



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: «Система контроля версий»

Выполнил студент группы ИКБО-20-21

Хитров Н.С.

Принял к.т.н. доцент

Петренко А.А.

Практическая работа выполнена

✓

«4» марта 2023г.

✓

(подпись студента)

«Зачтено»

✓

«__» _____ 2023г.

(подпись руководителя)

Москва 2023

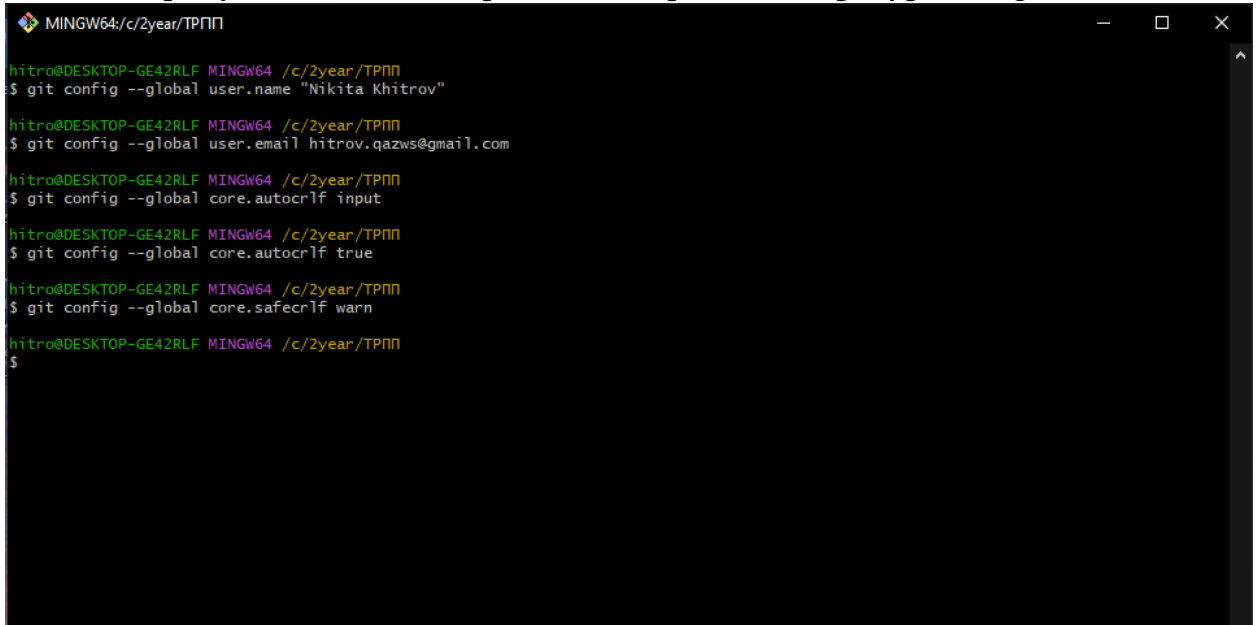
ОГЛАВЛЕНИЕ

Часть 1. Основные команды git	2
Часть 2. Системы управления репозиториями	12
Часть 3. Работа с ветвлением и оформление кода.....	19
Ответы на вопросы.....	26
Выводы	28

Часть 1. Основные команды git

1. Установите и настройте клиент git на своей рабочей станции

На рисунке 1 показан процесс настройки конфигурации git.

A screenshot of a terminal window titled 'MINGW64:/c/2year/ТПП'. The terminal shows a series of git configuration commands being executed. The prompt is 'hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП'. The commands and their outputs are: 1. '\$ git config --global user.name "Nikita Khitrov"' followed by a blank line. 2. '\$ git config --global user.email hitrov.qazws@gmail.com' followed by a blank line. 3. '\$ git config --global core.autocrlf input' followed by a blank line. 4. '\$ git config --global core.autocrlf true' followed by a blank line. 5. '\$ git config --global core.safecrlf warn' followed by a blank line. 6. '\$' followed by a blank line. The terminal has a dark background with light-colored text. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```
MINGW64:/c/2year/ТПП
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП
$ git config --global user.name "Nikita Khitrov"

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП
$ git config --global user.email hitrov.qazws@gmail.com

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП
$ git config --global core.autocrlf input

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП
$ git config --global core.autocrlf true

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП
$ git config --global core.safecrlf warn

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП
$
```

Рисунок 1 – Настройка конфигурации git

2. Создайте локальный репозиторий и добавьте в него несколько файлов.

На рисунке 2 показан процесс инициализации пустого репозитория и добавления файлов в папку проекта.

```
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1
$ git init
Initialized empty Git repository in C:/2year/ТПП/пр1/.git/

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ touch first.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ touch second.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$
```

Рисунок 2 – Инициализация репозитория git

3. Внесите изменения в один из файлов.

```
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ echo "first file first change" > first.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$
```

Рисунок 3 – Внесение изменений в файл

4. Проиндексируйте изменения и проверьте состояние.

```
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ git add first.txt
warning: in the working copy of 'first.txt', LF will be replaced by CRLF the next time Git touches it

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ git status
On branch master

No commits yet

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

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

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ |
```

Рисунок 4 – Индексация изменений

5. Сделайте коммит того, что было проиндексировано в репозиторий. Добавьте к коммиту комментарий.

На рисунке 5 показан процесс изменения файла index.html, индексация изменений и создание коммита.

```
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ git commit -m "First commit"
[master (root-commit) dbd5c18] First commit
1 file changed, 1 insertion(+)
create mode 100644 first.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ |
```

Рисунок 5 – изменение файла и создание коммита

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.

На рисунке 6 показан процесс создания коммитов с разными изменениями одного файла.

