

ДИСЦИПЛИНА	ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНЫХ ПРИЛОЖЕНИЙ
	(полное наименование дисциплины без сокращений)
ИНСТИТУТ	ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
КАФЕДРА	МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ И СТАНДАРТИЗАЦИИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
	(полное наименование кафедры)
ВИД УЧЕБНОГО МАТЕРИАЛА	МАТЕРИАЛЫ ДЛЯ ПРАКТИЧЕСКИХ/СЕМИНАРСКИХ ЗАНЯТИЙ
	(в соответствии с пп.1-11)
ПРЕПОДАВАТЕЛЬ	ЖМАТОВ ДМИТРИЙ ВЛАДИМИРОВИЧ, МЕЛЬНИКОВ ДЕНИС АЛЕКСАНДРОВИЧ, МИРОНОВ АЛЕКСЕЙ ИГОРЕВИЧ, ОВЧИННИКОВ МИХАИЛ АНДРЕЕВИЧ, ПЕТРЕНКО АЛЕКСАНДР АНАТОЛЬЕВИЧ, ПЕТРОВА АННА АЛЕКСАНДРОВНА, СНЕГИРЕВ ДМИТРИЙ АЛЕКСЕЕВИЧ, ТУМАНОВА МАРИНА БОРИСОВНА, ХЛЕБНИКОВА ВАЛЕРИЯ ЛЕОНИДОВНА, ЧЕРНОВ ЕВГЕНИЙ АЛЕКСАНДРОВИЧ
	(фамилия, имя, отчество)
СЕМЕСТР	4, 2023-2024
	(указать семестр обучения, учебный год)

Часть 1. Основные команды Git

1. Установите и настройте клиент Git на своей рабочей станции.






	Git Bash	08.12.2023 17:33	Ярлык	2 КБ
	Git CMD	08.12.2023 17:33	Ярлык	2 КБ
	Git FAQs (Frequently Asked Questions)	08.12.2023 17:33	Ярлык Интернета	1 КБ
	Git GUI	08.12.2023 17:33	Ярлык	2 КБ
	Git Release Notes	08.12.2023 17:33	Ярлык	2 КБ

Рисунок 1 – установленный Git на рабочей станции

2. Настройка Git

```
PS C:\Users\Rarmash\Desktop\prac1> git config --global user.name "Rarmash"
PS C:\Users\Rarmash\Desktop\prac1> git config --global user.email "andrey2004.andrey2004@outlook.com"
PS C:\Users\Rarmash\Desktop\prac1> git config --global core.autocrlf true
PS C:\Users\Rarmash\Desktop\prac1> git config --global core.safecrlf warn
PS C:\Users\Rarmash\Desktop\prac1> git config --global core.quotePath off
PS C:\Users\Rarmash\Desktop\prac1> git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
user.name=Rarmash
user.email=andrey2004.andrey2004@outlook.com
core.autocrlf=true
core.safecrlf=warn
core.quotePath=off
```

Рисунок 2 – Указание логина и почты для настройки Git

3. Начало работы – создание папки и файлов

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop
$ mkdir prac1

Rarmash@RARMASH-PC MINGW64 ~/Desktop
$ cd prac1

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1
$ touch projekt.html

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1
$ nano projekt.html

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1
$
```

Рисунок 3 – Создание директории и внесение изменений в файлы директории

4. Создание репозитория

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1
$ git init
Initialized empty Git repository in C:/Users/Rarmash/Desktop/prac1/.git/
```

Рисунок 4 – Инициализация репозитория

5. Добавление файла в репозиторий и добавление первого коммита

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git add proekt.html
warning: in the working copy of 'proekt.html', LF will be replaced by CRLF the n
ext time Git touches it

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git commit -m "Initial commit"
[master (root-commit) 18a0e0b] Initial commit
1 file changed, 1 insertion(+)
create mode 100644 proekt.html

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git status
On branch master
nothing to commit, working tree clean
```

Рисунок 5 – Сделанный коммит с комментарием к нему с последующей проверкой состояния ветки

6. Индексация изменений

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ nano proekt.html

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ 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:   proekt.html

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

Рисунок 6 – Внесение изменений в файл

7. Коммиты нескольких изменений

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ nano proekt.html

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git add proekt.html
warning: in the working copy of 'proekt.html', LF will be replaced by CRLF the n
ext time Git touches it

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ nano proekt.html

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   proekt.html

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:   proekt.html

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git commit -m "Added standart HTML page tags"
[master 29762a8] Added standart HTML page tags
1 file changed, 6 insertions(+), 1 deletion(-)

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ 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:   proekt.html

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

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git add .
warning: in the working copy of 'proekt.html', LF will be replaced by CRLF the n
ext time Git touches it

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   proekt.html

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git commit -m "Added HTML header"
[master 86ab874] Added HTML header
1 file changed, 2 insertions(+), 1 deletion(-)
```

Рисунок 7 – создание двух коммитов с изменениями файла proekt.html

8. Просмотр истории коммитов

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git log
commit 86ab8744953c01c27c55f09f80bcecf223764aaf (HEAD -> master)
Author: Rarmash <andrey2004.andrey2004@outlook.com>
Date: Sat Feb 10 21:07:11 2024 +0300

