



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-20-08:

Ипполитова В.В.

(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:

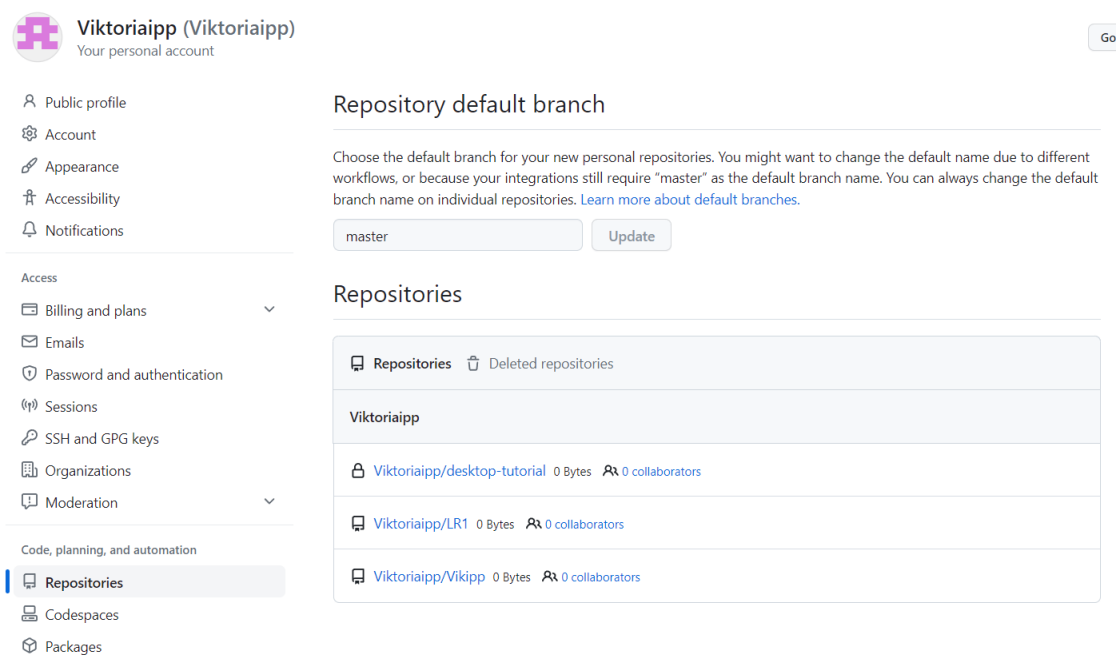
1) Создать учетную запись на Git Hub.

2) Создать репозиторий и научиться догбавлять туда файлы.

3) Научиться отправлять изменения на сервер.

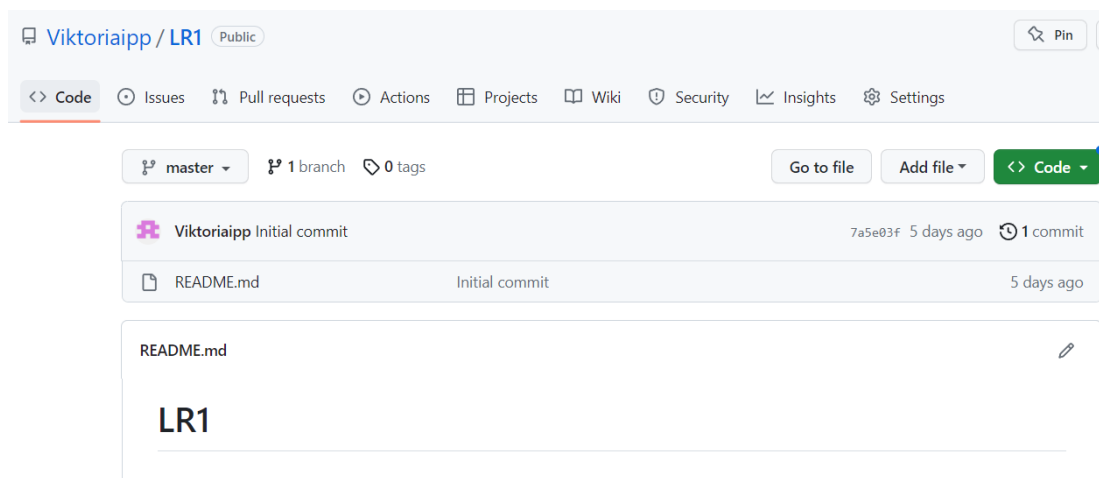
3. Описание и выполнение тестовых и индивидуальных заданий

Создадим новый репозиторий LR1.



