



МИНОБРНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное учреждение  
высшего образования*

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

---

Институт Информационных технологий (ИТ)

Кафедра Математического обеспечения и стандартизации информационных  
технологий (МОСИТ)

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 1**

**по дисциплине**

**«Технология разработки программных приложений»**

**Тема: «Системы контроля версий»**

Выполнил студент группы ИКБО-25-22

Ракитин В.А.

Принял

Мельников Д. А.

Практическая работа выполнена

«\_\_»\_\_\_\_\_2024 г.

(подпись студента)

«Зачтено»

«\_\_»\_\_\_\_\_2024 г.

(подпись руководителя)

Москва 2024

## СОДЕРЖАНИЕ

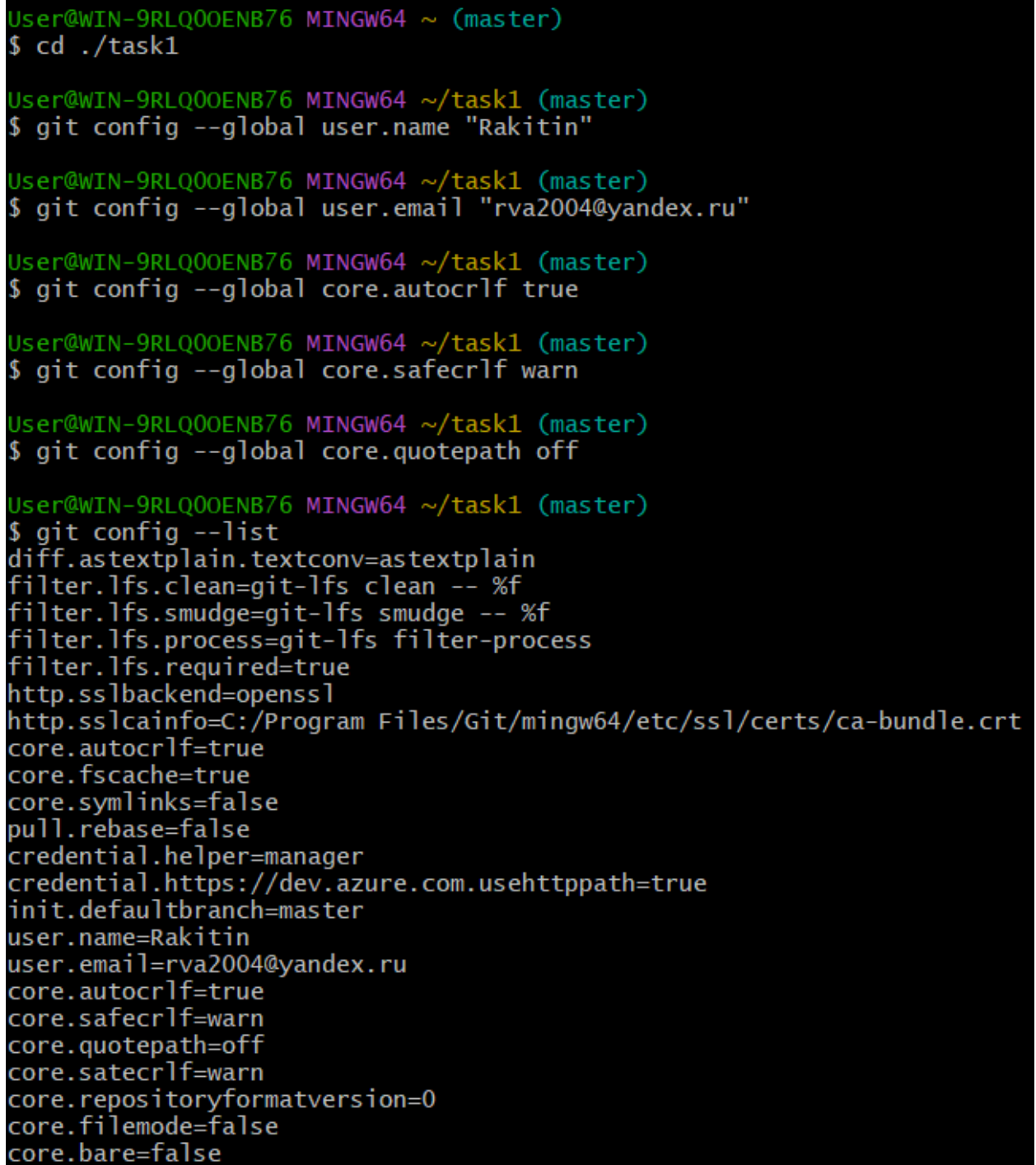
<b>1. Цель работы.....</b>	<b>3</b>
<b>2. Основные команды Git.....</b>	<b>4</b>
<b>3. Система управления репозиториями.....</b>	<b>12</b>
<b>4. Работа с ветвлением и оформление кода.....</b>	<b>19</b>
<b>5. Ответы на контрольные вопросы.....</b>	<b>26</b>
<b>6. Вывод .....</b>	<b>27</b>

## **1. Цель работы**

Получить навыки по работе с командами в Git.

## 2. Основные команды Git

1) Установите и настройте клиент git на своей рабочей станции (рисунок 1).



```
User@WIN-9RLQ00ENB76 MINGW64 ~ (master)
$ cd ./task1

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git config --global user.name "Rakitin"

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git config --global user.email "rva2004@yandex.ru"

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git config --global core.autocrlf true

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git config --global core.safecrlf warn

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git config --global core.quotepath off

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ 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/etc/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
user.name=Rakitin
user.email=rva2004@yandex.ru
core.autocrlf=true
core.safecrlf=warn
core.quotepath=off
core.safecrlf=warn
core.repositoryformatversion=0
core.filemode=false
core.bare=false
```

Рисунок 1 – Настройка клиента git

2) Создайте локальный репозиторий и добавьте в него несколько файлов (рисунок 2).

```

User@WIN-9RLQ00ENB76 MINGW64 ~/task1
$ git init
Initialized empty Git repository in C:/Users/User/task1/.git/

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ echo "Hello, my dog!" > file.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ echo "My cat's name - Ellis" > cat.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git add .
warning: in the working copy of 'cat.txt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git status
On branch master

No commits yet

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

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git commit -m "first commit"
[master (root-commit) 257b897] first commit
 2 files changed, 2 insertions(+)
 create mode 100644 cat.txt
 create mode 100644 file.txt

```

Рисунок 2 – Создание локального репозитория

- 3) Внесите изменения в один из файлов (рисунок 3).

```

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ echo "tree 38" >> file.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ cat file.txt
Hello, my dog!
tree 38

```

Рисунок 3 – Внесение изменения в файл

- 4) Проиндексируйте изменения и проверьте состояние (рисунок 4).

```

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git add .
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git status
On branch master

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   file.txt