    Added HTML header

commit 29762a82e69e1695ffb1c535b40c7f117bff24a1
Author: Rarmash <andrey2004.andrey2004@outlook.com>
Date: Sat Feb 10 21:05:46 2024 +0300

    Added standart HTML page tags

commit 18a0e0b733bc75efab271558f1067c82132a102b
Author: Rarmash <andrey2004.andrey2004@outlook.com>
Date: Sat Feb 10 21:02:39 2024 +0300

    Initial commit

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git log --pretty=oneline
86ab8744953c01c27c55f09f80bcecf223764aaf (HEAD -> master) Added HTML header
29762a82e69e1695ffb1c535b40c7f117bff24a1 Added standart HTML page tags
18a0e0b733bc75efab271558f1067c82132a102b Initial commit

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git log --pretty=oneline --max-count=2
86ab8744953c01c27c55f09f80bcecf223764aaf (HEAD -> master) Added HTML header
29762a82e69e1695ffb1c535b40c7f117bff24a1 Added standart HTML page tags

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git log --pretty=oneline --since='5 minutes ago'
86ab8744953c01c27c55f09f80bcecf223764aaf (HEAD -> master) Added HTML header
29762a82e69e1695ffb1c535b40c7f117bff24a1 Added standart HTML page tags

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git log --pretty=oneline --until='5 minutes ago'
18a0e0b733bc75efab271558f1067c82132a102b Initial commit

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git log --pretty=oneline --author=Rarmash
86ab8744953c01c27c55f09f80bcecf223764aaf (HEAD -> master) Added HTML header
29762a82e69e1695ffb1c535b40c7f117bff24a1 Added standart HTML page tags
18a0e0b733bc75efab271558f1067c82132a102b Initial commit

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git log --pretty=oneline --all
86ab8744953c01c27c55f09f80bcecf223764aaf (HEAD -> master) Added HTML header
29762a82e69e1695ffb1c535b40c7f117bff24a1 Added standart HTML page tags
18a0e0b733bc75efab271558f1067c82132a102b Initial commit

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git log --pretty=format:"%h %ad | %s%d [%an]" --graph --date=short
* 86ab874 2024-02-10 | Added HTML header (HEAD -> master) [Rarmash]
* 29762a8 2024-02-10 | Added standart HTML page tags [Rarmash]
* 18a0e0b 2024-02-10 | Initial commit [Rarmash]
```

Рисунок 8 – вывод истории коммитов с различными параметрами

9. Получение старых версий

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git log --pretty=format:"%h %ad | %s%d [%an]" --graph --date=short
* 86ab874 2024-02-10 | Added HTML header (HEAD -> master) [Rarmash]
* 29762a8 2024-02-10 | Added standart HTML page tags [Rarmash]
* 18a0e0b 2024-02-10 | Initial commit [Rarmash]

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git checkout 18a0e0b
Note: switching to '18a0e0b'.

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 18a0e0b Initial commit

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 ((18a0e0b...))
$ cat projekt.html
123

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 ((18a0e0b...))
$ git checkout master
Previous HEAD position was 18a0e0b Initial commit
Switched to branch 'master'

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ cat projekt.html
<html>
<head>
</head>
<body>
<h1>Hello, World!</h1>
</body>
</html>
```

Рисунок 9 – получение хеша коммита с последующим переключением на него и чтением файла projekt.html. Затем переключение на ветку master и чтение файла из неё.

10. Отмена локальных изменений (до индексации)

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git checkout master
Already on 'master'

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ nano projekt.html

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ 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:   projekt.html

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

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git checkout projekt.html
Updated 1 path from the index

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git status
On branch master
nothing to commit, working tree clean

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ cat projekt.html
<html>
<head>
</head>
<body>
<h1>Hello, World!</h1>
</body>
</html>
```

Рисунок 10 – отмена локальных изменений до индексации

11. Отмена локальных изменений (после индексации и до коммита)

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ nano projekt.html

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git add projekt.html

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   projekt.html

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git reset HEAD projekt.html
Unstaged changes after reset:
M       projekt.html

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git checkout projekt.html
Updated 1 path from the index

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git status
On branch master
nothing to commit, working tree clean
```

Рисунок 11 – отмена локальных изменений после индексации и до коммита

12. Отмена коммита.