The screenshot shows the GitHub profile page for user Viktoriaipp. On the left is a sidebar with navigation links: Public profile, Account, Appearance, Accessibility, Notifications, Access, Billing and plans, Emails, Password and authentication, Sessions, SSH and GPG keys, Organizations, Moderation, Code, planning, and automation, Repositories (selected), Codespaces, and Packages. The main content area has a section 'Repository default branch' with a dropdown set to 'master' and an 'Update' button. Below this is the 'Repositories' section, which lists three repositories: Viktoriaipp/desktop-tutorial, Viktoriaipp/LR1, and Viktoriaipp/Vikipp. Each repository entry shows its name, size (0 Bytes), and number of collaborators (0).

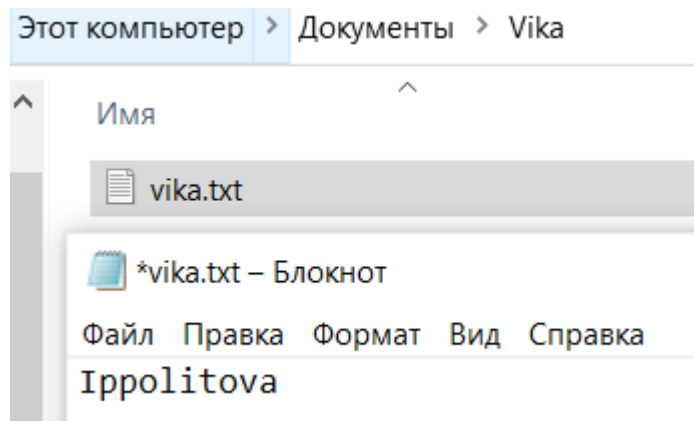
Увидим, что наш репозиторий создан.



The screenshot shows the GitHub repository page for Viktoriaipp/LR1. The repository is public. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation bar, there's a section for the repository's state: master branch, 1 branch, 0 tags. There are buttons for 'Go to file', 'Add file', and 'Code'. The main content area shows the repository's history, with an initial commit by Viktoriaipp 5 days ago. Below the history, there's a section for the README.md file, which contains the text 'LR1'.

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

В ней создадим файл `vika.txt` с фамилией.



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

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.2965]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\Viktoriaipp\Documents\Vika>git config --global user.name "Viktoriaipp"

C:\Users\Viktoriaipp\Documents\Vika>git config --global user.email "kornvik@yandex.ru"

C:\Users\Viktoriaipp\Documents\Vika>git init
Initialized empty Git repository in C:/Users/Viktoriaipp/Documents/Vika/.git/
```

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\Viktoriaipp\Documents\Vika>git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
C:\Users\Viktoriaipp\Documents\Vika>
```

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\Viktoriaipp\Documents\Vika>git add .

C:\Users\Viktoriaipp\Documents\Vika>git status
On branch master

No commits yet

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

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\Viktoriaipp\Documents\Vika>git commit -m "one"
[master (root-commit) 2f9be0b] one
1 file changed, 1 insertion(+)
create mode 100644 vika.txt