```

Рисунок 4 – Индексация изменений

- 5) Сделайте коммит того, что было проиндексировано в репозиторий. Добавьте к коммиту комментарий (рисунок 5).

```

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git commit -m "tree 38 in file.txt"
[master 9a242cc] tree 38 in file.txt
 1 file changed, 1 insertion(+)

```

Рисунок 5 – Коммит изменения файла file.txt

6) Измените еще один файл. Добавьте это изменение в индекс git. Измените файл еще раз. Проверьте состояние и произведите коммит проиндексированного изменения. Теперь добавьте второе изменение в индекс, а затем проверьте состояние с помощью команды `git status`. Сделайте коммит второго изменения (рисунок 6).

```
User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ echo "She's 6 years old" >> cat.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git add cat.txt
warning: in the working copy of 'cat.txt', LF will be replaced by CRLF the next time Git touches it

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ echo "She's beautiful" >> cat.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   cat.txt

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:   cat.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git commit -m "She's beautiful in file.txt"
[master ac9cad9] She's beautiful in file.txt
1 file changed, 1 insertion(+)
```

Рисунок 6 – Коммит изменений файла cat.txt

7) Просмотрите историю коммитов с помощью команды `git log`. Ознакомьтесь с параметрами команды и используйте некоторые из них для различного формата отображения истории коммитов (рисунок 7).

```

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git log -p cat.txt
commit ac9cad9b0425c0d68f84c0b2f501a0eed52ab109 (HEAD -> master)
Author: Rakitin <rva2004@yandex.ru>
Date:   Wed Feb 28 23:18:54 2024 +0300

    She's beautiful in file.txt

diff --git a/cat.txt b/cat.txt
index 5f73bc3..c93228b 100644
--- a/cat.txt
+++ b/cat.txt
@@ -1,2 @@
-My cat's name - Ellis
+She's 6 years old

commit 257b897a934a18dd437de2d845f4a566c8ceeafa
Author: Rakitin <rva2004@yandex.ru>
Date:   Wed Feb 28 15:40:39 2024 +0300

    first commit

diff --git a/cat.txt b/cat.txt
new file mode 100644
index 0000000..5f73bc3
--- /dev/null
+++ b/cat.txt
@@ -0,0 +1 @@
+My cat's name - Ellis

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git log --graph
* commit ac9cad9b0425c0d68f84c0b2f501a0eed52ab109 (HEAD -> master)
| Author: Rakitin <rva2004@yandex.ru>
| Date:   Wed Feb 28 23:18:54 2024 +0300
|
|     She's beautiful in file.txt
|
| * commit 9a242ccaf8fd477d73544d1e565b3a93dfc4e434
| | Author: Rakitin <rva2004@yandex.ru>
| | Date:   Wed Feb 28 15:47:20 2024 +0300
| |
| |     tree 38 in file.txt
| |
| | * commit 257b897a934a18dd437de2d845f4a566c8ceeafa
| | | Author: Rakitin <rva2004@yandex.ru>
| | | Date:   Wed Feb 28 15:40:39 2024 +0300
| | |
| | |     first commit

```

Рисунок 7 – Просмотр истории коммитов

8) Верните рабочий каталог к одному из предыдущих состояний (рисунок 8).

```

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git checkout 257b897a
error: Your local changes to the following files would be overwritten by checkout:
    cat.txt
Please commit your changes or stash them before you switch branches.
Aborting

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (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:   cat.txt

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

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git restore cat.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git checkout 257b897a
Note: switching to '257b897a'.

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 257b897 first commit

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 ((257b897...))
$ cat file.txt
Hello, my dog!

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 ((257b897...))
$ cat cat.txt
My cat's name - Ellis

```

Рисунок 8 – Возврат каталога к предыдущему состоянию

9) Изучите, как создавать теги для коммитов для использования в будущем (рисунок 9).



```

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 ((257b897...))
$ git tag "potato"

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 ((potato))
$ git log
commit 257b897a934a18dd437de2d845f4a566c8ceeafa (HEAD, tag: potato)
Author: Rakitin <rva2004@yandex.ru>
Date:   Wed Feb 28 15:40:39 2024 +0300

    first commit

```

Рисунок 9 – Создание тега для коммита

10) Отмените некоторые изменения в рабочем каталоге (до и после индексирования) (рисунки 10 – 11).

```

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 ((potato))
$ git checkout master
Previous HEAD position was 257b897 first commit
Switched to branch 'master'

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ cat file.txt
Hello, my dog!
tree 38

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ echo "hair" >> file.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (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:   file.txt

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

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git checkout file.txt
Updated 1 path from the index

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git status
On branch master
nothing to commit, working tree clean

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ cat file.txt
Hello, my dog!
tree 38

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ echo "QWERTY" > file.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ cat file.txt
QWERTY

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git add file.txt
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   file.txt

```

Рисунок 10 – Отмена изменений в рабочем каталоге

```
User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git reset HEAD file.txt
Unstaged changes after reset:
M   file.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (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:   file.txt

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

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ cat file.txt
QWERTY

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git checkout file.txt
Updated 1 path from the index

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ cat file.txt
Hello, my dog!
tree 38
```

Рисунок 11 – Отмена изменений в рабочем каталоге. Продолжение

11) Отмените один из коммитов в локальном репозитории (рисунок 12).

```

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ echo "Tomato" >> file.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git add file.txt
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git commit -m "add Tomato in file.txt"
[master ed9eb07] add Tomato in file.txt
1 file changed, 1 insertion(+)

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git revert HEAD --no-edit
[master 8b613f4] Revert "add Tomato in file.txt"
Date: Thu Feb 29 00:09:30 2024 +0300
1 file changed, 1 deletion(-)

User@WIN-9RLQ00ENB76 MINGW64 ~/task1 (master)
$ git log
commit 8b613f48bda9747d72a8844875ac72acea92497a (HEAD -> master)
Author: Rakitin <rva2004@yandex.ru>
Date: Thu Feb 29 00:09:30 2024 +0300

    Revert "add Tomato in file.txt"

    This reverts commit ed9eb0743fa20d921c94af6272ee519ef1d5f108.

commit ed9eb0743fa20d921c94af6272ee519ef1d5f108
Author: Rakitin <rva2004@yandex.ru>
Date: Thu Feb 29 00:09:00 2024 +0300

    add Tomato in file.txt

commit ac9cad9b0425c0d68f84c0b2f501a0eed52ab109
Author: Rakitin <rva2004@yandex.ru>
Date: Wed Feb 28 23:18:54 2024 +0300

    She's beautiful in file.txt

commit 9a242ccaf8fd477d73544d1e565b3a93dfc4e434
Author: Rakitin <rva2004@yandex.ru>
Date: Wed Feb 28 15:47:20 2024 +0300

    tree 38 in file.txt

commit 257b897a934a18dd437de2d845f4a566c8ceeaafa (tag: potato)
Author: Rakitin <rva2004@yandex.ru>
Date: Wed Feb 28 15:40:39 2024 +0300

    first commit

```

Рисунок 12 – Отмена коммита

### 3. Системы управления репозиториями

1) Создайте аккаунт на GitHub (рисунок 13).

