



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-20-23

Кузнецов Л. А.

Принял

Чернов Е. А.

Практическая работа
выполнена

«__»_____202__г.

(подпись
студента)

«Зачтено»

«__»_____202__г.

(подпись
руководителя)

Москва 2025

ОГЛАВЛЕНИЕ

ЗАДАНИЕ.....	4
Часть 1. Как начать работу с git'ом?.....	4
Шаг 1. Устанавливаем Git на локальную машину.....	4
Шаг 2. Настройка Git	4
Шаг 3. Начало работы – создание папки и файлов.....	5
Шаг 4. Создание репозитория	6
Шаг 5. Добавление файла в репозиторий и добавление первого коммита.....	6
Шаг 6. Индексация изменений.....	6
Шаг 6*. Индексация и коммит (не обязательный) – для ознакомления.....	6
Шаг 7. Коммиты нескольких изменений	7
Шаг 8. Просмотр истории коммитов.....	8
Шаг 9. Получение старых версий.....	9
Шаг 10. Отмена локальных изменений (до индексации)	11
Шаг 11. Отмена локальных изменений (после индексации и до коммита).....	12
Шаг 12. Отмена коммита	13
Часть 2. Управление репозиториями	14
Шаг 1. Создание аккаунта на GitHub	14
Шаг 2. Создание SSH-ключа для авторизации.....	14
Шаг 3. Создание нового репозитория для своего проекта.....	16
Шаг 4. Связываем локальный и удаленный репозитории.....	17
Шаг 5. Создание веток и переключение между ними	18
Шаг 6. Слияние веток	18
Часть 3. Ветвление и оформление кода.....	19
Шаг 1. Форк репозитория	19
Шаг 2. Клонирование репозитория на локальную машину	20
Шаг 3. Создание двух веток в master'e	21

Шаг 4. 3 коммита в каждую из веток, которые меняют один и тот же кусочек файла	21
Шаг 5. Слияние ветки branch1 в ветку branch2 с разрешением конфликтов.....	22
Шаг 6. Выгрузка всех изменений во всех ветках в удаленный репозиторий.....	22
Шаг 7. Выполнение 3-х коммитов в branch1	23
Шаг 8. Повторное клонирование репозитория в другую директорию.	23
Шаг 9. Выполнение 3-х коммитов в ветку branch1.....	24
Шаг 10. Выгрузка всех изменений в удаленный репозиторий	24
Шаг 11. Выгрузка изменения из старого репозитория с опцией --force.	24
Шаг 12. Получение всех изменения в новом репозитории	24
Часть 4. Индивидуальное задание №1	24
Часть 5. Индивидуальное задание №2.....	28
Часть 6. Ответы на вопросы.....	35
Вывод.....	36

ЗАДАНИЕ

Цель работы: получить навыки по работе с командной строкой и git'ом.

Часть 1. Основные команды Git

Шаг 1. Устанавливаем Git на локальную машину

Проведём работу по заданию.

Установили Git на Windows 10 (Рисунок 1).

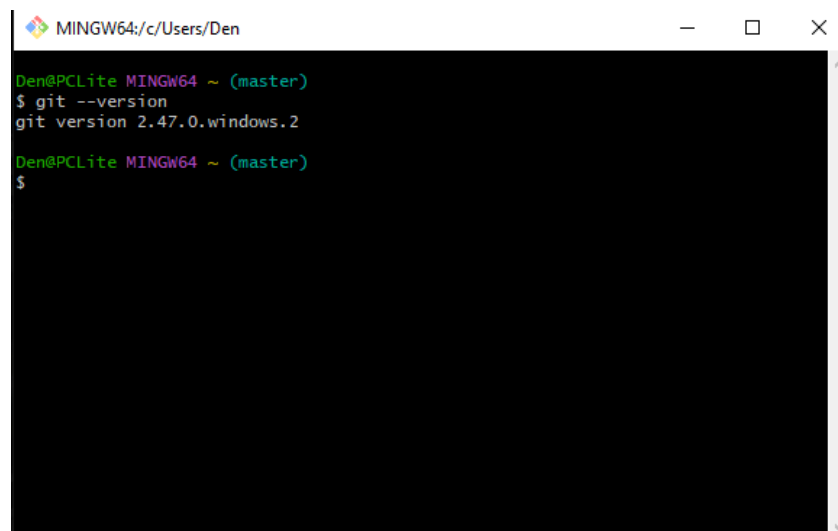
A screenshot of a terminal window titled 'MINGW64/c/Users/Den'. The prompt is 'Den@PCLite MINGW64 ~ (master)'. The user enters '\$ git --version' and the output is 'git version 2.47.0.windows.2'. The prompt returns to '\$'.

Рисунок 1 – Открытая консоль Git на Windows

Шаг 2. Настройка Git

Настроим Git в соответствии с требованиями (Рисунки 2-3).

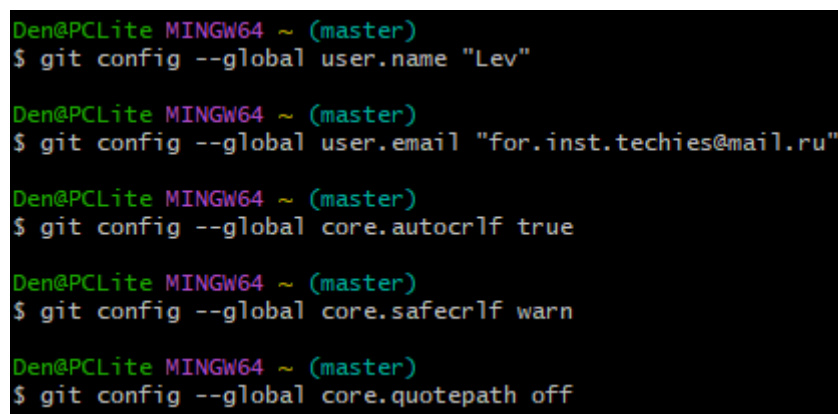
A screenshot of a terminal window titled 'MINGW64/c/Users/Den'. The prompt is 'Den@PCLite MINGW64 ~ (master)'. The user enters five commands: '\$ git config --global user.name "Lev"', '\$ git config --global user.email "for.inst.techies@mail.ru"', '\$ git config --global core.autocrlf true', '\$ git config --global core.safecrlf warn', and '\$ git config --global core.quotepath off'. Each command is followed by the prompt returning to '\$'.