```
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ echo "second file first change" > second.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ git add second.txt
warning: in the working copy of 'second.txt', LF will be replaced by CRLF the next time Git touches it

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ echo "second file second change" > second.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   second.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:   second.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ git commit -m "Second commit"
[master 15f5e7e] Second commit
1 file changed, 1 insertion(+)
create mode 100644 second.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ git add second.txt
warning: in the working copy of 'second.txt', LF will be replaced by CRLF the next time Git touches it

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   second.txt

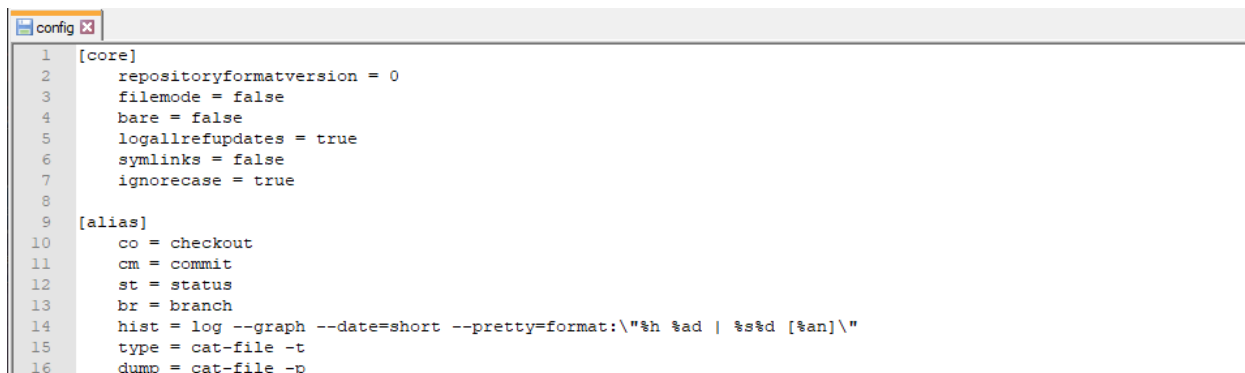
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ git commit -m "Third commit second.txt_v2"
[master 08f0617] Third commit second.txt_v2
1 file changed, 1 insertion(+), 1 deletion(-)

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$
```

Рисунок 6 – Изменение другого файла, индексация, повторное изменение и тд.

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

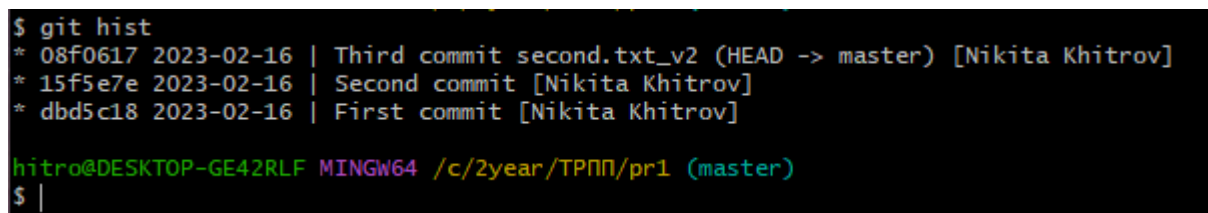
На рисунке 7 показана пользовательская конфигурация команд.



```
1 [core]
2   repositoryformatversion = 0
3   filemode = false
4   bare = false
5   logallrefupdates = true
6   symlinks = false
7   ignorecase = true
8
9 [alias]
10  co = checkout
11  cm = commit
12  st = status
13  br = branch
14  hist = log --graph --date=short --pretty=format:"%h %ad | %s%d [%an]"
15  type = cat-file -t
16  dump = cat-file -p
```

Рисунок 7 – пользовательская конфигурация команд

На рисунке 8 показана пользовательская конфигурация команд.



```
$ git hist
* 08f0617 2023-02-16 | Third commit second.txt_v2 (HEAD -> master) [Nikita Khitrov]
* 15f5e7e 2023-02-16 | Second commit [Nikita Khitrov]
* dbd5c18 2023-02-16 | First commit [Nikita Khitrov]

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ |
```

Рисунок 8 – История коммитов

1. Верните рабочий каталог к одному из предыдущих состояний.

На рисунке 9 показан процесс возврата репозитория к одному из предыдущих состояний.

```

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ git co 15f5e7e
Note: switching to '15f5e7e'.

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 15f5e7e Second commit

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 ((15f5e7e...))
$ .....

```

Рисунок 9 – Возврат репозитория к предыдущему состоянию

2. Изучите, как создавать теги для коммитов для использования в будущем.

На рисунке 10 показан процесс создания тегов для коммита.

```

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 ((v1))
$ git co dbd5c18
Previous HEAD position was 15f5e7e Second commit
HEAD is now at dbd5c18 First commit

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 ((dbd5c18...))
$ git tag v1_1

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 ((v1_1))
$ git co 15f5e7e
Previous HEAD position was dbd5c18 First commit
HEAD is now at 15f5e7e Second commit

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 ((v1))
$ git tag v1_2

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 ((v1))
$ git co 08f0617
Previous HEAD position was 15f5e7e Second commit
HEAD is now at 08f0617 Third commit second.txt_v2

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 ((08f0617...))
$ git tag v2_1

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 ((v2_1))
$ |

```

Рисунок 10 – Создание тегов для коммита

3. Отмените некоторые изменения в рабочем каталоге (до и после индексирования).

На рисунке 11 показан процесс отмены изменений до индексации.

```
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 ((v2_1))
$ git co master
Switched to branch 'master'

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ git status
On branch master
nothing to commit, working tree clean

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ echo "second file third change" > second.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (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:   second.txt

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

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ git co second.txt
Updated 1 path from the index

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ git status
On branch master
nothing to commit, working tree clean

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ cat second.txt
second file second change

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$
```

Рисунок 11 – Отмена изменений до индексации

На рисунке 12 показан процесс отмены изменений после индексации.

```
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ echo "first filst third change" > first.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ git add first.txt
warning: in the working copy of 'first.txt', LF will be replaced by CRLF the next time Git touches it

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   first.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ git reset HEAD first.txt
Unstaged changes after reset:
M       first.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ git checkout first.txt
Updated 1 path from the index

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ git status
On branch master
nothing to commit, working tree clean

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$
```

Рисунок 12 – Отмена проиндексированных изменений

4. Отмените один из коммитов в локальном репозитории.

На рисунке 14 показан процесс создания нежелаемого коммита.