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ nano projekt.html

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git add projekt.html

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git commit -m "Commit to be removed"
[master 532304f] Commit to be removed
1 file changed, 1 insertion(+)

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git config --global core.editor nano

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git revert HEAD
[master 2dbdee1] Revert "Commit to be removed"
1 file changed, 1 deletion(-)

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1 (master)
$ git log --pretty=format:"%h %ad | %s%d [%an]" --graph --date=short
* 2dbdee1 2024-02-10 | Revert "Commit to be removed" (HEAD -> master) [Rarmash]
* 532304f 2024-02-10 | Commit to be removed [Rarmash]
* 86ab874 2024-02-10 | Added HTML header [Rarmash]
* 29762a8 2024-02-10 | Added standart HTML page tags [Rarmash]
* 18a0e0b 2024-02-10 | Initial commit [Rarmash]
```

Рисунок 12 – отмена коммита

Часть 2. Системы управления репозиториями

1. Создание аккаунта на GitHub

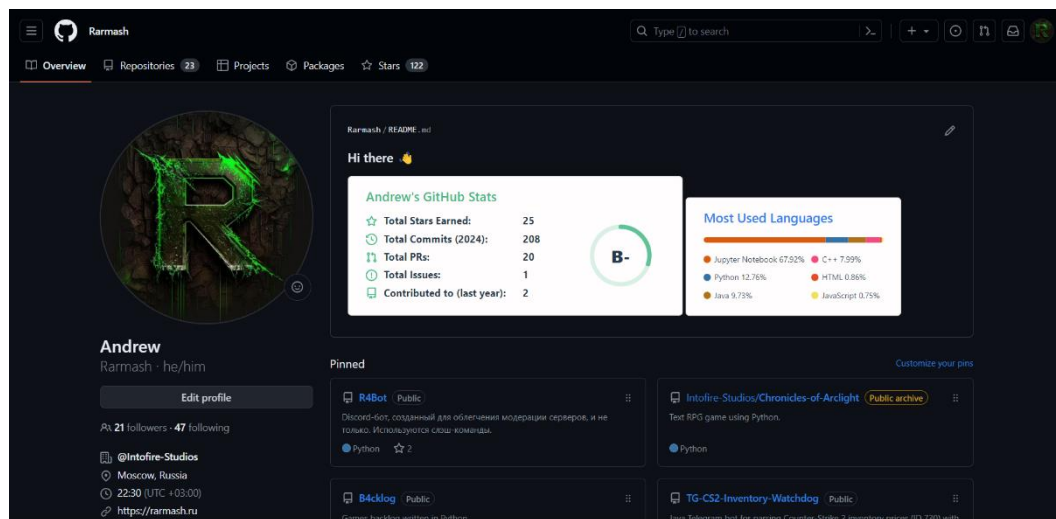


Рисунок 13 – созданный профиль на GitHub

2. Создание SSH-ключа для авторизации

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop
$ cd ~/.ssh

Rarmash@RARMASH-PC MINGW64 ~/.ssh
$ ssh-keygen -t rsa -b 4096 -C "andrey2004.andrey2004@outlook.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Rarmash/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Rarmash/.ssh/id_rsa
Your public key has been saved in /c/Users/Rarmash/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:8DzLqYxGCikwoTpTyofAgPufy7Vks03Lm6K4EGha4mE andrey2004.andrey2004@outlook.com
The key's randomart image is:
+---[RSA 4096]-----+
|Bo
|B
|==.
|Oo. +
|OE.. S
|=*+ . . +
|o. o . =
| . o+*.O o
| oo=0=. *.
+---[SHA256]-----+
```

Рисунок 14 – генерация SSH-ключа на рабочей машине


```
Rarmash@RARMASH-PC MINGW64 ~/.ssh
$ ssh-agent
SSH_AUTH_SOCK=/tmp/ssh-mLb8tN13Hm2/agent.612; export SSH_AUTH_SOCK;
SSH_AGENT_PID=613; export SSH_AGENT_PID;
echo Agent pid 613;
```

Рисунок 15 – добавление ключа в ssh-agent

```
Rarmash@RARMASH-PC MINGW64 ~/.ssh
$ eval "$(ssh-agent -s)"
Agent pid 618
```

Рисунок 16 – проверка доступности ключа

```
Rarmash@RARMASH-PC MINGW64 ~/.ssh
$ ssh-add ~/.ssh/id_rsa
Identity added: /c:/Users/Rarmash/.ssh/id_rsa (andrey2004.andrey2004@outlook.com)

Rarmash@RARMASH-PC MINGW64 ~/.ssh
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQC1Gt04CXiEp1jnDPqSqqk1YvAG6zdEYp+AhGvqAeBG
jt+tNpt4r6x1X0mvK1oc6eHB9zpLu3y0vVWRGyg8w/rVReU3ldTABuiEE25TmvJG7zdNQ8aaTw8Yc7rv
bfcG26AcA1cMG01veh0Nn9PUW3gWqfcta/RG0zLStdixBEVaSn718kc0sWsWqoE7JhZwa/gH18zN1vrr3vM
bJCKZOC87R6u0oRRhn9PUW3gWqfcta/RG0zLStdixBEVaSn718kc0sWsWqoE7JhZwa/gH18zN1vrr3vM
GAg8NL3gvGpg+9Cqsp18yUpq2z1DqffiHdRB0Wmdweqz762n31Mzfz8hyL1nvtbm8g1E8Irv97P6H81
Q/0Gxc0jPT5F0tG2TkW1Ju53/mH1ZcUtoK1+A121mFs6e1ZIDcgidkOdY2gMhRdcNzvjbXGUN1EGAiYi
yQ7Nbct9nQIGhDyFF7bf7/uTWFCwc20AkUvn4k8rjARAgLr7FwkB34149EcuKBo1X9kNdQEnioDP8jH
XS97nMwU1FQryv9811p0PF2Ubmgd6Jyq0Q9WRDEQ0hQKhAqb9VGhWxMp2RTq6qZTWtq16IX5Bpj1zjQ
CaJwo9m2ZGZqFXU4tqbMt1QzZK9EmcJkCxtv7QBGRO8ivgZREH89Q6SkOX3DjRJuLAVjksGVsgGdCEv
VQ== andrey2004.andrey2004@outlook.com
```