Рисунок 2 – Ввод команд настройки Git-а

```
user.name=Lev
user.email=for.inst.techies@mail.ru

core.autocrlf=true
core.safecrlf=warn
core.quotepath=off
```

Рисунок 3 – Результат настройки на Git

Шаг 3. Начало работы – создание папки и файлов

На Рисунке 4 работает по заголовку. На Рисунке 5 изображён созданный файл.

```
Den@PCLite MINGW64 ~ (master)
$ mkdir gitTRPP

Den@PCLite MINGW64 ~ (master)
$ cd gitTRPP

Den@PCLite MINGW64 ~/gitTRPP (master)
$ touch proekt.html

Den@PCLite MINGW64 ~/gitTRPP (master)
$ nano proekt.html

Den@PCLite MINGW64 ~/gitTRPP (master)
$ ls
proekt.html
```

Рисунок 4 – Работа с папками

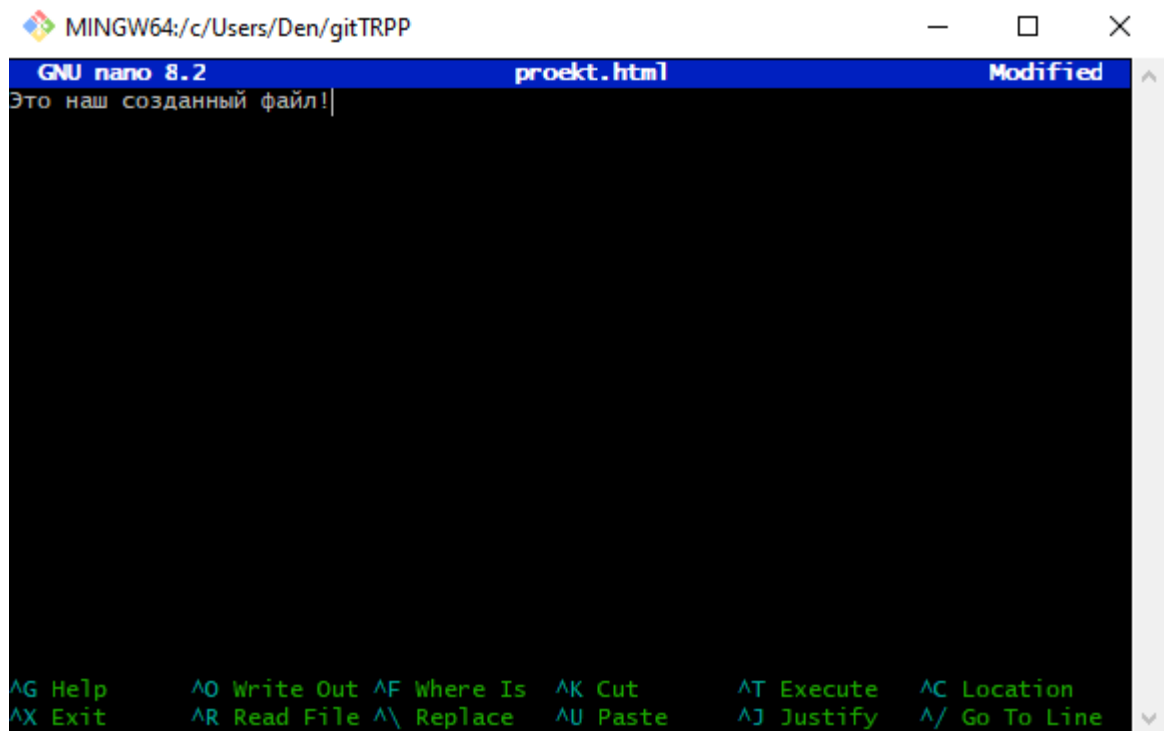


Рисунок 5 – Состав файла proekt.html

Шаг 4. Создание репозитория

```
Den@PCLite MINGW64 ~/gitTRPP (master)
$ git init
Initialized empty Git repository in C:/Users/Den/gitTRPP/.git/
```

Рисунок 6 – Создание репозитория

Шаг 5. Добавление файла в репозиторий и добавление первого коммита

```
Den@PCLite MINGW64 ~/gitTRPP (master)
$ git commit -m "smth"
[master (root-commit) 7513fb6] smth
1 file changed, 1 insertion(+)
create mode 100644 proekt.html

Den@PCLite MINGW64 ~/gitTRPP (master)
$
```

```
Den@PCLite MINGW64 ~/gitTRPP (master)
$ git status
On branch master
nothing to commit, working tree clean
```

Рисунок 7 – Добавление файла в репозиторий и добавление первого коммита

Шаг 6. Индексация изменений

```
Den@PCLite MINGW64 ~/gitTRPP (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

Den@PCLite MINGW64 ~/gitTRPP (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   proekt.html
```

Рисунок 8 – Фиксация изменений файла в Git

Шаг 6*. Индексация и коммит (не обязательный) – для ознакомления

```

Den@PCLite MINGW64 ~/gitTRPP (master)
$ git add a.file

Den@PCLite MINGW64 ~/gitTRPP (master)
$ git add b.file

Den@PCLite MINGW64 ~/gitTRPP (master)
$ git commit -m "Changes for a and b"
[master 0db89dd] Changes for a and b
3 files changed, 1 insertion(+)
create mode 100644 a.file
create mode 100644 b.file

Den@PCLite MINGW64 ~/gitTRPP (master)
$ git add c.file

Den@PCLite MINGW64 ~/gitTRPP (master)
$ git commit -m "Unrelated change to c"
[master d3cc681] Unrelated change to c
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 c.file

```

Рисунок 9 – Индексация и коммит

Шаг 7. Коммиты нескольких изменений

```

Den@PCLite MINGW64 ~/gitTRPP (master)
$ nano proekt.html

Den@PCLite MINGW64 ~/gitTRPP (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    a.file
        deleted:    b.file
        deleted:    c.file
        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

```

Рисунок 10 – 1 часть коммита нескольких изменений

```

Den@PCLite MINGW64 ~/gitTRPP (master)
$ git commit -m "Added standard HTML page tags"
[master 38746f8] Added standard HTML page tags
4 files changed, 6 insertions(+), 2 deletions(-)
delete mode 100644 a.file
delete mode 100644 b.file
delete mode 100644 c.file

Den@PCLite MINGW64 ~/gitTRPP (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 11 – 2 часть коммита нескольких изменений

```

Den@PCLite MINGW64 ~/gitTRPP (master)
$ git add .
warning: in the working copy of 'proekt.html', LF will be replaced by CRLF the n
ext time Git touches it

Den@PCLite MINGW64 ~/gitTRPP (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   proekt.html