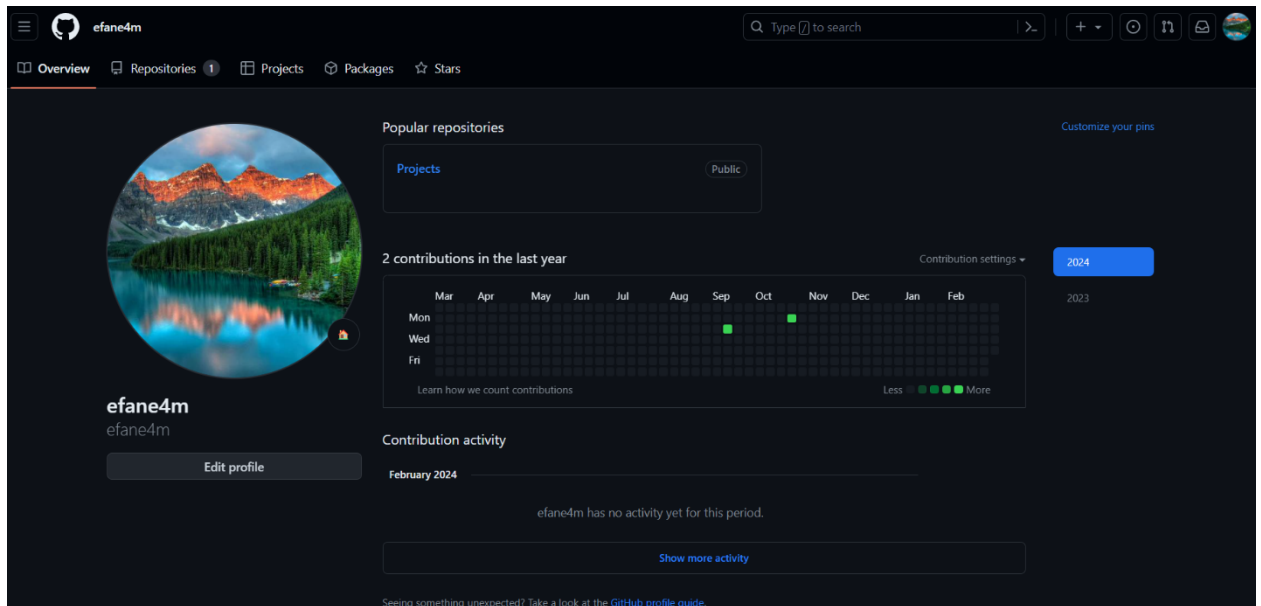


Рисунок 13 – Аккаунт на GitHub

2) Создайте репозиторий на GitHub и на локальной машине, согласно выбранной теме (рисунки 14 – 15).

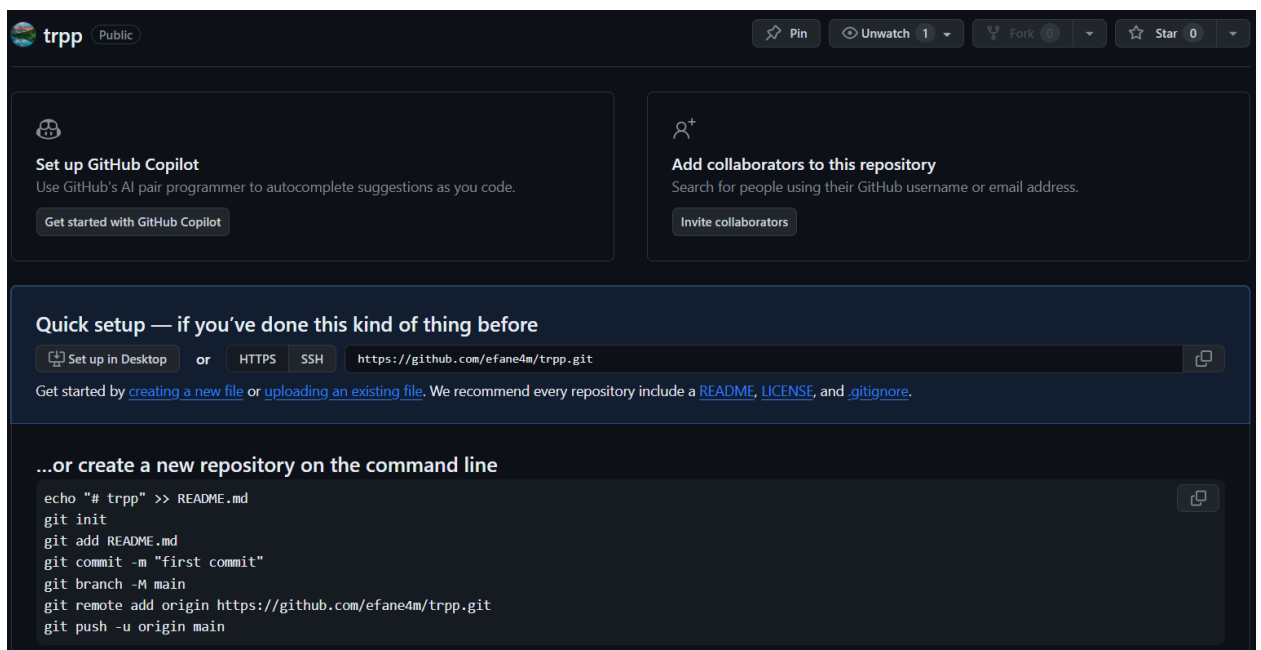


Рисунок 14 – Создание репозитория на GitHub

```

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract
$ mkdir trpp

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract
$ cd ./trpp

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/trpp
$ git init
Initialized empty Git repository in C:/Users/User/trpp_pract/trpp/.git/

```

Рисунок 15 – Создание репозитория на локальной машине

3) Создайте несколько файлов на локальной машине при помощи консоли (рисунок 16).

```

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/trpp (master)
$ touch fil1.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/trpp (master)
$ touch fil2.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/trpp (master)
$ git add .

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/trpp (master)
$ git commit -m "first commit"
[master (root-commit) 5108f65] first commit
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 fil1.txt
create mode 100644 fil2.txt

```

Рисунок 16 – Создание файлов и первого коммита

4) Создайте SSH-ключ для авторизации (рисунки 17 – 19).

```

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/trpp (master)
$ ssh-keygen -t rsa -b 4096 -C "rva2004@yandex.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/User/.ssh/id_rsa):
Created directory '/c/Users/User/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/User/.ssh/id_rsa
Your public key has been saved in /c/Users/User/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:nLrUot6xcNXOGAQz6o+FmEmv8TkIX+bnZqVGZanvzeQ rva2004@yandex.ru
The key's randomart image is:
+---[RSA 4096]-----+
|
|  .
|  . +
|  . . .
|  . * . o+o
|  . = * .+S .
|  o O =o+.=
|  + B.O+o +
|  X==.=
|  .o+=.. E
+---[SHA256]-----+