```
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ touch unwanted.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ echo "some unwanted text" > unwanted.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ cat unwanted.txt
some unwanted text

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ git add unwanted.txt
warning: in the working copy of 'unwanted.txt', LF will be replaced by CRLF the next time Git touches it

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ git commit -m "Some unwanted commit"
[master a71c84b] Some unwanted commit
1 file changed, 1 insertion(+)
create mode 100644 unwanted.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ git log
commit a71c84b99e5a0137dd39a2204ec39b540b862097 (HEAD -> master)
Author: Nikita Khitrov <hitrov.qazws@gmail.com>
Date: Fri Feb 17 00:50:25 2023 +0300

    Some unwanted commit

commit 08f061794379ae77769489eee3a4efd2aaaddece (tag: v2_1)
Author: Nikita Khitrov <hitrov.qazws@gmail.com>
Date: Thu Feb 16 19:30:43 2023 +0300

    Third commit second.txt_v2

commit 15f5e7ef9a2ae1cd1a837b64c5d0239877f3e8c3 (tag: v1_2, tag: v1)
Author: Nikita Khitrov <hitrov.qazws@gmail.com>
Date: Thu Feb 16 19:29:02 2023 +0300

    Second commit

commit dbd5c18e1654b93efbbc0ab119d23d0c874fb163 (tag: v1_1)
Author: Nikita Khitrov <hitrov.qazws@gmail.com>
Date: Thu Feb 16 19:25:12 2023 +0300

    First commit

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
```

Рисунок 14 – создание нежелаемого коммита

На рисунке 15 показан процесс удаления последнего коммита.

```

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/TPNN/pr1 (master)
$ git revert HEAD --no-edit
[master 4dbf97b] Revert "Some unwanted commit"
Date: Fri Feb 17 00:53:42 2023 +0300
1 file changed, 1 deletion(-)
delete mode 100644 unwanted.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/TPNN/pr1 (master)
$ git log
commit 4dbf97bc45e4019cfae4e894f369f81f6217f7ed (HEAD -> master)
Author: Nikita Khitrov <hitrov.qazws@gmail.com>
Date: Fri Feb 17 00:53:42 2023 +0300

    Revert "Some unwanted commit"

    This reverts commit a71c84b99e5a0137dd39a2204ec39b540b862097.

commit a71c84b99e5a0137dd39a2204ec39b540b862097
Author: Nikita Khitrov <hitrov.qazws@gmail.com>
Date: Fri Feb 17 00:50:25 2023 +0300

    Some unwanted commit

commit 08f061794379ae77769489eee3a4efd2aaaddece (tag: v2_1)
Author: Nikita Khitrov <hitrov.qazws@gmail.com>
Date: Thu Feb 16 19:30:43 2023 +0300

    Third commit second.txt_v2

commit 15f5e7ef9a2ae1cd1a837b64c5d0239877f3e8c3 (tag: v1_2, tag: v1)
Author: Nikita Khitrov <hitrov.qazws@gmail.com>
Date: Thu Feb 16 19:29:02 2023 +0300

    Second commit

commit dbd5c18e1654b93efbbc0ab119d23d0c874fb163 (tag: v1_1)
Author: Nikita Khitrov <hitrov.qazws@gmail.com>
Date: Thu Feb 16 19:25:12 2023 +0300

    First commit

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/TPNN/pr1 (master)
$ git hist
* 4dbf97b 2023-02-17 | Revert "Some unwanted commit" (HEAD -> master) [Nikita Khitrov]
* a71c84b 2023-02-17 | Some unwanted commit [Nikita Khitrov]
* 08f0617 2023-02-16 | Third commit second.txt_v2 (tag: v2_1) [Nikita Khitrov]
* 15f5e7e 2023-02-16 | Second commit (tag: v1_2, tag: v1) [Nikita Khitrov]
* dbd5c18 2023-02-16 | First commit (tag: v1_1) [Nikita Khitrov]

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/TPNN/pr1 (master)
$

```

Рисунок 15 – Удаление последнего коммита

Часть 2. Системы управления репозиториями

Вариант 3

1. Создайте аккаунт на GitHub

На рисунке 16 показан созданный аккаунт на GitHub.

```
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ git revert HEAD --no-edit
[master 4dbf97b] Revert "Some unwanted commit"
Date: Fri Feb 17 00:53:42 2023 +0300
1 file changed, 1 deletion(-)
delete mode 100644 unwanted.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ git log
commit 4dbf97bc45e4019cfcae4e894f369f81f6217f7ed (HEAD -> master)
Author: Nikita Khitrov <hitrov.qazws@gmail.com>
Date: Fri Feb 17 00:53:42 2023 +0300

    Revert "Some unwanted commit"

    This reverts commit a71c84b99e5a0137dd39a2204ec39b540b862097.

commit a71c84b99e5a0137dd39a2204ec39b540b862097
Author: Nikita Khitrov <hitrov.qazws@gmail.com>
Date: Fri Feb 17 00:50:25 2023 +0300

    Some unwanted commit

commit 08f061794379ae77769489eee3a4efd2aaaddece (tag: v2_1)
Author: Nikita Khitrov <hitrov.qazws@gmail.com>
Date: Thu Feb 16 19:30:43 2023 +0300

    Third commit second.txt_v2

commit 15f5e7ef9a2ae1cd1a837b64c5d0239877f3e8c3 (tag: v1_2, tag: v1)
Author: Nikita Khitrov <hitrov.qazws@gmail.com>
Date: Thu Feb 16 19:29:02 2023 +0300

    Second commit

commit dbd5c18e1654b93efbbc0ab119d23d0c874fb163 (tag: v1_1)
Author: Nikita Khitrov <hitrov.qazws@gmail.com>
Date: Thu Feb 16 19:25:12 2023 +0300

    First commit

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$ git hist
* 4dbf97b 2023-02-17 | Revert "Some unwanted commit" (HEAD -> master) [Nikita Khitrov]
* a71c84b 2023-02-17 | Some unwanted commit [Nikita Khitrov]
* 08f0617 2023-02-16 | Third commit second.txt_v2 (tag: v2_1) [Nikita Khitrov]
* 15f5e7e 2023-02-16 | Second commit (tag: v1_2, tag: v1) [Nikita Khitrov]
* dbd5c18 2023-02-16 | First commit (tag: v1_1) [Nikita Khitrov]