Den@PCLite MINGW64 ~/gitTRPP (master)
$

```

Рисунок 12 – 3 часть коммита нескольких изменений

Шаг 8. Просмотр истории коммитов

```

commit 38746f8540dfa1089c9d785a5ba6b0b47f14fc5a (HEAD -> master)
Author: Lev <for.inst.techies@mail.ru>
Date:   Fri Feb 21 22:44:15 2025 +0300

    Added standard HTML page tags

commit d3cc68127fa8a8782be9489a2e1d6bf1f7f56365
Author: Lev <for.inst.techies@mail.ru>
Date:   Fri Feb 21 22:24:36 2025 +0300

    Unrelated change to c

commit 0db89dd63b0b2435dfffc41b2f4413f33d5030794
Author: Lev <for.inst.techies@mail.ru>
Date:   Fri Feb 21 22:22:58 2025 +0300

    Changes for a and b

commit 7013fdc7eed701c188951e76b8517dee14c01ae5
Author: Lev <for.inst.techies@mail.ru>
Date:   Fri Feb 21 22:12:11 2025 +0300

    something
:|

```

Рисунок 13 – 1 часть просмотра истории коммитов

```

Den@PCLite MINGW64 ~/gitTRPP (master)
$ git log --pretty=oneline
38746f8540dfa1089c9d785a5ba6b0b47f14fc5a (HEAD -> master) Added standard HTML pa
ge tags
d3cc68127fa8a8782be9489a2e1d6bf1f7f56365 Unrelated change to c
0db89dd63b0b2435dfffc41b2f4413f33d5030794 Changes for a and b
7013fdc7eed701c188951e76b8517dee14c01ae5 something
7513fb64b900cb902524032cae28f16f2ebbdda1 smth

```

Рисунок 14 – 2 часть просмотра истории коммитов


```

Den@PCLite MINGW64 ~/gitTRPP (master)
$ git log --pretty=oneline --max-count=2
git log --pretty=oneline --since='5 minutes ago'
git log --pretty=oneline --until='5 minutes ago'
git log --pretty=oneline --author=<your name>
git log --pretty=oneline --all
38746f8540dfa1089c9d785a5ba6b0b47f14fc5a (HEAD -> master) Added standard HTML pa
ge tags
d3cc68127fa8a8782be9489a2e1d6bf1f7f56365 Unrelated change to c
38746f8540dfa1089c9d785a5ba6b0b47f14fc5a (HEAD -> master) Added standard HTML pa
ge tags
d3cc68127fa8a8782be9489a2e1d6bf1f7f56365 Unrelated change to c
0db89dd63b0b2435dffc41b2f4413f33d5030794 Changes for a and b
7013fdc7eed701c188951e76b8517dee14c01ae5 something
7513fb64b900cb902524032cae28f16f2ebbdda1 smth
bash: syntax error near unexpected token `newline'
38746f8540dfa1089c9d785a5ba6b0b47f14fc5a (HEAD -> master) Added standard HTML pa
ge tags
d3cc68127fa8a8782be9489a2e1d6bf1f7f56365 Unrelated change to c
0db89dd63b0b2435dffc41b2f4413f33d5030794 Changes for a and b
7013fdc7eed701c188951e76b8517dee14c01ae5 something
7513fb64b900cb902524032cae28f16f2ebbdda1 smth

Den@PCLite MINGW64 ~/gitTRPP (master)
$ git log --pretty=format:"%h %ad | %s%d [%an]" --graph --date=short
* 38746f8 2025-02-21 | Added standard HTML page tags (HEAD -> master) [Lev]
* d3cc681 2025-02-21 | Unrelated change to c [Lev]
* 0db89dd 2025-02-21 | Changes for a and b [Lev]
* 7013fdc 2025-02-21 | something [Lev]
* 7513fb6 2025-02-21 | smth [Lev]

Den@PCLite MINGW64 ~/gitTRPP (master)
$

```

Рисунок 15 – 3 часть просмотра истории коммитов

Шаг 9. Получение старых версий

```

Den@PCLite MINGW64 ~/gitTRPP (master)
$ git log --pretty=format:"%h %ad | %s%d [%an]" --graph --date=short
* 38746f8 2025-02-21 | Added standard HTML page tags (HEAD -> master) [Lev]
* d3cc681 2025-02-21 | Unrelated change to c [Lev]
* 0db89dd 2025-02-21 | Changes for a and b [Lev]
* 7013fdc 2025-02-21 | something [Lev]
* 7513fb6 2025-02-21 | smth [Lev]

Den@PCLite MINGW64 ~/gitTRPP (master)
$ |

```

Рисунок 16 – Получение всех коммитов

```

Den@PCLite MINGW64 ~/gitTRPP (master)
$ git checkout 7513fb6
Note: switching to '7513fb6'.

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 7513fb6 smth
Den@PCLite MINGW64 ~/gitTRPP ((7513fb6...))
$

```

Рисунок 17 – Изменение файлов на старые версии

```

Den@PCLite MINGW64 ~/gitTRPP ((7513fb6...))
$ cat projekt.html
Это наш созданный файл!

```

Рисунок 18 – Проверка на изменения в файлах

```

Den@PCLite MINGW64 ~/gitTRPP ((7513fb6...))
$ git checkout master
Previous HEAD position was 7513fb6 smth
Switched to branch 'master'

Den@PCLite MINGW64 ~/gitTRPP (master)
$

```

Рисунок 19 – Смена версии на новую

```

Den@PCLite MINGW64 ~/gitTRPP ((7513fb6...))
$ git checkout master
Previous HEAD position was 7513fb6 smth
Switched to branch 'master'

Den@PCLite MINGW64 ~/gitTRPP (master)
$ cat projekt.html
<html>
<head>
</head>
<body>
<h1>Hello, World!</h1>
</body>
</html>

Den@PCLite MINGW64 ~/gitTRPP (master)
$ |

```

Рисунок 20 – Проверка возвращения к новейшей

Шаг 10. Отмена локальных изменений (до индексации)

```
Den@PCLite MINGW64 ~/gitTRPP (master)
$ git checkout master
Already on 'master'
```

Рисунок 21 – Проверка текущей версии

```
GNU nano 8.2                                proekt.html
<html>
<head>
</head>
<body>
<h1>Hello, World!</h1>
<!-- Вот это плохой комментарий. Мы его уберём -->
</body>
</html>
```

Рисунок 22 – Изменяем файл