C:\Users\Viktoriaipp\Documents\Vika>git status
On branch master
nothing to commit, working tree clean
```

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\Viktoriaipp\Documents\Vika>git remote add origin https://github.com/Viktoriaipp/LR1.git

C:\Users\Viktoriaipp\Documents\Vika>git remote -v
origin  https://github.com/Viktoriaipp/LR1.git (fetch)
origin  https://github.com/Viktoriaipp/LR1.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 репозиторий 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.

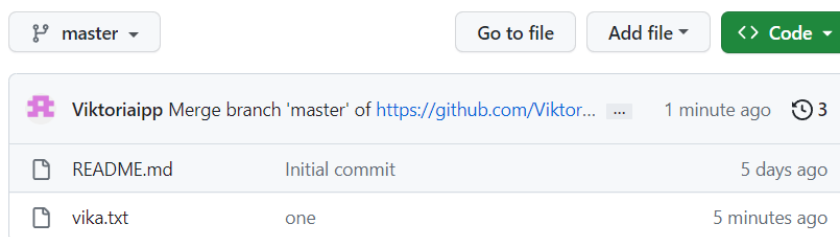
Мы сделали изменения в нашем репозитории, теперь другие пользователи могут скачать изменения при помощи команды `pull`.

`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).

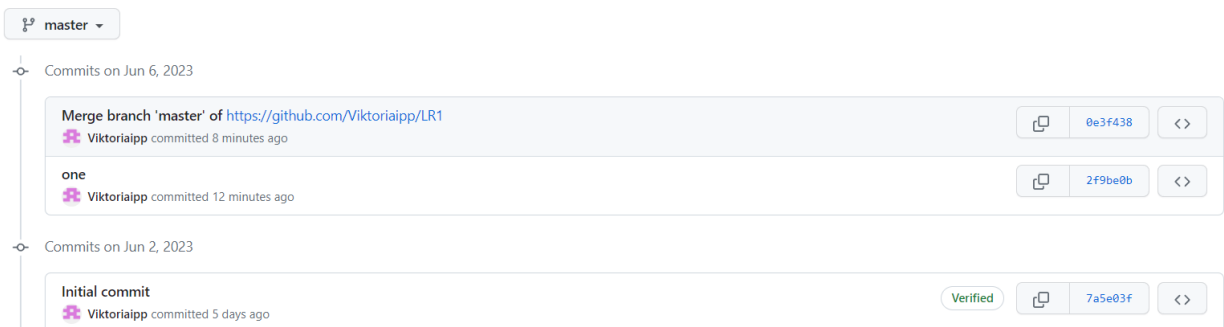
```
C:\Users\Viktoriaipp\Documents\Vika>git push origin master
info: please complete authentication in your browser...
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 514 bytes | 171.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Viktoriaipp/LR1.git
7a5e03f..0e3f438 master -> master
```

Видим, что изменения отобразились и на нашем сервере.



Если нажмем на

«commit», то увидим все наши изменения.



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\Viktoriaipp\Documents\Vika>git branch second

C:\Users\Viktoriaipp\Documents\Vika>git branch
* master
  second

C:\Users\Viktoriaipp\Documents\Vika>git checkout second
Switched to branch 'second'
```




Создаем в нашем локальном репозитории `vika2.txt` и напишем там свою фамилию, затем добавляем в области подготовленных файлов, закомитем и отправим на сервер.


```
C:\Users\Viktoriaipp\Documents\Vika>git add .


C:\Users\Viktoriaipp\Documents\Vika>git status
On branch second
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   vika2.txt



C:\Users\Viktoriaipp\Documents\Vika>git push origin second
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), 297 bytes | 297.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Viktoriaipp/LR1.git
   0e3f438..9d8ee20  second -> second
```

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

 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 ×

 Viktoriaipp Merge branch 'master' of <https://github.com/Viktoriaipp/LR1> 0e3f438 35 minutes ago 🕒 3 commits

 README.md	Initial commit	5 days ago
 vika.txt	one	39 minutes ago

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

second 2 branches 0 tags Go to file Add file Code

This branch is 1 commit ahead of master. Contribute

Viktoriaipp two 9d8ee20 3 minutes ago 4 commits

README.md	Initial commit	5 days ago
vika.txt	one	40 minutes ago
vika2.txt	two	3 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\Viktoriaipp\Documents\Vika>git checkout master
Switched to branch 'master'
```

тот компьютер > Документы > Vika

Поиск в: Vika

Имя	Дата изменения	Тип	Размер
.git	06.06.2023 14:24	Папка с файлами	
README.md	06.06.2023 13:48	Файл "MD"	1 КБ
vika.txt	06.06.2023 13:14	Текстовый докум...	1 КБ

Чтобы он появился, нужно воспользоваться merge для объединения веток.

```
C:\Users\Viktoriaipp\Documents\Vika>git merge second
Updating 0e3f438..9d8ee20
Fast-forward
 vika2.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 vika2.txt
```

Теперь удалим ветку second.

```
C:\Users\Viktoriaipp\Documents\Vika>git branch -d second
Deleted branch second (was 9d8ee20).
```

Создадим ветку, в которой будут одинаковые файлы.

```
C:\Users\Viktoriaipp\Documents\Vika>git branch newdew
C:\Users\Viktoriaipp\Documents\Vika>git branch
* master
  newdew
```

Переключимся на неё

```
C:\Users\Viktoriaipp\Documents\Vika>git checkout newdew
Switched to branch 'newdew'
```

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


```

C:\Users\Viktoriaipp\Documents\Vika>git add .