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/пр1 (master)
$
```

Рисунок 16 – Аккаунт GitHub

2. Создайте репозиторий согласно варианту

На рисунке 17 показан процесс создания удалённого репозитория.

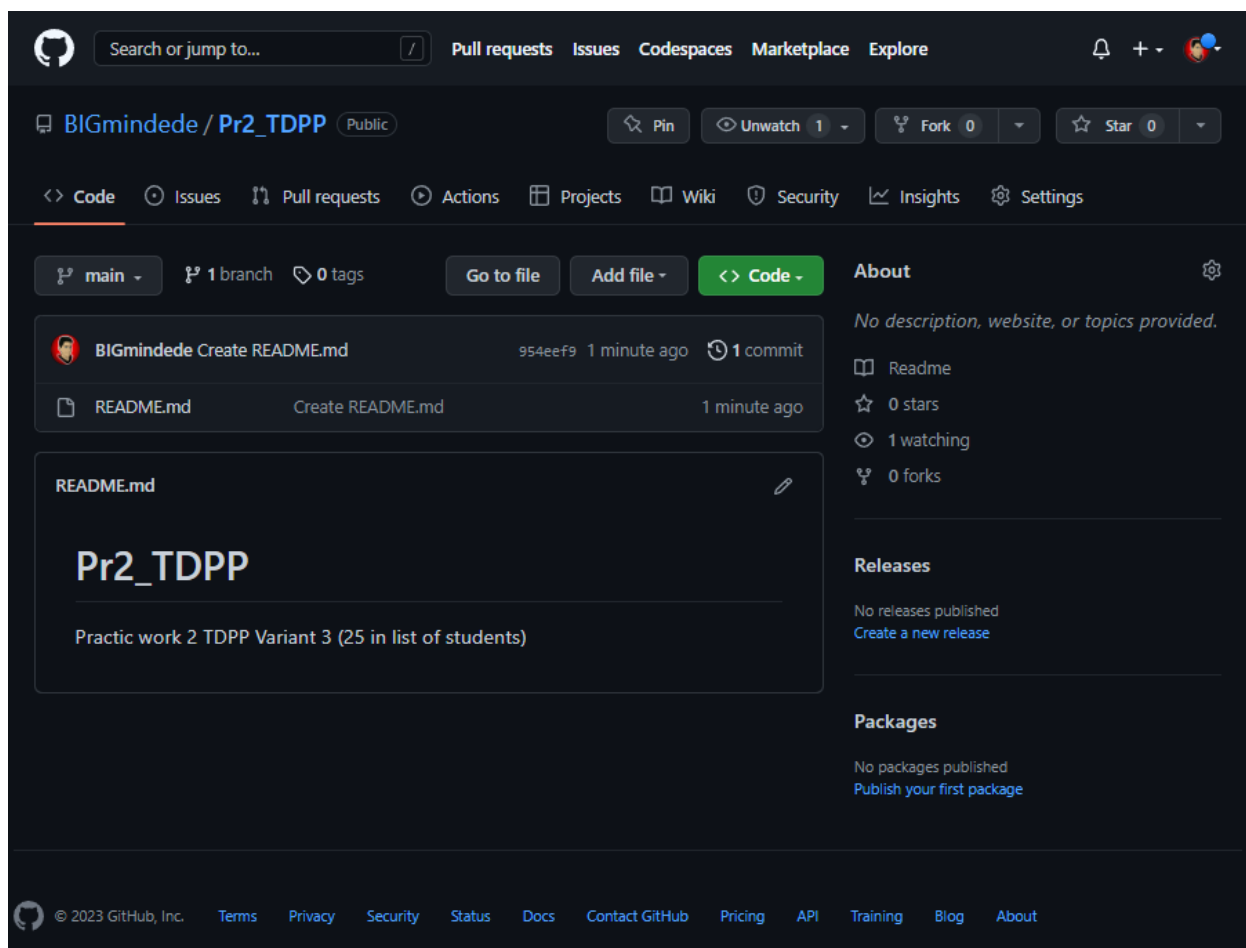


Рисунок 17 – Создание удалённого репозитория

3. Создайте новый локальный репозиторий с несколькими файлами на рабочей станции и загрузите его содержимое на GitHub

На рисунке 17 показан процесс загрузки существующего репозитория на GitHub.

```
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2
$ git remote add origin https://github.com/BIGminded/Pr2_TDPP
fatal: not a git repository (or any of the parent directories): .git

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2
$ git init
Initialized empty Git repository in C:/2year/ТПП/Pr1/Pr1_2/.git/

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (master)
$ git touch file_1.txt
git: 'touch' is not a git command. See 'git --help'.

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (master)
$ touch file_1.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (master)
$ touch file_2.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (master)
$ git add file_1.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (master)
$ git add file_2.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (master)
$ git commit -m "Added empty .txt files #1 and #2"
[master (root-commit) 956c106] Added empty .txt files #1 and #2
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file_1.txt
 create mode 100644 file_2.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (master)
$ git remote add origin https://github.com/BIGminded/Pr2_TDPP

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (master)
$ git push -u origin main
error: src refspec main does not match any
error: failed to push some refs to 'https://github.com/BIGminded/Pr2_TDPP'

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 244 bytes | 244.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/BIGminded/Pr2_TDPP/pull/new/master
remote:
To https://github.com/BIGminded/Pr2_TDPP
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (master)
$ |
```

Рисунок 17 – Загрузка репозитория на GitHub

4. Чтобы избежать ввода логина и пароля, создайте SSH-ключ для авторизации

На рисунке 18 показан процесс генерации SSH ключа.

```
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (master)
$ ssh-keygen -o
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/hitro/.ssh/id_rsa): ssh_key
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ssh_key
Your public key has been saved in ssh_key.pub
The key fingerprint is:
SHA256:P1H5KbG+LwwzaDNKyVC5Rear+eLbq6Xa0kqz1sAAIKM hitro@DESKTOP-GE42RLF
The key's randomart image is:
+---[RSA 3072]-----+
| =      oo      |
| +.    oo.    . |
| E      o.    +  |
| .      . . . . + . |
| o  o .S.. o o |
| o  +o=.+o . |
| oo .oo.oo=. |
| ..oo.o=   .o. |
| .o.o=*+++. .o. |
+-----[SHA256]-----+
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (master)
$ |
```

Рисунок 18 – Генерация SSH ключа

5. Создайте в репозитории новую ветку. Произведите в ней несколько изменений и слейте с веткой master

На рисунках 19-20 показан процесс создания новой ветки и слияние её с веткой master.