```
Den@PCLite MINGW64 ~/gitTRPP (master)
$ nano proekt.html

Den@PCLite MINGW64 ~/gitTRPP (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
```

Рисунок 23 - Проверка наличия изменений

```
Den@PCLite MINGW64 ~/gitTRPP (master)
$ git checkout proekt.html
Updated 1 path from the index

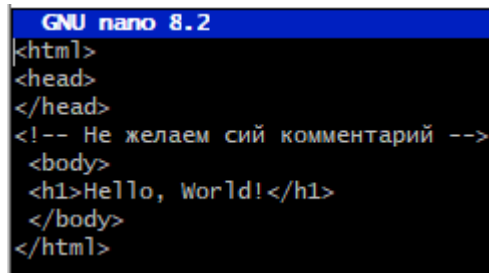
Den@PCLite MINGW64 ~/gitTRPP (master)
$ git status
On branch master
nothing to commit, working tree clean

Den@PCLite MINGW64 ~/gitTRPP (master)
$ cat proekt.html
<html>
<head>
</head>
<body>
<h1>Hello, World!</h1>
</body>
</html>

Den@PCLite MINGW64 ~/gitTRPP (master)
$ |
```

Рисунок 24 – Отмена изменений

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



```
GNU nano 8.2
<html>
<head>
</head>
<!-- Не желаем сий комментарий -->
<body>
<h1>Hello, World!</h1>
</body>
</html>
```

Рисунок 25 – Делаем изменения в файле

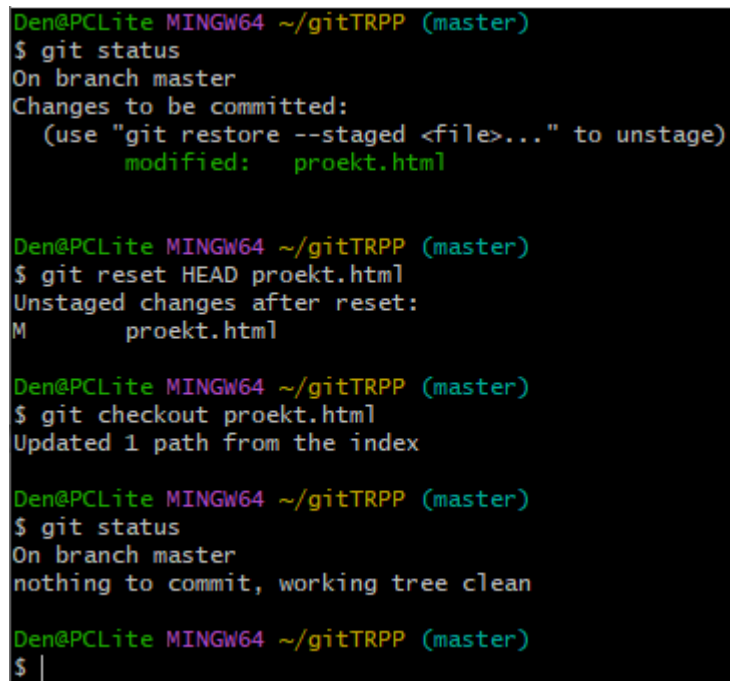


```
Den@PCLite MINGW64 ~/gitTRPP (master)
$ nano proekt.html

Den@PCLite MINGW64 ~/gitTRPP (master)
$ git add proekt.html

Den@PCLite MINGW64 ~/gitTRPP (master)
$
```

Рисунок 26 – Добавление коммита в рассмотрение изменений git-a



```
Den@PCLite MINGW64 ~/gitTRPP (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   proekt.html

Den@PCLite MINGW64 ~/gitTRPP (master)
$ git reset HEAD proekt.html
Unstaged changes after reset:
M   proekt.html

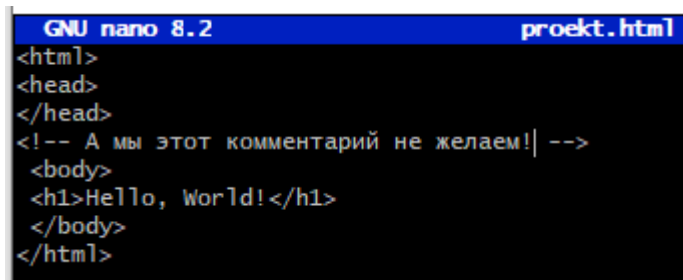
Den@PCLite MINGW64 ~/gitTRPP (master)
$ git checkout proekt.html
Updated 1 path from the index

Den@PCLite MINGW64 ~/gitTRPP (master)
$ git status
On branch master
nothing to commit, working tree clean

Den@PCLite MINGW64 ~/gitTRPP (master)
$ |
```

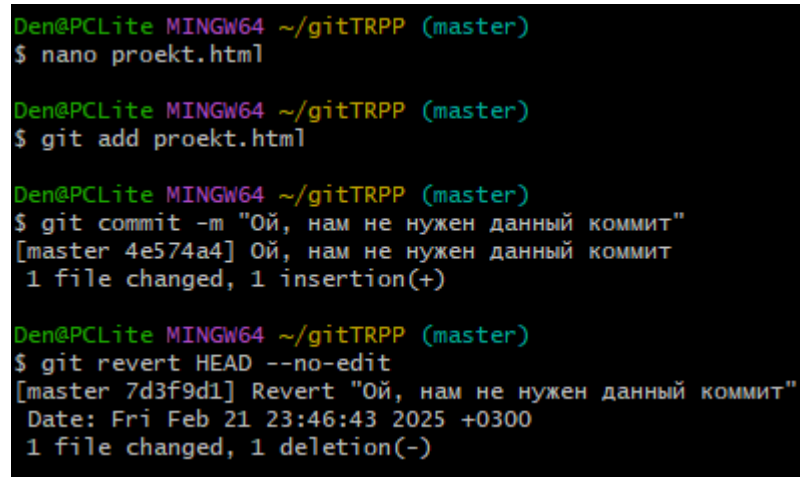
Рисунок 27 – Отмена изменения после индексации и до коммита

Шаг 12. Отмена коммита