C:\Users\Viktoriaipp\Documents\Vika>git status
On branch newdew
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   vika.txt

C:\Users\Viktoriaipp\Documents\Vika>git commit -m "three"
[newdew a5091d0] three
1 file changed, 1 insertion(+), 1 deletion(-)

```

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

Теперь объединим ветки

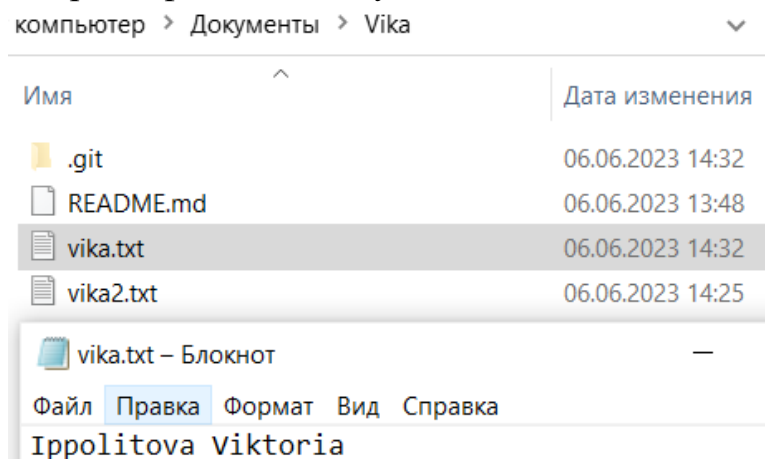
```

C:\Users\Viktoriaipp\Documents\Vika>git checkout master
Switched to branch 'master'

C:\Users\Viktoriaipp\Documents\Vika>git merge newdew
Updating 9d8ee20..a5091d0
Fast-forward
 vika.txt | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

```

Откроем файл снова и увидим изменения



Откроем файл снова, там есть и имя, и отчество

Переключимся на ветку newdev и удалим в имени несколько букв

Зафиксируем, закоммитим

Переключимся на ветку master, добавим к отчеству несколько букв

Зафиксируем, закоммитим

Объединим ветки

```

C:\Users\Viktoriaipp\Documents\Vika>git add .

C:\Users\Viktoriaipp\Documents\Vika>git status
On branch newdew
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   vika.txt

C:\Users\Viktoriaipp\Documents\Vika>git commit -m "four"
[newdew 20c1e6d] four
1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\Viktoriaipp\Documents\Vika>git checkout master
Switched to branch 'master'

C:\Users\Viktoriaipp\Documents\Vika>git commit -m "five"
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:   vika.txt

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

C:\Users\Viktoriaipp\Documents\Vika>git add .

C:\Users\Viktoriaipp\Documents\Vika>git commit -m "five"
[master 75f651e] five
1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\Viktoriaipp\Documents\Vika>_

```

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

Разрешим конфликт.

```

C:\Users\Viktoriaipp\Documents\Vika>git diff
diff --cc vika.txt
index faac2d0,0589652..0000000
--- a/vika.txt
+++ b/vika.txt
@@@ -1,1 -1,1 +1,5 @@@
- Ippolitova Viktoriaeeee
- Ippolitova Vik
++<<<<<< HEAD
++Ippolitova Viktoriaeeee
++=====
++Ippolitova Vik
++>>>>>> newdew

```

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

```
C:\Users\Viktoriaipp\Documents\Vika>git commit -m "six"
[master c6d2943] six

C:\Users\Viktoriaipp\Documents\Vika>git push origin master
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (12/12), 1.11 KiB | 189.00 KiB/s, done.
Total 12 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Viktoriaipp/LR1.git
0e3f438..c6d2943  master -> master
```

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

second 2 branches 0 tags

Go to file Add file <> Code

Switch branches/tags

Find or create a branch...

Branches Tags

master default

✓ second

View all branches

9d8ee20 27 minutes ago 4 commits

Initial commit 5 days ago

one 1 hour ago

Удалим их.