```
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (master)
$ git branch newBranch

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (master)
$ git checkout newBranch
Switched to branch 'newBranch'

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (newBranch)
$ echo "first change from branch [newBranch]" > file_1.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (newBranch)
$ echo "first change from branch [newBranch]" > file_2.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (newBranch)
$ git add file_1.txt
warning: in the working copy of 'file_1.txt', LF will be replaced by CRLF the next time Git touches it

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (newBranch)
$ git add file_2.txt
warning: in the working copy of 'file_2.txt', LF will be replaced by CRLF the next time Git touches it

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (newBranch)
$ git commit -m "Commit some changes from newBranch"
[newBranch 113eca4] Commit some changes from newBranch
 2 files changed, 2 insertions(+)

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (newBranch)
```

Рисунок 19 – Создание новой ветки и первый коммит в нее

```
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (newBranch)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (master)
$ git merge newBranch
Updating 956c106..113eca4
Fast-forward
 file_1.txt | 1 +
 file_2.txt | 1 +
 2 files changed, 2 insertions(+)

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (master)
$ .....
```

Рисунок 19 – Слияние новой ветки с веткой master

6. Клонировать непустой удаленный репозиторий на локальную машину

На рисунке 21 показан процесс клонирования удалённого репозитория.

```
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (master)
$ git clone https://github.com/BIGminded/Pr2_TDPP
Cloning into 'Pr2_TDPP'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 15 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (15/15), done.

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (master)
$
```

Рисунок 21 – Клонирование репозитория

7. Создайте тег указывающий на последний коммит в ветке

На рисунке 22 показан процесс создания тега, указывающего на последний коммит в ветке.

```
$ git log
commit 113eca40c6fe1ecec9fe0b4546c3d62ab2fa26f9 (HEAD -> master, newBranch)
Author: Nikita Khitrov <hitrov.qazws@gmail.com>
Date: Fri Feb 17 23:47:33 2023 +0300

    Commit some changes from newBranch

commit 956c106124c5129c1bf13fe6745c426992b933ad (origin/master)
Author: Nikita Khitrov <hitrov.qazws@gmail.com>
Date: Fri Feb 17 23:28:58 2023 +0300

    Added empty .txt files #1 and #2

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (master)
$ git tag v1

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (master)
$ git log
commit 113eca40c6fe1ecec9fe0b4546c3d62ab2fa26f9 (HEAD -> master, tag: v1, newBranch)
Author: Nikita Khitrov <hitrov.qazws@gmail.com>
Date: Fri Feb 17 23:47:33 2023 +0300

    Commit some changes from newBranch

commit 956c106124c5129c1bf13fe6745c426992b933ad (origin/master)
Author: Nikita Khitrov <hitrov.qazws@gmail.com>
Date: Fri Feb 17 23:28:58 2023 +0300

    Added empty .txt files #1 and #2

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (master)
$ |
```

Рисунок 22 – Создание тега, указывающего на последний коммит в ветке

8. Создайте новую ветку и выведите список всех веток

На рисунке 23 показан процесс создания новой ветки и вывода списка всех веток.

```
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (master)
$ git branch
* master
  newBranch
  newBranch2

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (master)
$ .....
```

Рисунок 23 – Создание новой ветки и вывод списка всех веток

9. Произведите 3 коммита в новой ветке

На рисунках 24-26 показан процесс производства трех коммитов в новой ветке.

```
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (master)
$ git checkout newBranch2
Switched to branch 'newBranch2'

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (newBranch2)
$ touch file_3.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (newBranch2)
$ echo "some code" > file_3.txt
```

```
$ git add file_3.txt
warning: in the working copy of 'file_3.txt', LF will be replaced by CRLF the next time Git touches it

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (newBranch2)
$ git commit -m "Added .txt file #3 with some code"
[newBranch2 a7e494c] Added .txt file #3 with some code
1 file changed, 1 insertion(+)
create mode 100644 file_3.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (newBranch2)
$ echo "second change in code of first file" > file_1.txt
```

```
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (newBranch2)
$ git add file_1.txt
warning: in the working copy of 'file_1.txt', LF will be replaced by CRLF the next time Git touches it

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (newBranch2)
$ git commit -m "Added changes in .txt file #1"
[newBranch2 8c25608] Added changes in .txt file #1
1 file changed, 1 insertion(+), 1 deletion(-)

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (newBranch2)
$ echo "Second change in code to second file" > file_2.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (newBranch2)
$ git add file_2.txt
warning: in the working copy of 'file_2.txt', LF will be replaced by CRLF the next time Git touches it

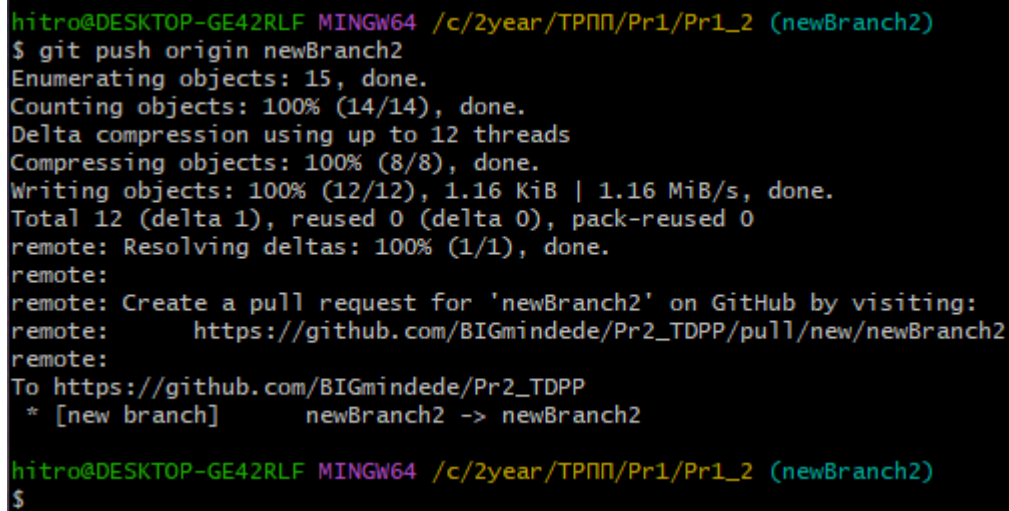
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (newBranch2)
$ git commit -m "Added changes in .txt file #2"
[newBranch2 a03ebf7] Added changes in .txt file #2
1 file changed, 1 insertion(+), 1 deletion(-)

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (newBranch2)
$
```

Рисунки 24-26 – Производство трех коммитов в новой ветке

10. Выгрузите все изменения в удаленный репозиторий

На рисунке 27 показана выгрузка изменений в удаленный репозиторий.