```

Рисунок 17 – Создание SSH-ключа для авторизации

```

User@WIN-9RLQ00ENB76 MINGW64 ~/.ssh
$ ssh-add ~/.ssh/id_rsa
Could not open a connection to your authentication agent.

User@WIN-9RLQ00ENB76 MINGW64 ~/.ssh
$ eval "$(ssh-agent -s)"
Agent pid 1678

User@WIN-9RLQ00ENB76 MINGW64 ~/.ssh
$ ssh-add ~/.ssh/id_rsa
Identity added: /c/Users/User/.ssh/id_rsa (rva2004@yandex.ru)

User@WIN-9RLQ00ENB76 MINGW64 ~/.ssh
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDAuS2Gm2hPdxIFciUiEw2fAXj+zQ2snZrEqCF900155FEFyGQ0aomwD/bluBTZRULNB+gI6jVT/Cgub9gDkMsou0sqDPNdtRmbbf4rLfjFWowj+CDX/bsruR5gxzyRJct8Y5YkdVRmXt/8N9CeOAut/8ioPjS+qwCBUJnaec2wUlw1rsTMLcPy4iAXgCyp29801PdVwMhEw4ZR20NBuJXbSLVBVkyteVJeCDebts196o7+6dlz/0UhwfPUB4c613aPS2jQTXQ0AFwi0NL3LqxytNy6wxfEiMA/UBi/Xt5y4R8owk8mUt1dYY1x6ngEWPQwmhJiPbCgGyJ4267SS7ZvtLEPQa4CofJ7npbBjeo8Iu9C08Yskwv7p+pPjAg69UfpiXgZKvvogwO5j51KCyGHLNwt6mk3rhJe3XOAjtp95kGC+qhWZR49N19Qbga4AdO5Gqlm3u4PwaT7Vks48ZtdCOhzSdKcbTBkVZ8HiZZgdmqD3ABJ7IdCFCgKaYvkkIuSnrNmBC7w9uPMq1pMjyROH6tUGFMzOq9H7RSQRbaLEAR2H0monVEikv3HEsN104D1uKpvKvIXcrq3kHfo89iZcjP9LtdDpWmsUAvIrddCVhsW9jS/SFZS2lna7fgIVW0FXxbqrRxYioz4hL0vgKBXoH068MiBsWHbtm3Q== rva2004@yandex.ru

```

Рисунок 18 – Создание SSH-ключа. Продолжение

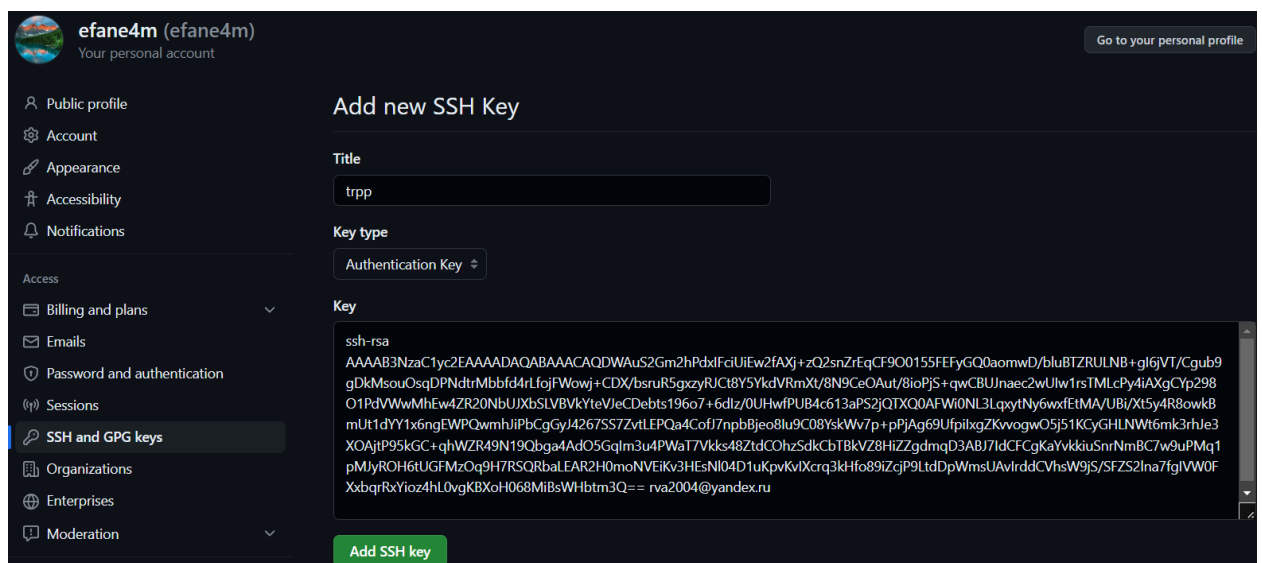


Рисунок 19 – Создание SSH-ключа на GitHub

5) Свяжите репозиторий локальной машины с репозиторием на GitHub при помощи консоли (рисунок 20).

```

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/trpp (master)
$ git remote add trpp git@github.com:efane4n/trpp.git

```

Рисунок 20 – Связывание репозитория

6) Создайте новую ветку в репозитории с помощью команды, произведите в ней какие-нибудь изменения, а после слейте с веткой master (рисунок 21).

```

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/trpp (master)
$ git checkout -b second
Switched to a new branch 'second'

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/trpp (second)
$ echo "Banana" >> fill1.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/trpp (second)
$ git add fill1.txt
warning: in the working copy of 'fill1.txt', LF will be replaced by CRLF the next time Git touches it

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/trpp (second)
$ git commit -m "change fill1.txt"
[second 6ac0675] change fill1.txt
1 file changed, 1 insertion(+)

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/trpp (second)
$ git checkout master
Switched to branch 'master'

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/trpp (master)
$ git merge second
Updating 5108f65..6ac0675
Fast-forward
 fill1.txt | 1 +
 1 file changed, 1 insertion(+)
User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/trpp (master)
$ git log --graph
* commit 6ac0675b96e50cfb9598c288d3ce69565b1ab62c (HEAD -> master, second)
  | Author: Rakitin <rva2004@yandex.ru>
  | Date: Thu Feb 29 04:21:31 2024 +0300
  |
  | change fill1.txt
  |
* commit 5108f652f895f260d7da1fdfe3ff9f5cb8e2273a
  | Author: Rakitin <rva2004@yandex.ru>
  | Date: Thu Feb 29 02:26:00 2024 +0300
  |
  | first commit

```

Рисунок 21 – Создание ветки, внесение изменений и сливание с master

7) Выполните цепочку действий в репозитории, согласно варианту 4:

- Клонировать непустой удаленный репозиторий на локальную машину (рисунок 22).

```

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract
$ git clone git@github.com:efane4m/Project.git
Cloning into 'Project'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.