Рисунок 17 – Добавление ключа в систему и получение его содержимого

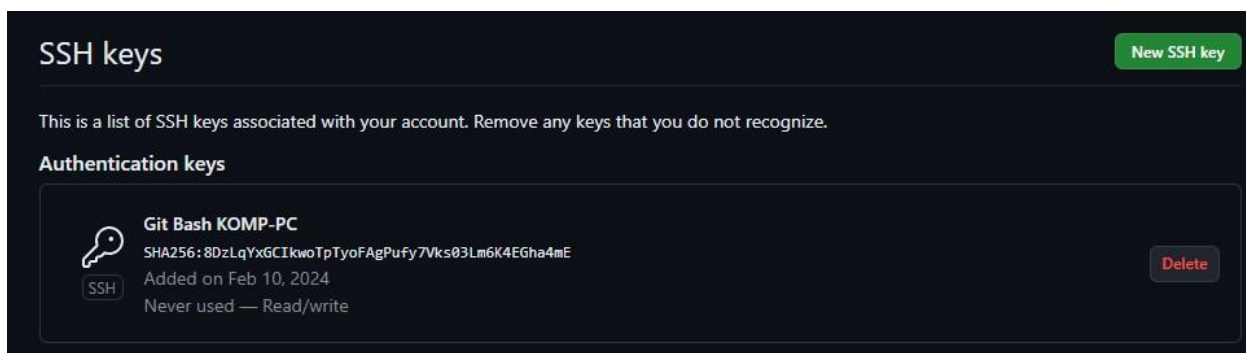


Рисунок 18 – Добавление ключа на Github

3. Создание нового репозитория для своего проекта

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop
$ mkdir prac1_1; cd prac1_1

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1_1
$ touch readme.txt

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1_1
$ nano readme.txt

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1_1
$ touch helloworld.py; nano helloworld.py

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1_1
$ touch aplusb.py; nano aplusb.py

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1_1
$ git init
Initialized empty Git repository in C:/Users/Rarmash/Desktop/prac1_1/.git/
```

Рисунок 19 – создание репозитория с несколькими файлами

4. Связываем локальный и удаленный репозитории

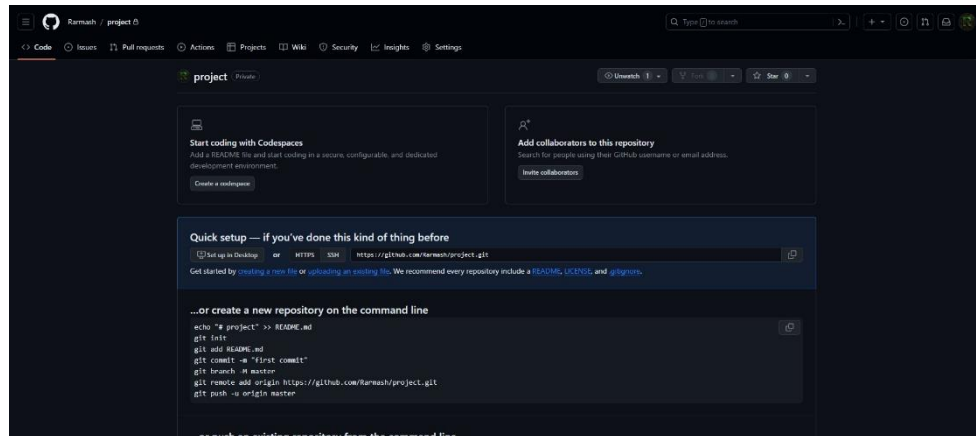


Рисунок 20 – созданный репозиторий на Github



Рисунок 21 – связывание локального репозитория с удалённым

5. Создание веток и переключение между ними

```
Rarmash@RARMASH-LAPTOP MINGW64 ~/Desktop/prac1_1 (master)
$ git checkout -b dev
Switched to a new branch 'dev'

Rarmash@RARMASH-LAPTOP MINGW64 ~/Desktop/prac1_1 (dev)
$ touch 123.py; nano 123.py

Rarmash@RARMASH-LAPTOP MINGW64 ~/Desktop/prac1_1 (dev)
$ git add 123.py
warning: in the working copy of '123.py', LF will be replaced by CRLF the next time Git touches it

Rarmash@RARMASH-LAPTOP MINGW64 ~/Desktop/prac1_1 (dev)
$ git status
On branch dev

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   123.py

Rarmash@RARMASH-LAPTOP MINGW64 ~/Desktop/prac1_1 (dev)
$ git commit -m "Initial commit"
[dev (root-commit) 07f6a7f] Initial commit
1 file changed, 1 insertion(+)
create mode 100644 123.py
```

Рисунок 22 – работа в новой ветке

6. Слияние веток

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1_1 (dev)
$ git checkout master
Switched to branch 'master'

Rarmash@RARMASH-PC MINGW64 ~/Desktop/prac1_1 (master)
$ git merge dev
Updating 7da5302..4c312dd
Fast-forward
 123.py | 3 +++
1 file changed, 3 insertions(+)
```

Рисунок 23 – слияние веток master и dev

7. Выполните цепочку действий в репозитории, согласно вариантам.