```
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (newBranch2)
$ git push origin newBranch2
Enumerating objects: 15, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 12 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (12/12), 1.16 KiB | 1.16 MiB/s, done.
Total 12 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
remote:
remote: Create a pull request for 'newBranch2' on GitHub by visiting:
remote:   https://github.com/BIGmindede/Pr2_TDPP/pull/new/newBranch2
remote:
To https://github.com/BIGmindede/Pr2_TDPP
 * [new branch]      newBranch2 -> newBranch2

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (newBranch2)
$
```

Рисунок 27 – Выгрузка изменений в удаленный репозиторий

11. Откатите ветку к созданному тегу (в том числе в удаленном репозитории)

На рисунке 28 показан процесс отката ветки к созданному тегу, в том числе в удаленном репозитории.

```

$ git checkout v1
Note: switching to 'v1'.

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 113eca4 Commit some changes from newBranch
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 ((v1))
$ git log
commit 113eca40c6fe1ecec9fe0b4546c3d62ab2fa26f9 (HEAD, tag: v1, origin/newBranch
, newBranch, master)
Author: Nikita Khitrov <hitrov.qazws@gmail.com>
Date:   Fri Feb 17 23:47:33 2023 +0300

    Commit some changes from newBranch

commit 956c106124c5129c1bf13fe6745c426992b933ad (origin/master)
Author: Nikita Khitrov <hitrov.qazws@gmail.com>
Date:   Fri Feb 17 23:28:58 2023 +0300

    Added empty .txt files #1 and #2

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 ((v1))
$ |

```

Рисунок 26 – откат ветки к созданному тэгу

12. Выведите в консоли различия между веткой master и новой веткой

На рисунке 29 показаны различия между веткой newBranch и master.

```
$ git diff master newBranch2
diff --git a/file_1.txt b/file_1.txt
index 7e7a0d6..673294c 100644
--- a/file_1.txt
+++ b/file_1.txt
@@ -1,1 @@
-first change from branch [newBranch]
+second change in code of first file
diff --git a/file_2.txt b/file_2.txt
index 7e7a0d6..a380606 100644
--- a/file_2.txt
+++ b/file_2.txt
@@ -1,1 @@
-first change from branch [newBranch]
+Second change in code to second file
diff --git a/file_3.txt b/file_3.txt
new file mode 100644
index 0000000..bb88ddf
--- /dev/null
+++ b/file_3.txt
@@ -0,0 +1 @@
+some code

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_2 (newBranch2)
$
```

Рисунок 29 – Сравнение веток branch2 и master

Часть 3. Работа с ветвлением и оформление кода

Вариант 5

1. Сделайте форк репозитория в соответствии с Вашим вариантом

На рисунке 30 показан процесс создания форка репозитория.

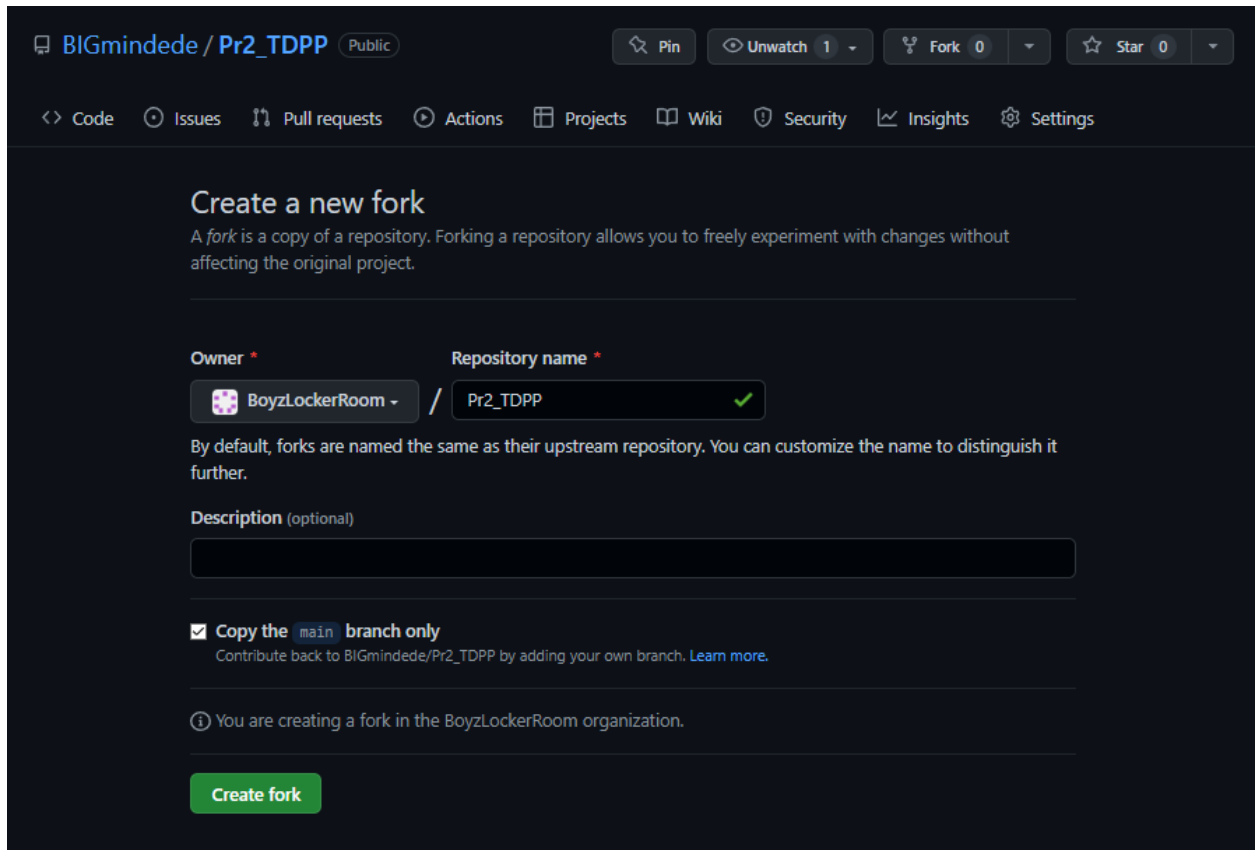


Рисунок 30 – Форк репозитория

2. Склонируйте его (форк репозитория) на локальную машину.

На рисунке 31 показан процесс клонирования репозитория на локальную машину.

```
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (master)
$ git clone https://github.com/BoyzLockerRoom/Pr2_TDPP.git
Cloning into 'Pr2_TDPP'...
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 13 (delta 0), reused 1 (delta 0), pack-reused 0
Receiving objects: 100% (13/13), 3.14 KiB | 3.14 MiB/s, done.
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (master)
$ |
```

Рисунок 31 – Клонирование репозитория на локальную машину

3. Создайте две ветки `branch1` и `branch2` от последнего коммита в `master`'е

На рисунке 32 показан процесс создания двух веток `branch1` и `branch2` от последнего коммита в `master`.