```

Рисунок 22 – Клонирование репозитория

- Создайте новую ветку и выведите список всех веток (рисунок 23).

```

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract
$ cd ./Project

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/Project (main)
$ git branch "second"

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/Project (main)
$ git branch
* main
  second

```

Рисунок 23 – Создание ветки и вывод списка веток

- Произведите 3 коммита в новой ветке в разные файлы (рисунок 24).

```

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/Project (main)
$ touch a.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/Project (main)
$ touch b.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/Project (main)
$ touch c.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/Project (main)
$ echo "qwerty" > a.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/Project (main)
$ git checkout second
Switched to branch 'second'

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/Project (second)
$ git add a.txt
warning: in the working copy of 'a.txt', LF will be replaced by CRLF the next time Git touches it

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/Project (second)
$ git commit -m "add a.txt"
[second b3f3819] add a.txt
1 file changed, 1 insertion(+)
create mode 100644 a.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/Project (second)
$ echo "1234" > b.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/Project (second)
$ git add b.txt
warning: in the working copy of 'b.txt', LF will be replaced by CRLF the next time Git touches it

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/Project (second)
$ git commit -m "add b.txt"
[second 0e7bc08] add b.txt
1 file changed, 1 insertion(+)
create mode 100644 b.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/Project (second)
$ echo "0909090909090909" > c.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/Project (second)
$ git add c.txt
warning: in the working copy of 'c.txt', LF will be replaced by CRLF the next time Git touches it

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/Project (second)
$ git commit -m "add c.txt"
[second 431f52b] add c.txt
1 file changed, 1 insertion(+)
create mode 100644 c.txt

```

Рисунок 24 – создание трёх коммитов в разных файлах

- Выгрузите изменения в удаленный репозиторий (рисунок 25).



```

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/Project (second)
$ git push origin main
Everything up-to-date

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/Project (second)
$ git push origin second
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 680 bytes | 340.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
remote:
remote: Create a pull request for 'second' on GitHub by visiting:
remote:   https://github.com/efane4m/Project/pull/new/second
remote:
To github.com:efane4m/Project.git
 * [new branch]      second -> second

```

Рисунок 25 – Выгрузка изменений в удаленный репозиторий

- Откатите в новой ветке предпоследний коммит (в том числе в удаленном репозитории) (рисунок 26).

```

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/Project (second)
$ git reset --hard HEAD~1
HEAD is now at 0e7bc08 add b.txt

```

Рисунок 26 – Откат предпоследнего коммита

- Выведите в консоли различия между веткой master и новой веткой (рисунок 27).

```

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/Project (second)
$ git diff main second
diff --git a/a.txt b/a.txt
new file mode 100644
index 0000000..19f0805
--- /dev/null
+++ b/a.txt
@@ -0,0 +1 @@
+qwerty
diff --git a/b.txt b/b.txt
new file mode 100644
index 0000000..81c545e
--- /dev/null
+++ b/b.txt
@@ -0,0 +1 @@
+1234

```

Рисунок 28 – Различия между master и second

- Слейте новую ветку с master при помощи merge (рисунок 28).

```
User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/Project (second)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/Project (main)
$ git merge second
Updating 915408d..0e7bc08
Fast-forward
 a.txt | 1 +
 b.txt | 1 +
 2 files changed, 2 insertions(+)
 create mode 100644 a.txt
 create mode 100644 b.txt
```

Рисунок 29 – Слияние веток при помощи merge

## 4. Работа с ветвлением и оформлением кода

1. Сделайте форк репозитория в соответствии с вариантом 4: <https://github.com/ripienaar/free-for-dev> (рисунок 30).

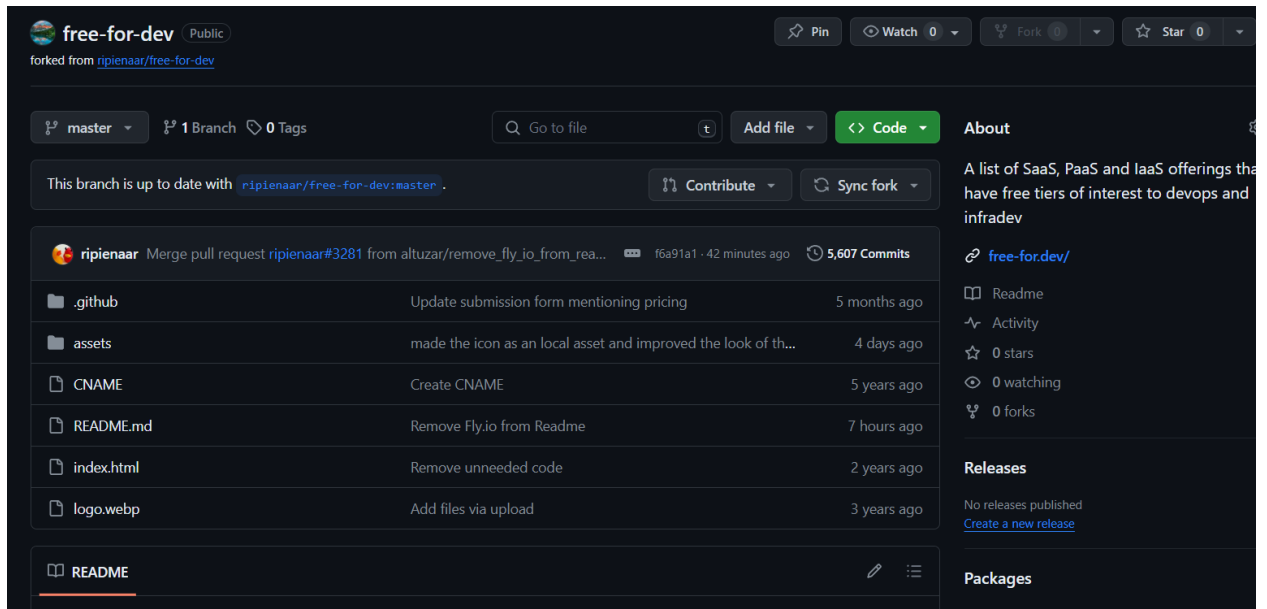


Рисунок 30 – Форк репозитория

2. Склонируйте его на локальную машину (рисунок 31).

```
User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract
$ git clone https://github.com/ripienaar/free-for-dev.git
Cloning into 'free-for-dev'...
remote: Enumerating objects: 13235, done.
remote: Counting objects: 100% (137/137), done.
remote: Compressing objects: 100% (96/96), done.
remote: Total 13235 (delta 74), reused 94 (delta 40), pack-reused 13098
Receiving objects: 100% (13235/13235), 7.20 MiB | 591.00 KiB/s, done.
Resolving deltas: 100% (6881/6881), done.
```