Вариант 3.

1) Клонировать непустой удаленный репозиторий на локальную машину

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop
$ git clone git@github.com:Rarmash/MIREA-Timetable.git; cd MIREA-Timetable
Cloning into 'MIREA-Timetable'...
remote: Enumerating objects: 28, done.
remote: Counting objects: 100% (28/28), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 28 (delta 12), reused 21 (delta 8), pack-reused 0
Receiving objects: 100% (28/28), 6.91 KiB | 6.91 MiB/s, done.
Resolving deltas: 100% (12/12), done.
```

Рисунок 24 – клонирование непустого удаленного репозитория

2) Создайте новую ветку и выведите список всех веток

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/MIREA-Timetable (master)
$ git checkout -b dev
Switched to a new branch 'dev'

Rarmash@RARMASH-PC MINGW64 ~/Desktop/MIREA-Timetable (dev)
$ git branch -a
* dev
master
remotes/origin/HEAD -> origin/master
remotes/origin/master
```

Рисунок 25 – создание новой ветки и вывод списка всех веток

3) Произведите 3 коммита в новой ветке в разные файлы

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/MIREA-Timetable (dev)
$ git add main.py

Rarmash@RARMASH-PC MINGW64 ~/Desktop/MIREA-Timetable (dev)
$ git commit -m "Edited main.py"
[dev 21d6409] Edited main.py
1 file changed, 1 insertion(+), 1 deletion(-)

Rarmash@RARMASH-PC MINGW64 ~/Desktop/MIREA-Timetable (dev)
$ git add options.py

Rarmash@RARMASH-PC MINGW64 ~/Desktop/MIREA-Timetable (dev)
$ git commit -m "Edited options.py"
[dev a390820] Edited options.py
1 file changed, 1 insertion(+), 1 deletion(-)

Rarmash@RARMASH-PC MINGW64 ~/Desktop/MIREA-Timetable (dev)
$ git add .env_template

Rarmash@RARMASH-PC MINGW64 ~/Desktop/MIREA-Timetable (dev)
$ git commit -m "Edited .env_template"
[dev cb4336b] Edited .env_template
1 file changed, 1 insertion(+), 1 deletion(-)
```

Рисунок 26 – 3 коммита в новой ветке в разные файлы

4) Выгрузите изменения в удаленный репозиторий

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/MIREA-Timetable (dev)
$ git push origin dev
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 12 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 814 bytes | 814.00 KiB/s, done.
Total 9 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 3 local objects.
remote:
remote: Create a pull request for 'dev' on GitHub by visiting:
remote:   https://github.com/Rarmash/MIREA-Timetable/pull/new/dev
remote:
To github.com:Rarmash/MIREA-Timetable.git
 * [new branch]   dev -> dev
```

Рисунок 27 – выгрузка изменений в удалённый репозиторий

5) Откатите в новой ветке предпоследний коммит (в том числе в удаленном репозитории)

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/MIREA-Timetable (dev)
$ git log --pretty=format:"%h %ad | %s%d [%an]" --graph --date=short
* cb4336b 2024-02-11 | Edited .env_template (HEAD -> dev, origin/dev) [Rarmash]
* a390820 2024-02-11 | Edited options.py [Rarmash]
* 21d6409 2024-02-11 | Edited main.py [Rarmash]
* bab98ee 2023-04-29 | Yesterday table (origin/master, origin/HEAD, master) [Andrew]
* a802bff 2023-04-29 | Separated funcs [Andrew]
* 1698c99 2023-04-28 | MongoDB update [Andrew]
* ba467bc 2023-04-25 | Initial commit [Andrew]
* f016795 2023-04-25 | Initial commit [Andrew]

Rarmash@RARMASH-PC MINGW64 ~/Desktop/MIREA-Timetable (dev)
$ git revert a390820
[dev b5587b8] Revert "Edited options.py"
1 file changed, 1 insertion(+), 1 deletion(-)
```

Рисунок 28 – откат предпоследнего коммита в локальном репозитории

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/MIREA-Timetable (dev)
$ git push --set-upstream origin dev

Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 340 bytes | 340.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:Rarmash/MIREA-Timetable.git
   cb4336b..b5587b8  dev -> dev
branch 'dev' set up to track 'origin/dev'.
```