```
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (main)
$ git branch newBranch1

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (main)
$ git branch newBranch2

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (main)
$ git branch
* main
  newBranch1
  newBranch2
```

Рисунок 32 – Создание двух веток от последнего коммита

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

На рисунках 33 - 34 показаны 3 коммита на каждой из веток.

```
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch1)
$ dir
README.md  some\ first\ code.txt  some\ second\ code.txt  some\ third\ code.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch1)
$ nano README.md

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch1)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch1)
$ git commit -m "First commit from newBranch1 to README"
[newBranch1 63303e2] First commit from newBranch1 to README
1 file changed, 2 insertions(+), 1 deletion(-)

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch1)
$ nano README.md

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch1)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch1)
$ git commit -m "Second commit from newBranch1 to README"
[newBranch1 8a9b61e] Second commit from newBranch1 to README
1 file changed, 1 insertion(+), 1 deletion(-)

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch1)
$ nano README.md

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch1)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch1)
$ git commit -m "Third commit from newBranch1 to README"
[newBranch1 b97aae1] Third commit from newBranch1 to README
1 file changed, 1 insertion(+), 1 deletion(-)

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch1)
$
```

```

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch1)
$ git checkout newBranch2
Switched to branch 'newBranch2'
D   some third code.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch2)
$ nano README.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch2)
$ nano README.md

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch2)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch2)
$ git commit -m "First commit from newBranch2 to README"
[newBranch2 40ed373] First commit from newBranch2 to README
1 file changed, 2 insertions(+), 1 deletion(-)

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch2)
$ nano README.md

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch2)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch2)
$ git commit -m "Second commit from newBranch2 to README"
[newBranch2 e807aa5] Second commit from newBranch2 to README
1 file changed, 1 insertion(+), 1 deletion(-)

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch2)
$ nano README.md

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch2)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch2)
$ git commit -m "Third commit from newBranch2 to README"
[newBranch2 24d9def] Third commit from newBranch2 to README
1 file changed, 1 insertion(+), 1 deletion(-)

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch2)
$

```

Рисунки 33-34 – Создание 3-х коммитов на ветках

5. Выполните слияние ветки branch1 в ветку branch2, разрешив при этом конфликты.

На рисунке 35 показан процесс слияния ветки branch и branch2 и разрешения при этом конфликтов.

```

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch2)
$ git merge newBranch1
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch2|MERGING)
$ git status
On branch newBranch2
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:   README.md

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:         some third code.txt

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

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

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch2|MERGING)
$ git add .

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch2|MERGING)
$ git commit -m "Merge of newBranch1 and newBranch2 and fixing conflicts"
[newBranch2 5000f9c] Merge of newBranch1 and newBranch2 and fixing conflicts

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch2)
$

```

Рисунок 35 – Слияние двух веток

6. Выгрузите все изменения во всех ветках в удаленный репозиторий.

На рисунке 36 показан процесс выгрузки всех изменений в удалённый репозиторий.

```

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch2)
$ git push --set-upstream origin newBranch2
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 12 threads
Compressing objects: 100% (18/18), done.
Writing objects: 100% (21/21), 2.24 KiB | 1.12 MiB/s, done.
Total 21 (delta 7), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (7/7), completed with 1 local object.
remote:
remote: Create a pull request for 'newBranch2' on GitHub by visiting:
remote:   https://github.com/BoyzLockerRoom/Pr2_TDPP/pull/new/newBranch2
remote:
To https://github.com/BoyzLockerRoom/Pr2_TDPP.git
 * [new branch]      newBranch2 -> newBranch2
branch 'newBranch2' set up to track 'origin/newBranch2'.

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch2)
$ git push --set-upstream origin newBranch1
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'newBranch1' on GitHub by visiting:
remote:   https://github.com/BoyzLockerRoom/Pr2_TDPP/pull/new/newBranch1
remote:
To https://github.com/BoyzLockerRoom/Pr2_TDPP.git
 * [new branch]      newBranch1 -> newBranch1
branch 'newBranch1' set up to track 'origin/newBranch1'.

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch2)
$ |

```

Рисунок 36 – Выгрузка изменений в удалённый репозиторий

7. Проведите еще 3 коммита в ветку branch1

На рисунке 37 показан процесс создания 3-ёх коммитов на ветке branch1.

```

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch2)
$ git checkout newBranch1
Switched to branch 'newBranch1'
Your branch is up to date with 'origin/newBranch1'.

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch1)
$ nano README.md

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch1)
$ git add README.md

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch1)
$ git commit -m "One of three more new commits from newBranch1"
[newBranch1 549c15e] One of three more new commits from newBranch1
1 file changed, 1 insertion(+)

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch1)
$ nano README.md

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch1)
$ git add README.md

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch1)
$ git commit -m "Second of three more new commits from newBranch1"
[newBranch1 b7a9d07] Second of three more new commits from newBranch1
1 file changed, 1 insertion(+)

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch1)
$ nano README.md

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch1)
$ git add README.md

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch1)
$ git commit -m "Third of three more new commits from newBranch1"
[newBranch1 97f5dc2] Third of three more new commits from newBranch1
1 file changed, 1 insertion(+)

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch1)
$ |

```

Рисунок 37 – 3 коммита на ветке branch1

8. Склонируйте репозиторий еще раз в другую директорию

На рисунке 38 показан процесс клонирования репозитория в другую директорию.

```

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3_v2
$ git clone https://github.com/BoyzLockerRoom/Pr2_TDPP.git
Cloning into 'Pr2_TDPP'...
remote: Enumerating objects: 34, done.
remote: Counting objects: 100% (34/34), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 34 (delta 7), reused 23 (delta 7), pack-reused 0
Receiving objects: 100% (34/34), 5.24 KiB | 5.24 MiB/s, done.
Resolving deltas: 100% (7/7), done.

```

Рисунок 38 – Клонирование репозитория

9. В новом клоне репозитории сделайте 3 коммита в ветку branch1

На рисунке 39 показан процесс создания 3-ёх коммитов на ветке branch1 в новом репозитории.