Рисунок 31 – Клонирование репозитория

3. Создайте две ветки branch1 и branch2 от последнего коммита в master'e (рисунок 32).

```
User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract
$ cd ./free-for-dev

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (master)
$ git branch "branch1"

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (master)
$ git branch "branch2"
```

Рисунок 32 – Создание веток branch1 и branch2

4. Проведите по 3 коммита в каждую из веток, которые меняют один и тот же кусочек файла (рисунок 33).

```
User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (master)
$ git checkout branch1
Switched to branch 'branch1'

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch1)
$ nano CNAME

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch1)
$ git add CNAME
warning: in the working copy of 'CNAME', LF will be replaced by CRLF the next time Git touches it

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch1)
$ git commit -m "first commit b1"
[branch1 f4a9f72] first commit b1
1 file changed, 2 insertions(+), 1 deletion(-)

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch1)
$ nano CNAME

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch1)
$ git add CNAME
warning: in the working copy of 'CNAME', LF will be replaced by CRLF the next time Git touches it

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch1)
$ git commit -m "second commit b1"
[branch1 a4a6885] second commit b1
1 file changed, 1 insertion(+)

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch1)
$ nano CNAME

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch1)
$ git add CNAME
warning: in the working copy of 'CNAME', LF will be replaced by CRLF the next time Git touches it

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch1)
$ git commit -m "third commit b1"
[branch1 c194ed7] third commit b1
1 file changed, 1 insertion(+)

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch1)
$ git checkout branch2
Switched to branch 'branch2'

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch2)
$ nano CNAME

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch2)
$ git add CNAME
warning: in the working copy of 'CNAME', LF will be replaced by CRLF the next time Git touches it

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch2)
$ git commit -m "first commit b2"
[branch2 ebbb328] first commit b2
1 file changed, 2 insertions(+), 1 deletion(-)

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch2)
$ nano CNAME

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch2)
$ git add CNAME
warning: in the working copy of 'CNAME', LF will be replaced by CRLF the next time Git touches it

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch2)
$ git commit -m "second commit b2"
[branch2 efbbfc2] second commit b2
1 file changed, 1 insertion(+)

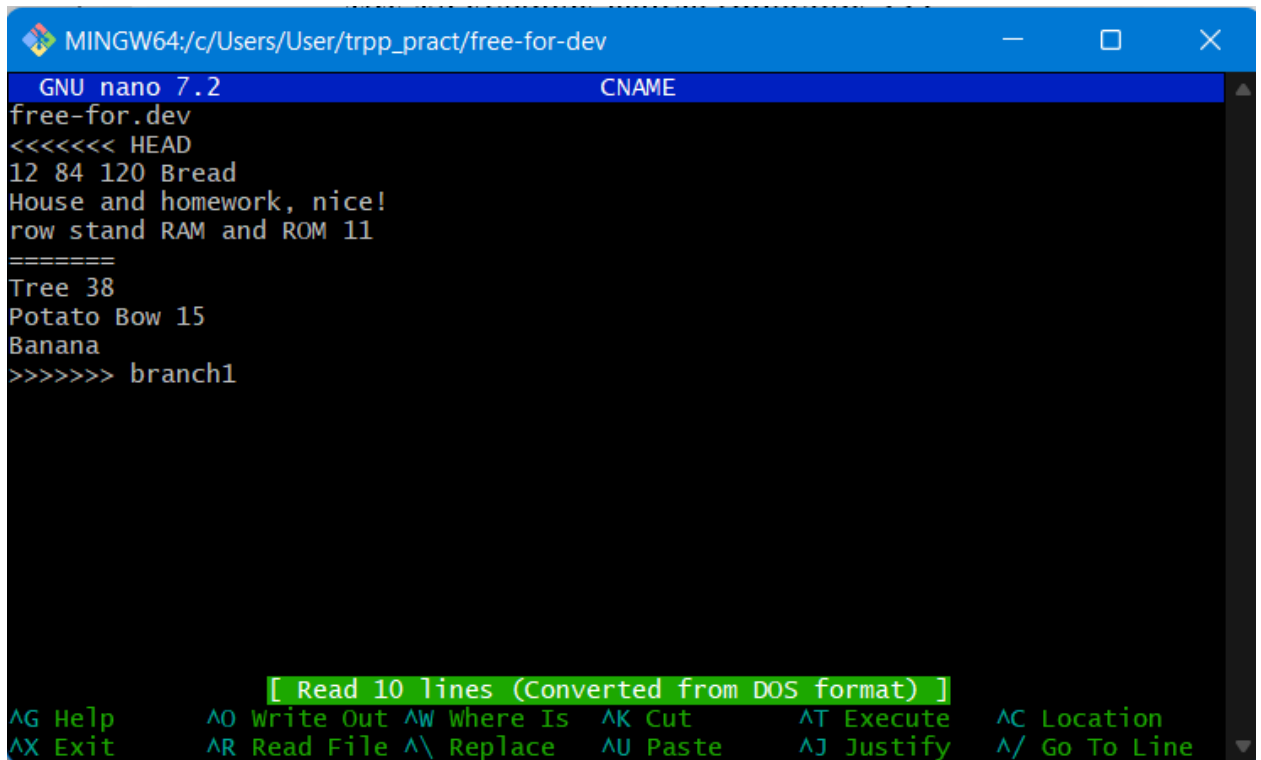
User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch2)
$ nano CNAME

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch2)
$ git add CNAME
warning: in the working copy of 'CNAME', LF will be replaced by CRLF the next time Git touches it

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch2)
$ git commit -m "third commit b2"
[branch2 624b1e3] third commit b2
1 file changed, 1 insertion(+)
```