```
GNU nano 8.2 proekt.html
<html>
<head>
</head>
<!-- А мы этот комментарий не желаем! -->
<body>
<h1>Hello, World!</h1>
</body>
</html>
```

Рисунок 28 – Изменение в файле proekt.html



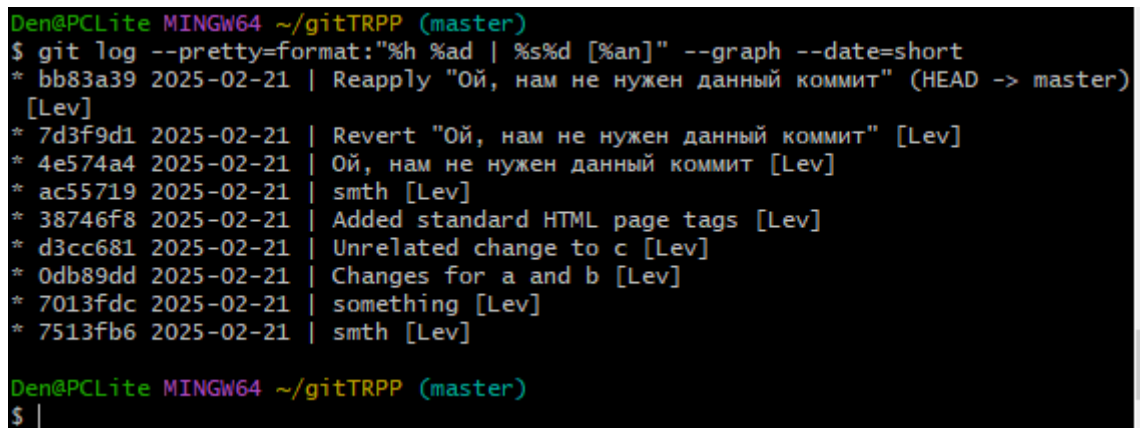
```
Den@PCLite MINGW64 ~/gitTRPP (master)
$ nano proekt.html

Den@PCLite MINGW64 ~/gitTRPP (master)
$ git add proekt.html

Den@PCLite MINGW64 ~/gitTRPP (master)
$ git commit -m "Ой, нам не нужен данный коммит"
[master 4e574a4] Ой, нам не нужен данный коммит
1 file changed, 1 insertion(+)

Den@PCLite MINGW64 ~/gitTRPP (master)
$ git revert HEAD --no-edit
[master 7d3f9d1] Revert "Ой, нам не нужен данный коммит"
Date: Fri Feb 21 23:46:43 2025 +0300
1 file changed, 1 deletion(-)
```

Рисунок 29 – Отмена коммита



```
Den@PCLite MINGW64 ~/gitTRPP (master)
$ git log --pretty=format:"%h %ad | %s%d [%an]" --graph --date=short
* bb83a39 2025-02-21 | Reapply "Ой, нам не нужен данный коммит" (HEAD -> master)
[Lev]
* 7d3f9d1 2025-02-21 | Revert "Ой, нам не нужен данный коммит" [Lev]
* 4e574a4 2025-02-21 | Ой, нам не нужен данный коммит [Lev]
* ac55719 2025-02-21 | smth [Lev]
* 38746f8 2025-02-21 | Added standard HTML page tags [Lev]
* d3cc681 2025-02-21 | Unrelated change to c [Lev]
* 0db89dd 2025-02-21 | Changes for a and b [Lev]
* 7013fdc 2025-02-21 | something [Lev]
* 7513fb6 2025-02-21 | smth [Lev]

Den@PCLite MINGW64 ~/gitTRPP (master)
$ |
```

Рисунок 30 – Проверка истории коммитов

Часть 2. Управление репозиториями

Шаг 1. Создание аккаунта на GitHub

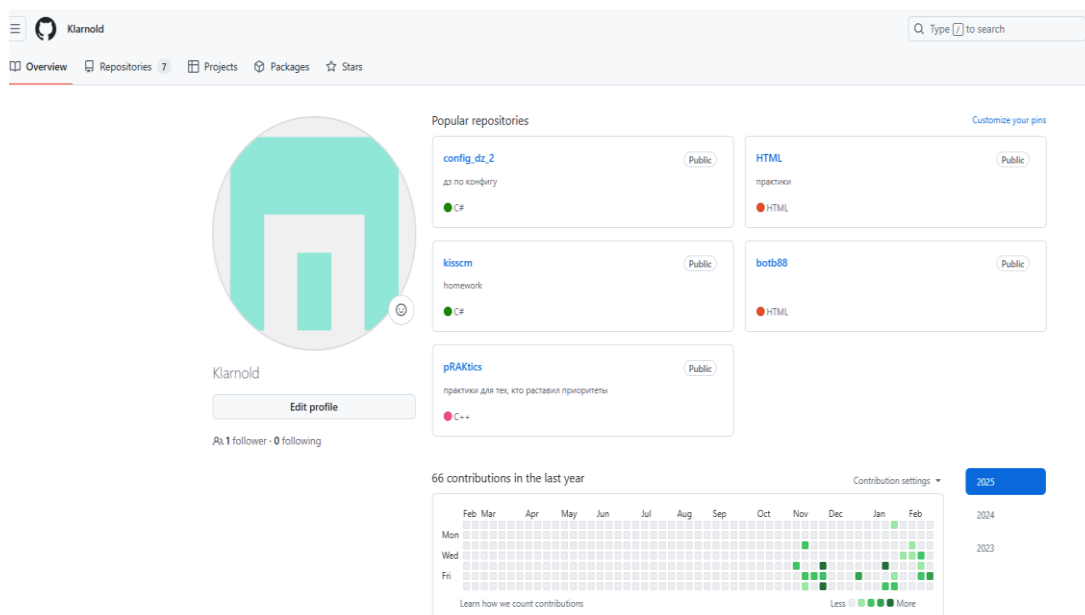


Рисунок 31 – Профиль аккаунта на GitHub-е

Шаг 2. Создание SSH-ключа для авторизации

```
Den@PCLite MINGW64 ~/gitTRPP (master)
$ ssh-keygen -t rsa -b 4096 -C "for.inst.techies@mail.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Den/.ssh/id_rsa):
Created directory '/c/Users/Den/.ssh'.
Enter passphrase for "/c/Users/Den/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Den/.ssh/id_rsa
Your public key has been saved in /c/Users/Den/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:t1Qm86LxkB035Q9ub5MXniIYWpmeDzPMBI4pAgntD8k for.inst.techies@mail.ru
The key's randomart image is:
+---[RSA 4096]---+
|..          . |
|...         o |
|oo . . . + = o |
|. E  + .ooO o o |
|. + o .S*= . o..|
|. o  BB+o ...+ |
|..Xo. . =+ |
|      = . ..o |
|      .      |
+---[SHA256]-----+
Den@PCLite MINGW64 ~/gitTRPP (master)
$
```

Рисунок 32 – Генерация SSH ключа