```
hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3_v2/Pr2_TDPP (main)
$ git checkout newBranch1
Switched to a new branch 'newBranch1'
branch 'newBranch1' set up to track 'origin/newBranch1'.

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3_v2/Pr2_TDPP (newBranch1)
$ nano README.txt

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3_v2/Pr2_TDPP (newBranch1)
$ nano README.md

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3_v2/Pr2_TDPP (newBranch1)
$ git add README.md

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3_v2/Pr2_TDPP (newBranch1)
$ git commit -m "First commit to again cloned rep from branch1"
[newBranch1 535b857] First commit to again cloned rep from branch1
1 file changed, 1 insertion(+)

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3_v2/Pr2_TDPP (newBranch1)
$ nano README.md

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3_v2/Pr2_TDPP (newBranch1)
$ git add README.txt
fatal: pathspec 'README.txt' did not match any files

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3_v2/Pr2_TDPP (newBranch1)
$ git add README.md

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3_v2/Pr2_TDPP (newBranch1)
$ git commit -m "Second commit to again cloned rep from branch1"
[newBranch1 a921f03] Second commit to again cloned rep from branch1
1 file changed, 1 insertion(+)

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3_v2/Pr2_TDPP (newBranch1)
$ nano README.md

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3_v2/Pr2_TDPP (newBranch1)
$ git add README.md

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3_v2/Pr2_TDPP (newBranch1)
$ git commit -m "Third commit to again cloned rep from branch1"
[newBranch1 632d905] Third commit to again cloned rep from branch1
1 file changed, 1 insertion(+)

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3_v2/Pr2_TDPP (newBranch1)
$ |
```

Рисунок 39 – Создание коммитов в новом репозитории

10. Выгрузите все изменения из нового репозитория в удаленный репозиторий

На рисунке 40 показан процесс выгрузки изменений в удалённые репозиторий.

```

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3_v2/Pr2_TDPP (newBranch1)
$ git push
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 12 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 946 bytes | 946.00 KiB/s, done.
Total 9 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 1 local object.
To https://github.com/BoyzLockerRoom/Pr2_TDPP.git
    b97aae1..632d905  newBranch1 -> newBranch1

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3_v2/Pr2_TDPP (newBranch1)
$

```

Рисунок 40 – Выгрузка изменений в удалённый репозиторий

11. Вернитесь в старый клон с репозиторием, выгрузите изменения с опцией `–force`

На рисунке 41 показан процесс выгрузки изменений с тегом `–force`.

```

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch1)
$ git push --force
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 12 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 940 bytes | 940.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 1 local object.
To https://github.com/BoyzLockerRoom/Pr2_TDPP.git
    + 632d905...97f5dc2 newBranch1 -> newBranch1 (forced update)

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3 (newBranch1)
$

```

Рисунок 41 – Выгрузка в удалённый репозиторий с тегом `–force`

12. Получите все изменения в новом репозитории

На рисунке 42 показан процесс получения изменений из удалённого репозитория.


```

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3_v2/Pr2_TDPP (newBranch1)
$ git pull origin newBranch1
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 9 (delta 5), reused 9 (delta 5), pack-reused 0
Unpacking objects: 100% (9/9), 920 bytes | 51.00 KiB/s, done.
From https://github.com/BoyzLockerRoom/Pr2_TDPP
* branch                newBranch1 -> FETCH_HEAD
+ 632d905...97f5dc2 newBranch1 -> origin/newBranch1 (forced update)
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3_v2/Pr2_TDPP (newBranch1|MERGING)
$ git status
On branch newBranch1
Your branch and 'origin/newBranch1' have diverged,
and have 3 and 3 different commits each, respectively.
(use "git pull" to merge the remote branch into yours)

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:   README.md

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

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3_v2/Pr2_TDPP (newBranch1|MERGING)
$ git add .

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3_v2/Pr2_TDPP (newBranch1|MERGING)
$ git commit -m "merging local and remote reps and fixing conflicts"
[newBranch1 d64efa9] merging local and remote reps and fixing conflicts

hitro@DESKTOP-GE42RLF MINGW64 /c/2year/ТПП/Pr1/Pr1_3_v2/Pr2_TDPP (newBranch1)
$

```

Рисунок 42 – Получение изменений из удалённого репозитория

Ответы на вопросы

2. К какому типу систем контроля версий относится Git?

Git относится к распределённой системе контроле версий.

3. Что такое репозиторий Git?

Репозиторий – место, где система контроля версий хранит свои метаданные и базу данных объектов проекта.

Локальный репозиторий — репозиторий, расположенный на локальном компьютере разработчика в каталоге. Именно в нём происходит разработка и фиксация изменений, которые отправляются на удалённый репозиторий.

Удалённый репозиторий — репозиторий, находящийся на удалённом сервере. Это общий репозиторий, в который приходят все изменения и из которого забираются все обновления.

4. Что такое коммит?

Коммиты – основные конструктивные элементы временной шкалы Git. Их можно рассматривать как снимки состояния или контрольные точки на временной шкале проекта Git. Коммиты создаются с помощью команды `git commit`, которая делает снимок состояния проекта на текущий момент времени.

5. Что такое ветка в репозитории Git?

Ветка (Branch) — это параллельная версия репозитория. Она включена в этот репозиторий, но не влияет на главную версию, тем самым позволяя свободно работать в параллельной. Когда вы внесли нужные изменения, то вы можете объединить их с главной версией.

11. Что делает команда `git status`?

Команда `git status` отображает состояние рабочего каталога и раздела проиндексированных файлов. С её помощью можно проверить индексацию изменений и увидеть файлы, которые не отслеживаются Git.

12. Что делает команда `git add`?

Команда `git add` добавляет изменение из рабочего каталога в раздел проиндексированных файлов. Она сообщает Git, что вы хотите включить изменения в конкретном файле в следующий коммит.

18. Что сделает команда "`git branch`" без какого-либо параметра?

Команда `git branch` без параметров выведет список всех веток локального репозитория. Также она выделит зелёным цветом ветку, на которой находится указатель HEAD.

27. Как сделать ветку с названием `my_branch`?

Для создания ветки с именем `my_branch` необходимо написать команду `git branch my_branch` в консоль, либо можно написать команду `git checkout -b my_branch`, которая создаст новую ветку и переключит указатель HEAD на неё.

Выводы

В ходе данной работы был изучен функционал и возможности системы контроля версий git.