Рисунок 33 – выполнение коммитов на каждой ветке

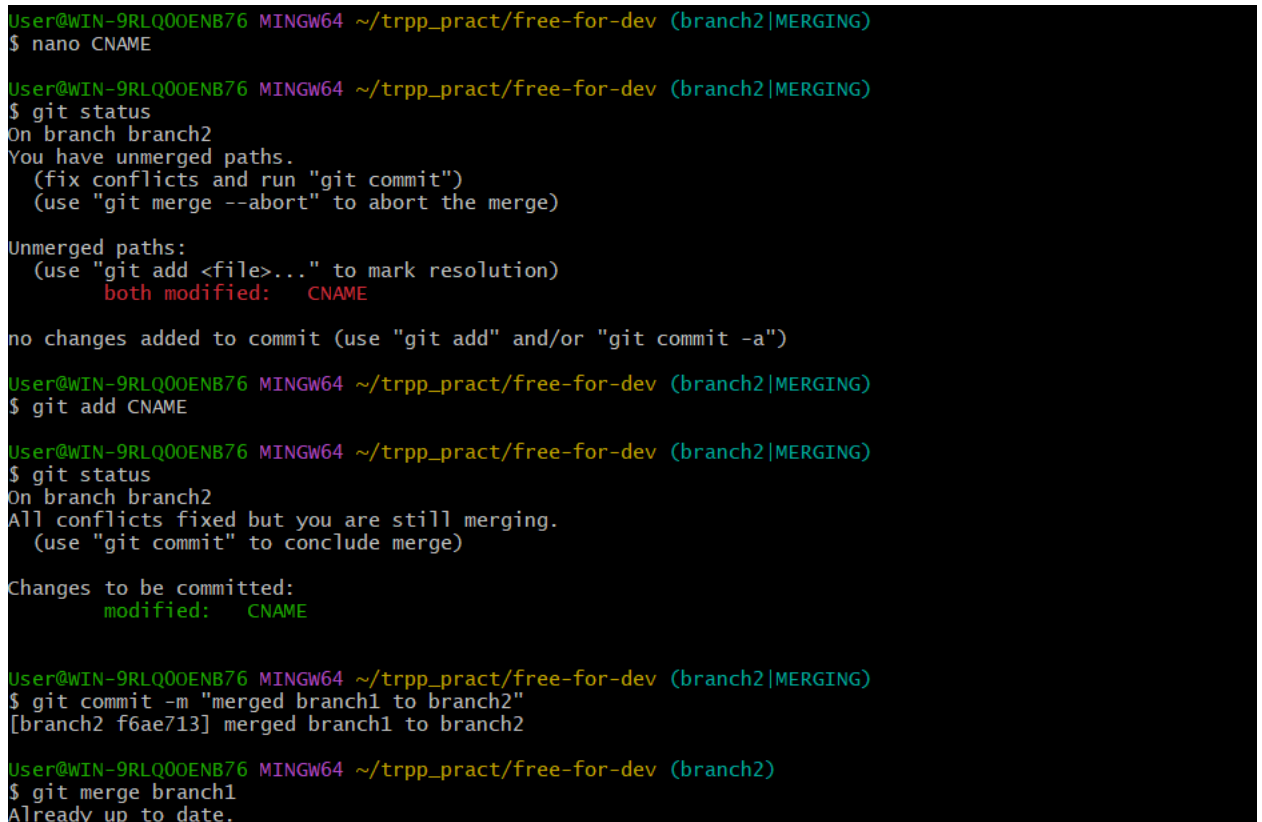
5. Выполните слияние ветки branch1 в ветку branch2, разрешив конфликты при этом (рисунки 34 – 35).



```
MINGW64:/c/Users/User/trpp_pract/free-for-dev
GNU nano 7.2 CNAME
free-for.dev
<<<<<<< HEAD
12 84 120 Bread
House and homework, nice!
row stand RAM and ROM 11
=====
Tree 38
Potato Bow 15
Banana
>>>>>> branch1

[ Read 10 lines (Converted from DOS format) ]
AG Help      AO Write Out  AW Where Is  AK Cut       AT Execute   AC Location
AX Exit      AR Read File  AL Replace   AU Paste     AJ Justify   AL Go To Line
```

Рисунок 34 – Конфликт слияния веток branch1 и branch2



```
User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch2|MERGING)
$ nano CNAME

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch2|MERGING)
$ git status
On branch branch2
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   CNAME

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

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch2|MERGING)
$ git add CNAME

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch2|MERGING)
$ git status
On branch branch2
All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:
  modified:   CNAME

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch2|MERGING)
$ git commit -m "merged branch1 to branch2"
[branch2 f6ae713] merged branch1 to branch2

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch2)
$ git merge branch1
Already up to date.
```

Рисунок 35 – Слияние веток branch1 и branch2

6. Выгрузите все изменения во всех ветках в удаленный репозиторий (рисунки 36 – 37).

```
User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch2)
$ git remote set-url origin git@github.com:efane4m/free-for-dev.git

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch2)
$ git push origin master
Everything up-to-date

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch2)
$ git push origin branch1
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 792 bytes | 396.00 KiB/s, done.
Total 9 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
remote:
remote: Create a pull request for 'branch1' on GitHub by visiting:
remote:   https://github.com/efane4m/free-for-dev/pull/new/branch1
remote:
To github.com:efane4m/free-for-dev.git
 * [new branch]      branch1 -> branch1

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch2)
$ git push origin branch2
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 12 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (12/12), 1.07 KiB | 546.00 KiB/s, done.
Total 12 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), completed with 1 local object.
remote:
remote: Create a pull request for 'branch2' on GitHub by visiting:
remote:   https://github.com/efane4m/free-for-dev/pull/new/branch2
remote:
To github.com:efane4m/free-for-dev.git
 * [new branch]      branch2 -> branch2
```