```

Den@PCLite MINGW64 ~/.ssh (master)
$ ssh-agent
SSH_AUTH_SOCK=/tmp/ssh-ix61kiPacBsw/agent.834; export SSH_AUTH_SOCK;
SSH_AGENT_PID=835; export SSH_AGENT_PID;
echo Agent pid 835;

Den@PCLite MINGW64 ~/.ssh (master)
$ eval "$(ssh-agent -s)"
Agent pid 841

Den@PCLite MINGW64 ~/.ssh (master)
$ ssh-add ~/.ssh/id_rsa
Identity added: /c/Users/Den/.ssh/id_rsa (for.inst.techies@mail.ru)

Den@PCLite MINGW64 ~/.ssh (master)
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDC0Uc1WKATDmuBtItIf39PsKcTZNCcbjzTSn62C0J
UwZgoF3xUbuJjfi3H4wfvBnXlcWaxgc0z+e5wMF4x03gTj4LU9sOLw6wx+Dp1NHwWACsx0/98gjpdcg2
Rzsu5H170xEN+61D70ZgxuqgWE5m4u9dJV0xW7jdAZQNLVo04RfItz//RUKWxEuFcxDu9ippPrTy1h8
YDq4TcCtIianlOnkafy2hYZEI4+i3mAg/YLL04XDZXiPB66Zndjr1gDxPQ8ZB34L3n2Dcwj+U/6zdnM
EOBtjOn12XMeKGA0mu3+YJS29F7VwHfoDiajIsb++h37a00jIvBuJ5Rnxn5dv9WFKXN30YgRYnjDuTI
6s/9jP5kN2cmRSwz9LANwSIP5HzbK1dVabiB+jBsSLy10em3PeQb4MjEIymBqg9h7p2gCC4DQqzHfOZi
J1ph1Fdyg+ttithj4hbyVo1XtOtWe9w95zMLgt/isWJ8g2pgiUJ/nahr4o0Dqg7U1ZG29PVY4LGNlr1/
XdfzHDGmhQImrddaQSh0kMnWHcIdXs+WCuzdgtmZqk++ht6+IRXay8+aGynjb7h+m10k2u8QAr3C1c4
xUHIFEBPWR+VKt0cRya+zNtj1pkfb9qnNQaXL+RjTey2RbYu8o5+HGRbHYUSSYxJLB/pB/aU/J8jUdOJ
PQ== for.inst.techies@mail.ru

Den@PCLite MINGW64 ~/.ssh (master)
$

```

Рисунок 33 – Получение SSH ключа

Klarnold (Klarnold)
Your personal account

Go to your personal profile

- Public profile
- Account
- Appearance
- Accessibility
- Notifications
- Access
 - Billing and plans
 - Emails
 - Password and authentication
 - Sessions
 - SSH and GPG keys**
 - Organizations
 - Enterprises
 - Moderation

Add new SSH Key


Title
project

Key type
Authentication Key

Key
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDC0Uc1WKATDmuBtItIf39PsKcTZNCcbjzTSn62C0J/UwZgoF3xUbuJjfi3H4wfvBnXlcWaxgc0z+e5wMF4x03gTj4LU9sOLw6wx+Dp1NHwWACsx0/98gjpdcg2Rzsu5H170xEN+61D70ZgxuqgWE5m4u9dJV0xW7jdAZQNLVo04RfItz//RUKWxEuFcxDu9ippPrTy1h8YDq4TcCtIianlOnkafy2hYZEI4+i3mAg/YLL04XDZXiPB66Zndjr1gDxPQ8ZB34L3n2Dcwj+U/6zdnMEOBtjOn12XMeKGA0mu3+YJS29F7VwHfoDiajIsb++h37a00jIvBuJ5Rnxn5dv9WFKXN30YgRYnjDuTI6s/9jP5kN2cmRSwz9LANwSIP5HzbK1dVabiB+jBsSLy10em3PeQb4MjEIymBqg9h7p2gCC4DQqzHfOZiJ1ph1Fdyg+ttithj4hbyVo1XtOtWe9w95zMLgt/isWJ8g2pgiUJ/nahr4o0Dqg7U1ZG29PVY4LGNlr1/XdfzHDGmhQImrddaQSh0kMnWHcIdXs+WCuzdgtmZqk++ht6+IRXay8+aGynjb7h+m10k2u8QAr3C1c4xUHIFEBPWR+VKt0cRya+zNtj1pkfb9qnNQaXL+RjTey2RbYu8o5+HGRbHYUSSYxJLB/pB/aU/J8jUdOJPQ== for.inst.techies@mail.ru

Add SSH key

Рисунок 34 – 1 часть добавления SSH ключа на GitHub



Klarnold (Klarnold)
Your personal account

[Go to your personal profile](#)

- Public profile
- Account
- Appearance
- Accessibility
- Notifications

Access


- Billing and plans
- Emails
- Password and authentication

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys




project
SHA256: t1Qm86Lxk8035Q9ub5MXniIYWpmeDzPMBI4pAgntD8k
Added on Feb 22, 2025
Never used — Read/write

Delete

Check out our guide to [connecting to GitHub using SSH keys](#) or troubleshoot [common SSH problems](#).

Рисунок 35 – 2 часть добавления SSH ключа на GitHub


Шаг 3. Создание нового репозитория для своего проекта


config_dz_2
Public

[Pin](#)
[Unwatch](#)
1

main
2 Branches
0 Tags

Go to file
Add file
Code


Klarnold
Add files via upload
f900079 · 3 months ago
2 Commits

Конфиг	Add files via upload	3 months ago
README.md	Initial commit	3 months ago

README

config_dz_2

дз по конфигу

Рисунок 36 – Создание нового репозитория

Шаг 4. Связываем локальный и удаленный репозитории

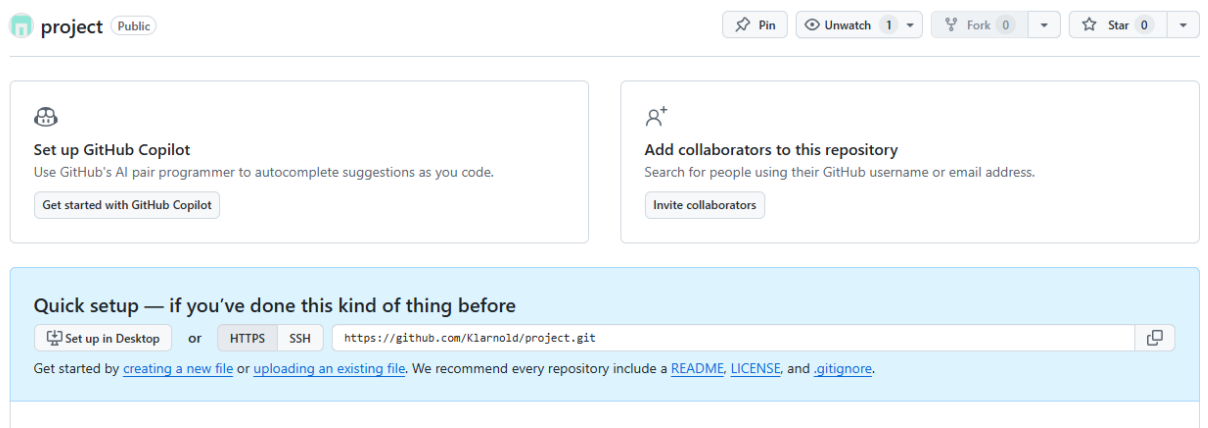


Рисунок 37 – Создание нового репозитория

