\Документы\Afonya21>git status
On branch master
nothing to commit, working tree clean

C:\Users\egor1\OneDrive\Документы\Afonya21>_

```

Если хотим изменить комментарий можно вызвать команду: `git commit – amend`

Откроется консоле в редакторе Vi.

```

C:\Users\egor1\OneDrive\Документы\Afonya21>git commit --amend
[master 0530249] head_commit
Date: Tue Jun 6 15:11:36 2023 +0300
 1 file changed, 1 insertion(+), 1 deletion(-)

```

В большинстве проектов есть файлы или целые директории, в которые мы не хотим (и, скорее всего, не захотим) коммитить. Создадим файл `.gitignore` и сохраним его в директорию проекта.

И перечислим файлы которые хотим игнорировать.

.git	06.06.2023 15:13	Папка с файлами	
.gitignore	06.06.2023 15:13	Текстовый докум...	0 КБ
egor.txt	06.06.2023 15:06	Текстовый докум...	1 КБ
egor2.txt	06.06.2023 15:05	Текстовый докум...	1 КБ
egor3.txt	06.06.2023 15:14	Текстовый докум...	0 КБ
README.md	06.06.2023 14:17	Файл "MD"	1 КБ

*.gitignore – Блокнот

Файл

Правка

Формат

Вид

Справка

/egor3.txt

```
C:\Users\egor1\OneDrive\Документы\Afonya21>git add .

C:\Users\egor1\OneDrive\Документы\Afonya21>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .gitignore

C:\Users\egor1\OneDrive\Документы\Afonya21>git commit -m "last"
[master 4919421] last
1 file changed, 1 insertion(+)
create mode 100644 .gitignore
```

```
C:\Users\egor1\OneDrive\Документы\Afonya21>git push origin master
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 535 bytes | 535.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/AfoninEK/FirstApp.git
1bf05e1..4919421  master -> master
```

The screenshot shows the GitHub interface for the repository 'AfoninEK / FirstApp'. The repository is public and has 1 branch (master) and 0 tags. The commit history shows a commit 'AfoninEK last' with hash 4919421, made 2 minutes ago, containing 10 commits. The commit details show a list of files: .gitignore (last, 2 minutes ago), README.md (Initial commit, 1 hour ago), egor.txt (head_commit, 5 minutes ago), and egor2.txt (two, 50 minutes ago).

File	Commit	Time
.gitignore	last	2 minutes ago
README.md	Initial commit	1 hour ago
egor.txt	head_commit	5 minutes ago
egor2.txt	two	50 minutes ago

Как мы видим наш файл есть среди локальных, но на сервер он не был отправлен и был проигнорирован.