```
C:\Users\Viktoriaipp\Documents\Vika>git push origin --delete second
To https://github.com/Viktoriaipp/LR1.git
- [deleted] second
```

Посмотрим изменения.

master 1 branch 0 tags

Go to file Add file <> Code

Switch branches/tags

Find or create a branch...

Branches Tags

master default

✓ master

View all branches

c6d2943 7 minutes ago 8 commits

Initial commit 5 days ago

six 7 minutes ago

two 29 minutes ago

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

```
C:\Users\Viktoriaipp\Documents\Vika>git log
commit c6d29436f8554ce0325b8a426aeda1ac3946c75d (HEAD -> master, origin/master)
Merge: 75f651e 20c1e6d
Author: Viktoriaipp <kornyvik@yandex.ru>
Date: Tue Jun 6 14:41:42 2023 +0300

    six

commit 75f651eb0c81c805300b8d9e5193ab6006b046f3
Author: Viktoriaipp <kornyvik@yandex.ru>
Date: Tue Jun 6 14:38:12 2023 +0300

    five

commit 20c1e6d7361d340f401c7611bb1bdf4bde0a1c91
Author: Viktoriaipp <kornyvik@yandex.ru>
Date: Tue Jun 6 14:37:20 2023 +0300

    four

commit a5091d0421edfc099427a7832f9fa87aa1511c65
Author: Viktoriaipp <kornyvik@yandex.ru>
Date: Tue Jun 6 14:30:40 2023 +0300

    three

commit 9d8ee20d4620beac64a32e10eb8b075d6f765ca6
Author: Viktoriaipp <kornyvik@yandex.ru>
Date: Tue Jun 6 14:20:06 2023 +0300

    two

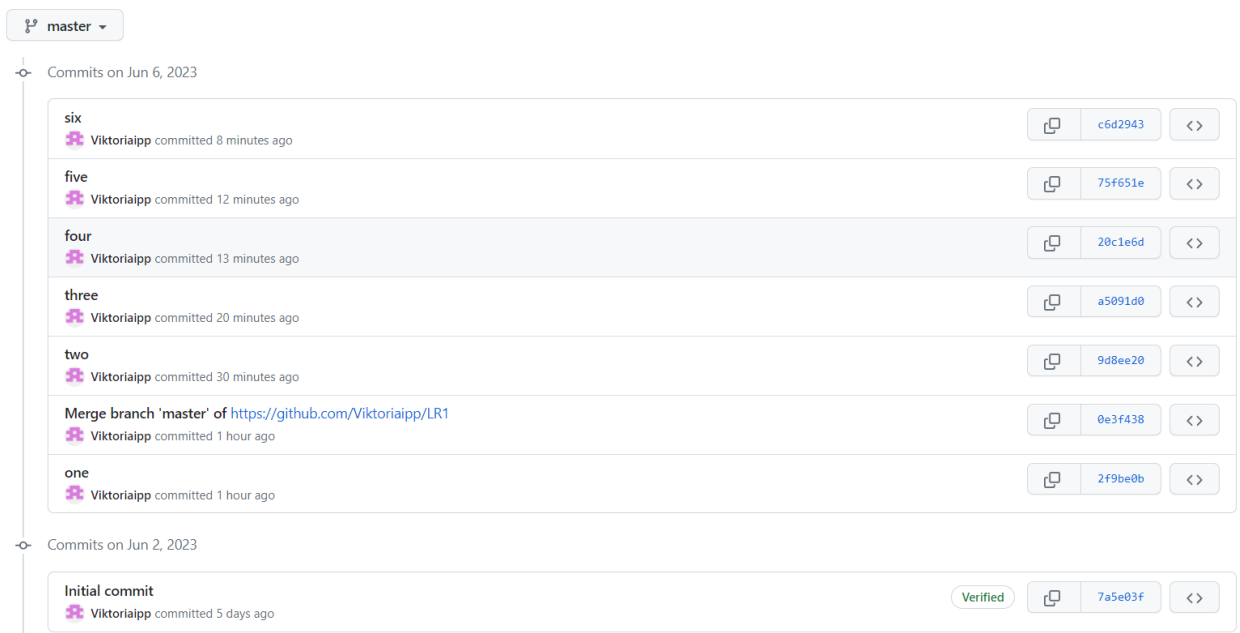
commit 0e3f4386c4308e5c5f7cc1f8f45363d5b860241d
Merge: 2f9be0b 7a5e03f
Author: Viktoriaipp <kornyvik@yandex.ru>
Date: Tue Jun 6 13:48:07 2023 +0300

    Merge branch 'master' of https://github.com/Viktoriaipp/LR1

commit 2f9be0b45383eace51fbc26007a255de54be281e
Author: Viktoriaipp <kornyvik@yandex.ru>
Date: Tue Jun 6 13:43:54 2023 +0300

    one
```

Или можно использовать github.



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

Git

```
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 0e3f438 Merge branch 'master' of https://github.com/Viktoriaipp/LR1

C:\Users\Viktoriaipp\Documents\Vika>git branch
* (HEAD detached at 0e3f438)
  master

C:\Users\Viktoriaipp\Documents\Vika>git checkout master
Previous HEAD position was 0e3f438 Merge branch 'master' of https://github.com/Viktoriaipp/LR1
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\Viktoriaipp\Documents\Vika>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:   vika.txt

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

C:\Users\Viktoriaipp\Documents\Vika>git add .