Рисунок 36 – Выгрузка всех изменений в удаленный репозиторий

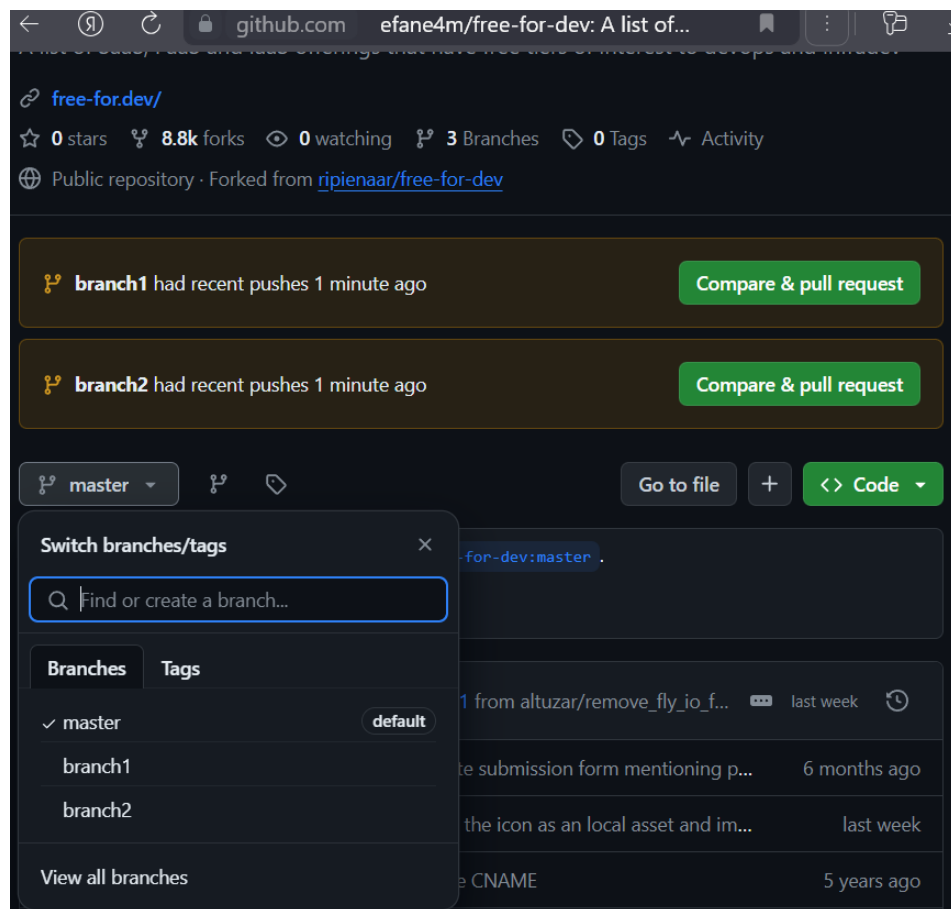


Рисунок 37 – Результат выгрузки изменений в удаленный репозиторий

7. Проведите еще 3 коммита в ветку branch1 (рисунок 38).

```
User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch1)
$ echo "How ar u?" > CNAME

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch1)
$ git add CNAME
warning: in the working copy of 'CNAME', LF will be replaced by CRLF the next time Git touches it

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch1)
$ git commit -m "New text in CNAME"
[branch1 bc0aa40] New text in CNAME
1 file changed, 1 insertion(+), 4 deletions(-)

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch1)
$ echo "I'm fine!" >> CNAME

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch1)
$ git add CNAME
warning: in the working copy of 'CNAME', LF will be replaced by CRLF the next time Git touches it

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch1)
$ git commit -m "Add str 1"
[branch1 5122986] Add str 1
1 file changed, 1 insertion(+)

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch1)
$ echo "Nice" >> CNAME

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch1)
$ git add CNAME
warning: in the working copy of 'CNAME', LF will be replaced by CRLF the next time Git touches it

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch1)
$ git commit -m "Add str 2"
[branch1 761f051] Add str 2
1 file changed, 1 insertion(+)
```

Рисунок 38 – Создание коммитов в branch1

8. Склонируйте репозиторий еще раз в другую директорию (рисунок 39).

```
User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch1)
$ cd ..

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract
$ mkdir newRep

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract
$ cd ./newRep

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/newRep
$ git clone https://github.com/efane4m/free-for-dev.git
Cloning into 'free-for-dev'...
remote: Enumerating objects: 13256, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 13256 (delta 8), reused 19 (delta 6), pack-reused 13235
Receiving objects: 100% (13256/13256), 7.14 MiB | 962.00 KiB/s, done.
Resolving deltas: 100% (6891/6891), done.
```

Рисунок 39 – Клонирования репозитория в папку newRep

9. В новом клоне репозитории сделайте 3 коммита в ветку branch1 (рисунок 40).



```

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/newRep/free-for-dev (master)
$ git checkout branch1
Switched to a new branch 'branch1'
branch 'branch1' set up to track 'origin/branch1'.

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/newRep/free-for-dev (branch1)
$ touch program.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/newRep/free-for-dev (branch1)
$ echo "for i in range(1, 11):" >> program.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/newRep/free-for-dev (branch1)
$ git add program.txt
warning: in the working copy of 'program.txt', LF will be replaced by CRLF the next time Git touches it

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/newRep/free-for-dev (branch1)
$ git commit -m "Add first str-code"
[branch1 0f78357] Add first str-code
1 file changed, 1 insertion(+)
create mode 100644 program.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/newRep/free-for-dev (branch1)
$ echo "    print(i, end=" ") >> program.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/newRep/free-for-dev (branch1)
$ git add program.txt
warning: in the working copy of 'program.txt', LF will be replaced by CRLF the next time Git touches it

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/newRep/free-for-dev (branch1)
$ git commit -m "Add second str-code"
[branch1 7003117] Add second str-code
1 file changed, 1 insertion(+)

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/newRep/free-for-dev (branch1)
$ echo "print('A')" >> program.txt

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/newRep/free-for-dev (branch1)
$ git add program.txt
warning: in the working copy of 'program.txt', LF will be replaced by CRLF the next time Git touches it

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/newRep/free-for-dev (branch1)
$ git commit -m "Add third str-code"
[branch1 c6ca813] Add third str-code
1 file changed, 1 insertion(+)

```

Рисунок 40 – Создание коммитов в branch1

10. Выгрузите все изменения из нового репозитория в удаленный репозиторий (рисунок 41).

```

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/newRep/free-for-dev (branch1)
$ git remote set-url origin git@github.com:efane4m/free-for-dev.git

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/newRep/free-for-dev (branch1)
$ git push origin branch1
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (9/9), 801 bytes | 400.00 KiB/s, done.
Total 9 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To github.com:efane4m/free-for-dev.git
c194ed7..c6ca813 branch1 -> branch1

```

Рисунок 41 – Выгрузка изменений в удаленный репозиторий

11. Вернитесь в старый клон с репозиторием, выгрузите изменения с опцией `–force` (рисунок 42).



```

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/newRep/free-for-dev (branch1)
$ cd ../../

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract
$ cd ./free-for-dev

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch1)
$ git push origin branch1 --force
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 754 bytes | 251.00 KiB/s, done.
Total 9 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To github.com:efane4m/free-for-dev.git
+ c6ca813...761f051 branch1 -> branch1 (forced update)

```

Рисунок 42 – Выгрузка изменений опцией --force

12. Получите все изменения в новом репозитории (рисунок 43).

```

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/free-for-dev (branch1)
$ cd ../

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract
$ cd ./newRep/free-for-dev

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/newRep/free-for-dev (branch1)
$ git config pull.rebase true

User@WIN-9RLQ00ENB76 MINGW64 ~/trpp_pract/newRep/free-for-dev (branch1)
$ git pull
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 9 (delta 3), reused 9 (delta 3), pack-reused 0
Unpacking objects: 100% (9/9), 734 bytes | 17.00 KiB/s, done.
From github.com:efane4m/free-for-dev
+ c6ca813...761f051 branch1 -> origin/branch1 (forced update)
Successfully rebased and updated refs/heads/branch1.

```

Рисунок 43 – Получение всех изменений в новом репозитории

## 5. Ответы на контрольные вопросы

1) Что такое коммит?

Коммит – это команда, которая фиксирует изменения в репозитории

2) Что такое ветка в репозитории Git?

Ветка в Git — это набор коммитов, расположенных в хронологическом порядке.

3) Что такое слияние двух веток?

Слияние веток – это перенос изменений с одной ветки на другую.

4) Что делает команда `git log`?

Команда `git log` показывает историю проекта, какие изменения происходили.

5) Что сделает команда “`git branch`” без какого-либо параметра?

Команда `git branch` без параметров отображает все ветки, которые созданы локально.

6) Чем отличается `master` и `origin/master`?

`Master` является веткой локального репозитория, а `origin/master` – веткой удаленного репозитория

7) Что такое GitHub?

GitHub – это крупнейший веб-сервис для хостинга своих проектов, основанный на системе контроля версий Git.

8) Что такое форк репозитория?

Форк репозитория – это создание его копии.

## **6. Вывод**

Во время выполнения работы были получены навыки по работе с командами в Git.