Рисунок 29 – откат предпоследнего коммита в удалённом репозитории

6) Выведите в консоли различия между веткой master и новой веткой

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/MIREA-Timetable (dev)
$ git diff master dev
diff --git a/.env_template b/.env_template
index a892d84..2c0d48e 100644
--- a/.env_template
+++ b/.env_template
@@ -1,1 @@
-MONGODB=
+MONGODB=
\ No newline at end of file
diff --git a/main.py b/main.py
index b60db03..4a19c67 100644
--- a/main.py
+++ b/main.py
@@ -71,4 @@ def delete_group(user_id):
     Collection.delete_one({"_id": user_id})

if __name__ == "__main__":
-    app.run()
\ No newline at end of file
+    app.run()
```

Рисунок 30 – вывод различий между ветками master и dev

7) Слейте новую ветку с master при помощи merge

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/MIREA-Timetable (dev)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

Rarmash@RARMASH-PC MINGW64 ~/Desktop/MIREA-Timetable (master)
$ git merge dev
Updating bab98ee..b5587b8
Fast-forward
 .env_template | 2 +-
  main.py      | 2 +-
 2 files changed, 2 insertions(+), 2 deletions(-)
```

Рисунок 31 – слияние веток master и dev

Часть 3. Работа с ветвлением и оформление кода

1. Форк репозитория

5	https://github.com/pystardust/ytfzf
---	---

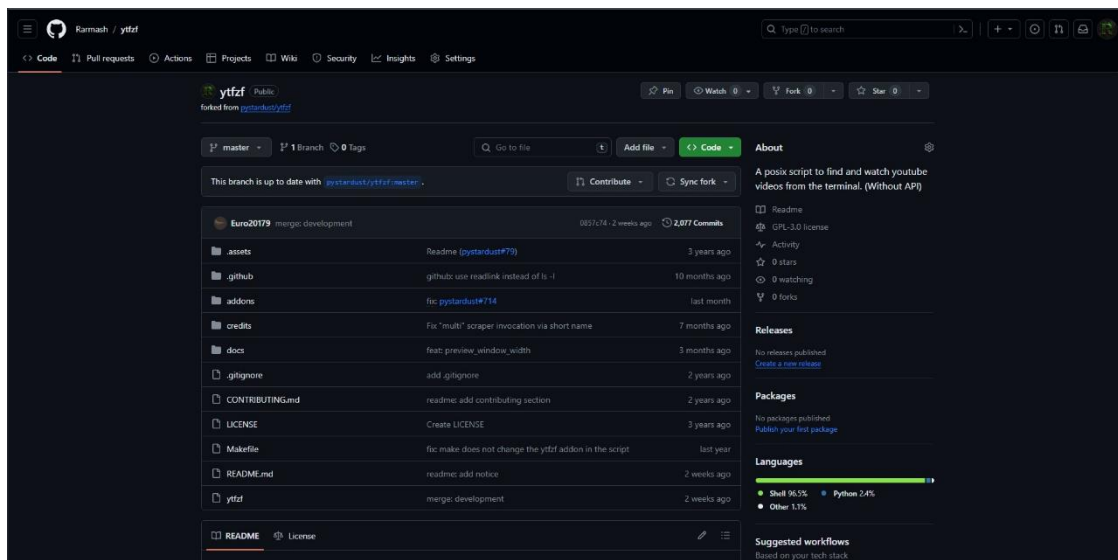


Рисунок 32 – сделанный форк репозитория

2. Клонирование репозитория на локальную машину

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop
$ git clone git@github.com:Rarmash/ytfzf.git
Cloning into 'ytfzf'...
remote: Enumerating objects: 7541, done.
remote: Counting objects: 100% (427/427), done.
remote: Compressing objects: 100% (185/185), done.
remote: Total 7541 (delta 251), reused 411 (delta 241), pack-reused 7114
Receiving objects: 100% (7541/7541), 4.78 MiB | 2.85 MiB/s, done.
Resolving deltas: 100% (4526/4526), done.

Rarmash@RARMASH-PC MINGW64 ~/Desktop
$ cd ytfzf

Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytfzf (master)
$
```

Рисунок 33 – клонирование репозитория на локальную машину

3. Создание двух веток в master'e

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytfzf (master)
$ git checkout -b branch1
Switched to a new branch 'branch1'

Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytfzf (branch1)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytfzf (master)
$ git checkout -b branch2
Switched to a new branch 'branch2'
```

Рисунок 34 – создание двух веток branch1 и branch2

4. 3 коммита в каждую из веток, которые меняют один и тот же кусочек файла

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytzfz (branch2)
$ git checkout branch1
Switched to branch 'branch1'

Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytzfz (branch1)
$ nano Makefile; git add Makefile; git commit -m "Edited Makefile"
[branch1 d9aa641] Edited Makefile
1 file changed, 1 insertion(+), 1 deletion(-)

Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytzfz (branch1)
$ nano README.md; git add README.md; git commit -m "Edited README.md"
[branch1 f045714] Edited README.md
1 file changed, 1 deletion(-)

Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytzfz (branch1)
$ nano ytfzf; git add ytfzf; git commit -m "Edited ytfzf"
[branch1 659b0ca] Edited ytfzf
1 file changed, 3 insertions(+), 3 deletions(-)

Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytzfz (branch1)
$ git checkout branch2
Switched to branch 'branch2'

Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytzfz (branch2)
$ nano Makefile; git add Makefile; git commit -m "Edited Makefile"
[branch2 be39437] Edited Makefile
1 file changed, 1 insertion(+), 1 deletion(-)

Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytzfz (branch2)
$ nano README.md; git add README.md; git commit -m "Edited README.md"
[branch2 97144fd] Edited README.md
1 file changed, 1 deletion(-)

Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytzfz (branch2)
$ nano ytfzf; git add ytfzf; git commit -m "Edited ytfzf"
[branch2 1a763d3] Edited ytfzf
1 file changed, 3 insertions(+), 3 deletions(-)
```

Рисунок 35 – создание 3-х коммитов в каждую из веток

5. Слияние ветки branch1 в ветку branch2 с разрешением конфликтов

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytzfz (branch2)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytzfz (master)
$ git merge branch1
Updating 0857c74..659b0ca
Fast-forward
 Makefile | 2 +-
 README.md | 1 -
 ytfzf     | 6 +++--
3 files changed, 4 insertions(+), 5 deletions(-)

Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytzfz (master)
$ git merge branch2
Auto-merging Makefile
CONFLICT (content): Merge conflict in Makefile
Auto-merging ytfzf
CONFLICT (content): Merge conflict in ytfzf
Automatic merge failed; fix conflicts and then commit the result.

Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytzfz (master|MERGING)
$ git checkout --theirs Makefile; git add Makefile
Updated 1 path from the index

Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytzfz (master|MERGING)
$ git checkout --theirs README.md; git add README.md
Updated 0 paths from the index

Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytzfz (master|MERGING)
$ git checkout --theirs ytfzf; git add ytfzf
Updated 1 path from the index

Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytzfz (master|MERGING)
$ git commit -m "added theirs"
[master 0d94cb7] added theirs
```

Рисунок 36 – слияние веток branch1 и branch2 с разрешением конфликтов

6. Выгрузка всех изменений во всех ветках в удаленный репозиторий

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytfzf (master)
$ git push origin master
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 12 threads
Compressing objects: 100% (18/18), done.
Writing objects: 100% (18/18), 1.68 KiB | 1.68 MiB/s, done.
Total 18 (delta 13), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (13/13), completed with 4 local objects.
To github.com:Rarmash/ytfzf.git
   0857c74..0d94cb7  master -> master

Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytfzf (master)
$ git checkout branch1
Switched to branch 'branch1'

Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytfzf (branch1)
$ git push origin branch1
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'branch1' on GitHub by visiting:
remote:   https://github.com/Rarmash/ytfzf/pull/new/branch1
remote:
To github.com:Rarmash/ytfzf.git
 * [new branch]      branch1 -> branch1

Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytfzf (branch1)
$ git checkout branch2
Switched to branch 'branch2'

Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytfzf (branch2)
$ git push origin branch2
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'branch2' on GitHub by visiting:
remote:   https://github.com/Rarmash/ytfzf/pull/new/branch2
remote:
To github.com:Rarmash/ytfzf.git
 * [new branch]      branch2 -> branch2
```

Рисунок 37 – выгрузка изменений во всех ветках в удаленный репозиторий

7. Выполнение 3-х коммитов в branch1

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytfzf (branch1)
$ nano CONTRIBUTING.md; git add CONTRIBUTING.md

Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytfzf (branch1)
$ git commit -m "Edited CONTRIBUTING.md"
[branch1 878f9b8] Edited CONTRIBUTING.md
1 file changed, 1 deletion(-)

Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytfzf (branch1)
$ nano LICENSE; git add LICENSE; git commit -m "Edited LICENSE"
[branch1 2f00538] Edited LICENSE
1 file changed, 1 insertion(+), 1 deletion(-)

Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytfzf (branch1)
$ nano .gitignore; git add .gitignore; git commit -m "Edited .gitignore"
[branch1 14f229d] Edited .gitignore
1 file changed, 1 insertion(+)
```

Рисунок 38 – Выполнение 3-х коммитов в branch1

8. Повторное клонирование репозитория в другую директорию

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytfzf (branch1)
$ cd ../; mkdir 123; cd 123