C:\Users\Viktoriaipp\Documents\Vika>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   vika.txt

C:\Users\Viktoriaipp\Documents\Vika>git reset HEAD
Unstaged changes after reset:
M       vika.txt

C:\Users\Viktoriaipp\Documents\Vika>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:   vika.txt

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

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\Viktoriaipp\Documents\Vika>git add .

C:\Users\Viktoriaipp\Documents\Vika>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   vika.txt

C:\Users\Viktoriaipp\Documents\Vika>git commit -m "last"
[master 4d8db3d] last
1 file changed, 2 insertions(+), 1 deletion(-)

C:\Users\Viktoriaipp\Documents\Vika>git reset HEAD

C:\Users\Viktoriaipp\Documents\Vika>git status
On branch master
nothing to commit, working tree clean

```

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

Откроется консоль в редакторе Vi.

```

C:\Windows\System32\cmd.exe - git commit --amend

```

```

commlast_

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Tue Jun 6 14:59:15 2023 +0300
#
# On branch master
# Changes to be committed:
#       modified:   vika.txt
#

```

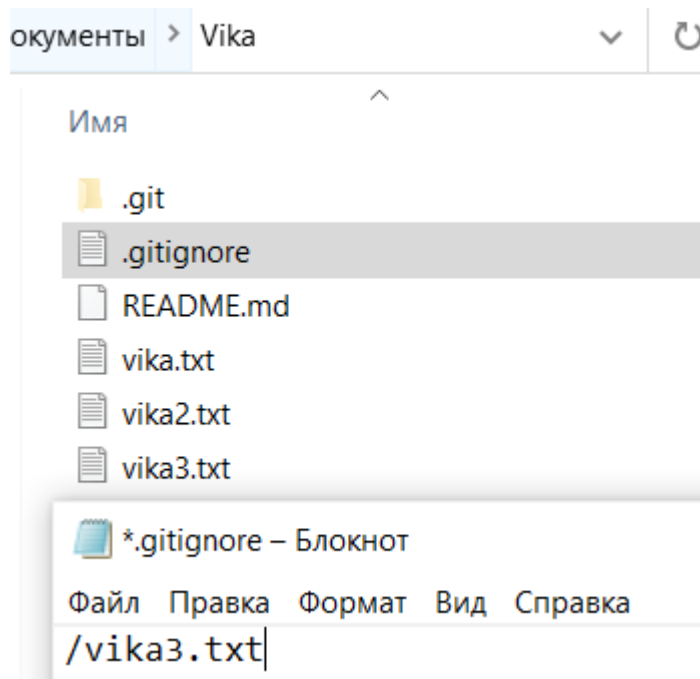
```

C:\Users\Viktoriaipp\Documents\Vika>git commit --amend
[master 3988d85] last
Date: Tue Jun 6 14:59:15 2023 +0300
1 file changed, 2 insertions(+), 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.



```
C:\Users\Viktoriaipp\Documents\Vika>git add .

C:\Users\Viktoriaipp\Documents\Vika>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   .gitignore

C:\Users\Viktoriaipp\Documents\Vika>git commit -m "last_commit"
[master 68e00e1] last_commit
 1 file changed, 1 insertion(+)
 create mode 100644 .gitignore

C:\Users\Viktoriaipp\Documents\Vika>git push origin master
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 548 bytes | 548.00 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/Viktoriaipp/LR1.git
 c6d2943..68e00e1  master -> master
```


Viktoriaipp / LR1Public

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

master

1 branch

0 tags

Go to file

Add file

<> Code

Viktoriaipp last_commit68e00e11 minute ago🕒 10 commits

.gitignore	last_commit	1 minute ago
README.md	Initial commit	5 days ago
vika.txt	last	8 minutes ago
vika2.txt	two	50 minutes ago

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