```
Den@PCLite MINGW64 ~/.ssh (master)
$ git remote add project git@github.com:Klarnold/project.git

Den@PCLite MINGW64 ~/.ssh (master)
$ |
```

Рисунок 38 – Связь локального и удалённого репозитория

Шаг 5. Создание веток и переключение между ними

```
Den@PCLite MINGW64 ~/gitTRPP (main)
$ git status
On branch main
nothing to commit, working tree clean

Den@PCLite MINGW64 ~/gitTRPP (main)
$
```

Рисунок 39 – Проверка основной ветки

```
GNU nano 8.2      projekt.html
<html>
<head>
</head>
<body>
  <h1>Hello, World! I'll be there for you</h1>
</body>
</html>
```

Рисунок 40 – Изменение файла

```

Den@PCLite MINGW64 ~/gitTRPP (main)
$ nano proekt.html

Den@PCLite MINGW64 ~/gitTRPP (main)
$ git add proekt.html

Den@PCLite MINGW64 ~/gitTRPP (main)
$ git commit -m "Changes of proekt"
[main 6a1efd7] Changes of proekt
1 file changed, 2 insertions(+), 3 deletions(-)

Den@PCLite MINGW64 ~/gitTRPP (main)
$

```

Рисунок 41 – Индексация и коммит изменений

Шаг 6. Слияние веток

MINGW64:/c/Users/Den/gitTRPP

```

GNU nano 8.2      proekt.html
<html>
<head>
</head>
<body>
  <h1> GGGG Hello, World! I'll be there for you</h1>
</body>
</html>

```

Рисунок 42 – Изменение исходного файла

```

Den@PCLite MINGW64 ~/gitTRPP (something)
$ git checkout main
Switched to branch 'main'

Den@PCLite MINGW64 ~/gitTRPP (main)
$ nano proekt.html

Den@PCLite MINGW64 ~/gitTRPP (main)
$ git merge something
Updating 6a1efd7..e1ed9f2
Fast-forward
 proekt.html | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

```

Рисунок 43 – Слияние веток

Часть 3. Ветвление и оформление кода


Шаг 1. Форк репозитория

Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk (*).

Owner * Repository name *

 Klarnold / sdf


✓ sdf is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

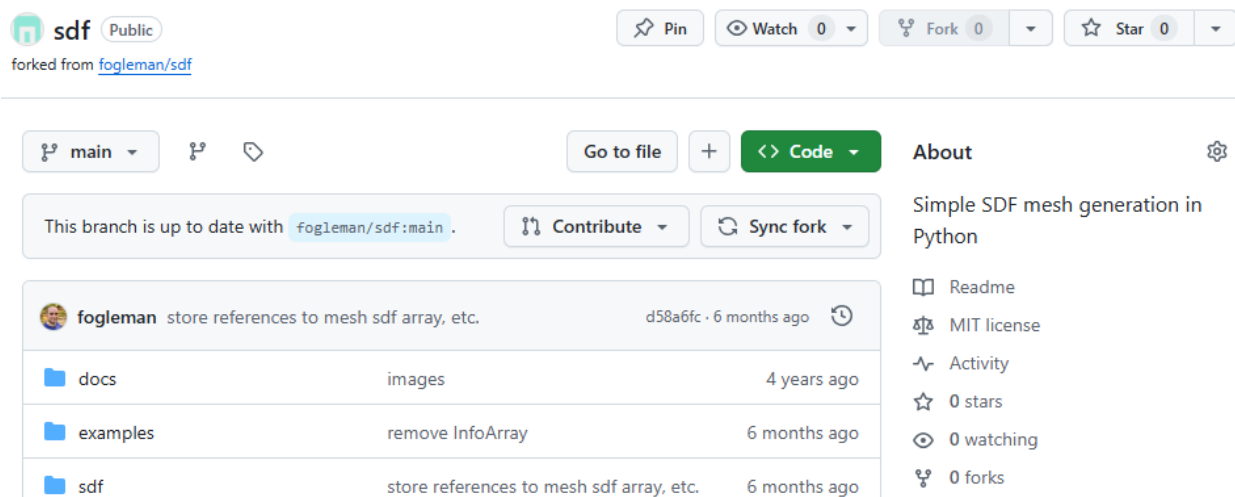
Simple SDF mesh generation in Python


☒ Copy the `main` branch only
Contribute back to fogleman/sdf by adding your own branch. [Learn more.](#)

 You are creating a fork in your personal account.





Create fork




Рисунок 44 – Форк репозитория






 sdf Public

forked from [fogleman/sdf](#)

 Pin  Watch 0  Fork 0  Star 0

 main  Go to file  Code

This branch is up to date with [fogleman/sdf:main](#).  Contribute  Sync fork

 fogleman store references to mesh sdf array, etc. d58a6fc · 6 months ago

docs	images	4 years ago
examples	remove InfoArray	6 months ago
sdf	store references to mesh sdf array, etc.	6 months ago

About

Simple SDF mesh generation in Python







 Readme  MIT license  Activity  0 stars  0 watching  0 forks

Рисунок 45 – Репозиторий после форка

Шаг 2. Клонирование репозитория на локальную машину

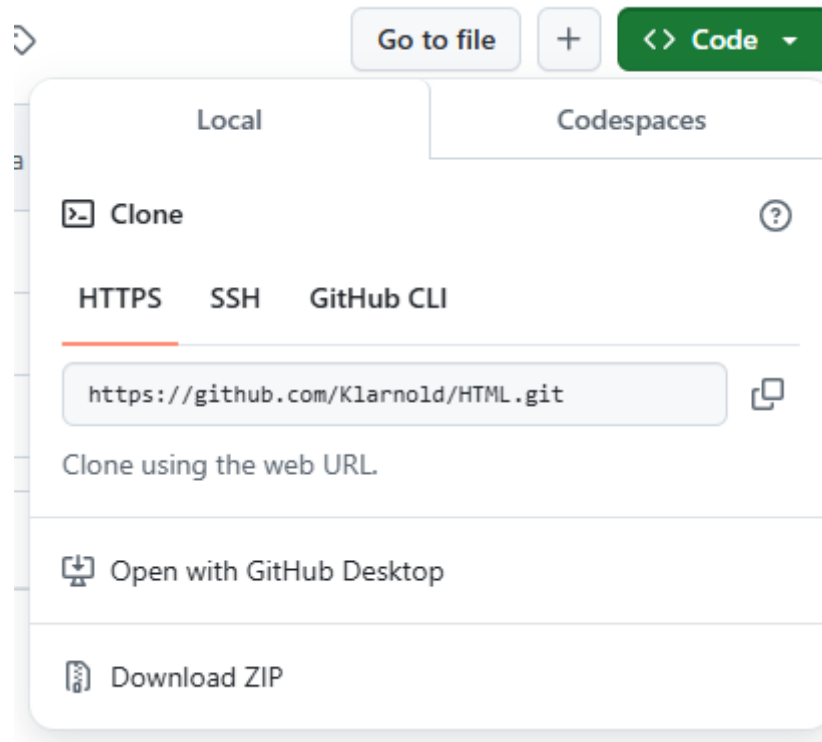


Рисунок 46 – Ссылка на клонируемы репозиторий