Rarmash@RARMASH-PC MINGW64 ~/Desktop/123
$ git clone git@github.com:Rarmash/ytfzf.git
Cloning into 'ytfzf'...
remote: Enumerating objects: 7559, done.
remote: Counting objects: 100% (445/445), done.
remote: Compressing objects: 100% (205/205), done.
remote: Total 7559 (delta 263), reused 413 (delta 239), pack-reused 7114
Receiving objects: 100% (7559/7559), 4.75 MiB | 5.54 MiB/s, done.
Resolving deltas: 100% (4537/4537), done.
```

Рисунок 39 – повторное клонирование репозитория в другую директорию

9. Выполнение 3-х коммитов в ветку branch1

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/123/ytfzf (master)
$ git checkout branch1
Switched to a new branch 'branch1'
branch 'branch1' set up to track 'origin/branch1'.

Rarmash@RARMASH-PC MINGW64 ~/Desktop/123/ytfzf (branch1)
$ cd credits

Rarmash@RARMASH-PC MINGW64 ~/Desktop/123/ytfzf/credits (branch1)
$ nano euro20179.md; git add euro20179.md; git commit -m "Edited euro20179.md"
[branch1 865fbe0] Edited euro20179.md
1 file changed, 1 deletion(-)

Rarmash@RARMASH-PC MINGW64 ~/Desktop/123/ytfzf/credits (branch1)
$ nano bsgalvan.md; git add bsgalvan.md; git commit -m "Edited bsgalvan.md"
[branch1 2ca867b] Edited bsgalvan.md
1 file changed, 1 deletion(-)

Rarmash@RARMASH-PC MINGW64 ~/Desktop/123/ytfzf/credits (branch1)
$ nano jac-zac.md; git add jac-zac.md; git commit -m "Edited jac-zac.md"
[branch1 46ff5b0] Edited jac-zac.md
1 file changed, 1 deletion(-)
```

Рисунок 40 – три коммита в ветке branch1

10. Выгрузка всех изменений в удаленный репозиторий

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/123/ytfzf/credits (branch1)
$ git push origin branch1
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 12 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (12/12), 1.08 KiB | 1.08 MiB/s, done.
Total 12 (delta 7), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (7/7), completed with 3 local objects.
To github.com:Rarmash/ytfzf.git
659b0ca..46ff5b0 branch1 -> branch1
```

Рисунок 41 – выгрузка всех изменений из локального репозитория в удаленный

11. Выгрузка изменения из старого репозитория с опцией –force

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/123/ytfzf/credits (branch1)
$ cd ../../../../ytfzf

Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytfzf (branch1)
$ git push origin branch1 --force
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 12 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 875 bytes | 875.00 KiB/s, done.
Total 9 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 3 local objects.
To github.com:Rarmash/ytfzf.git
+ 46ff5b0...14f229d branch1 -> branch1 (forced update)
```

Рисунок 42 – вернулись в старый клон с репозиторием и выгрузили изменения с опцией –force

12. Получение всех изменения в новом репозитории

```
Rarmash@RARMASH-PC MINGW64 ~/Desktop/ytfzf (branch1)
$ cd ../123/ytfzf

Rarmash@RARMASH-PC MINGW64 ~/Desktop/123/ytfzf (branch1)
$ git pull
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 9 (delta 5), reused 9 (delta 5), pack-reused 0
Unpacking objects: 100% (9/9), 855 bytes | 30.00 KiB/s, done.
From github.com:Rarmash/ytfzf
+ 46ff5b0...14f229d branch1 -> origin/branch1 (forced update)
Merge made by the 'ort' strategy.
 .gitignore      | 1 +
 CONTRIBUTING.md | 1 -
 LICENSE         | 2 +-
 3 files changed, 2 insertions(+), 2 deletions(-)
```

Рисунок 43 – перешли в новый репозиторий и получили в нём все изменения

Ответы на 8 любых вопросов:

9. Как отменить слияние веток, если произошел конфликт?

Отменить слияние веток при появлении конфликта можно командой **git merge --abort**.

10. Для чего нужен .gitignore?

.gitignore используется для указания файлов и директорий, которые не должны учитываться системой контроля версий Git. Это позволяет игнорировать временные файлы, конфигурационные файлы, и прочее, чтобы они не попадали в репозиторий.

11. Что делает команда git status?

Команда **git status** показывает текущий статус репозитория, включая измененные, добавленные и неотслеживаемые файлы.

12. Что делает команда git add?

Команда **git add** добавляет изменения файлов в индекс, подготавливая их к следующему коммиту.

13. Что делает команда git log?

Команда **git log** показывает историю коммитов. Она перечисляет коммиты, которые можно достичь, следуя связям родительских коммитов от указанных коммитов.

14. Что делает команда git diff?

Команда **git diff** показывает различия между рабочим деревом и индексом, а также между индексом и последним коммитом.

15. Что делает команда git show?

Команда **git show** отображает информацию о конкретном коммите, включая изменения, внесенные этим коммитом.

16. Что делает команда git stash?

Команда **git stash** временно сохраняет изменения в рабочей директории, чтобы можно было переключиться на другую ветку или выполнить другие операции.

Вывод

Мы освоили основные команды Git для управления локальным репозиторием: создание, изменение, индексирование, коммиты и отмена изменений. Также изучили работу с тегами для метки коммитов и использование систем управления репозиториями, таких как GitHub, в том числе создание, клонирование, связывание и выгрузка репозитория по SSH-ключу. Научились создавать, переключаться, сливать и удалять ветки, а также разрешать конфликты при слиянии. Познакомились с работой с ветками в удаленном репозитории и использованием опции --force для принудительной выгрузки изменений. Кроме того, мы провели форк репозитория и клонировали его для дальнейшей работы. Полученный опыт работы с Git и системами управления репозиториями будет полезен в будущих проектах.