```
Den@PCLite MINGW64 ~/gitTRPP (main)
$ git clone https://github.com/Klarnold/HTML.git
Cloning into 'HTML'...
remote: Enumerating objects: 34, done.
remote: Counting objects: 100% (34/34), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 34 (delta 3), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (34/34), 2.89 MiB | 4.37 MiB/s, done.
Resolving deltas: 100% (3/3), done.

Den@PCLite MINGW64 ~/gitTRPP (main)
$
```

Рисунок 47 – Производим процесс клонирования

Имя	Дата и время	Тип	Размер
HTML	22.02.2025 10:01	Папка с файлами	
proekt.html	22.02.2025 9:40	Chrome HTML Do...	1 КБ
smthFile	21.02.2025 23:13	Файл	0 КБ

Рисунок 48 – Результат клонирования

Шаг 3. Создание двух веток в master'e

```
Den@PCLite MINGW64 ~/gitTRPP (main)
$ git checkout master
Switched to branch 'master'

Den@PCLite MINGW64 ~/gitTRPP (master)
$ git branch first

Den@PCLite MINGW64 ~/gitTRPP (master)
$ git branch second

Den@PCLite MINGW64 ~/gitTRPP (master)
$
```

Рисунок 49 - Создание двух веток в master'e

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

```
Den@PCLite MINGW64 ~/gitTRPP (main)
$ git add proekt.html

Den@PCLite MINGW64 ~/gitTRPP (main)
$ git commit -m "s"
[main 779ee15] s
1 file changed, 1 insertion(+), 1 deletion(-)

Den@PCLite MINGW64 ~/gitTRPP (main)
$ git checkout first
Switched to branch 'first'

Den@PCLite MINGW64 ~/gitTRPP (first)
$ nano proekt.html

Den@PCLite MINGW64 ~/gitTRPP (first)
$ git add proekt.html

Den@PCLite MINGW64 ~/gitTRPP (first)
$ git commit -m "first"
[first 57b90c7] first
1 file changed, 1 insertion(+), 1 deletion(-)

Den@PCLite MINGW64 ~/gitTRPP (first)
$ git checkout second
Switched to branch 'second'

Den@PCLite MINGW64 ~/gitTRPP (second)
$ nano proekt.html

Den@PCLite MINGW64 ~/gitTRPP (second)
$ git add proekt.html

Den@PCLite MINGW64 ~/gitTRPP (second)
$ git commit -m "second"
[second fa44d25] second
1 file changed, 1 insertion(+), 1 deletion(-)

Den@PCLite MINGW64 ~/gitTRPP (second)
$ |
```

Рисунок 50 - 3 коммита в каждую из веток, которые меняют один и тот же кусочек файла

Шаг 5. Слияние ветки branch1 в ветку branch2 с разрешением конфликтов

```
Den@PCLite MINGW64 ~/gitTRPP (main)
$ git merge second
Auto-merging proekt.html
CONFLICT (content): Merge conflict in proekt.html
Automatic merge failed; fix conflicts and then commit the result.
```

Рисунок 51 – Возникновение конфликта версий

```
Den@PCLite MINGW64 ~/gitTRPP (main|MERGING)
$ git checkout --ours proekt.html
Updated 1 path from the index

Den@PCLite MINGW64 ~/gitTRPP (main|MERGING)
$ git checkout --theirs proekt.html
Updated 1 path from the index

Den@PCLite MINGW64 ~/gitTRPP (main|MERGING)
$ git add proekt.html

Den@PCLite MINGW64 ~/gitTRPP (main|MERGING)
$ git commit -m "added theirs"
[main 76a5376] added theirs

Den@PCLite MINGW64 ~/gitTRPP (main)
$ git merge second
Already up to date.

Den@PCLite MINGW64 ~/gitTRPP (main)
$ |
```

Рисунок 52 – Решение конфликта выбором версии (остаётся second)

Шаг 6. Выгрузка всех изменений во всех ветках в удаленный репозиторий

```
Den@PCLite MINGW64 ~/gitTRPP/HTML (main)
$ git push origin main
Everything up-to-date

Den@PCLite MINGW64 ~/gitTRPP/HTML (main)
$
```

Рисунок 53 – Отправка изменений в удалённый репозиторий

Шаг 7. Выполнение 3-х коммитов в branch1 (Часть 1. Шаг 5)

```
Den@PCLite MINGW64 ~/gitTRPP (main)
$ git checkout -b branch1
Switched to a new branch 'branch1'

Den@PCLite MINGW64 ~/gitTRPP (branch1)
$ nano p

Den@PCLite MINGW64 ~/gitTRPP (branch1)
$ nano projekt.html

Den@PCLite MINGW64 ~/gitTRPP (branch1)
$ git add projekt.html

Den@PCLite MINGW64 ~/gitTRPP (branch1)
$ git commit -m "Commit1"
[branch1 a284ed6] Commit1
1 file changed, 1 insertion(+), 1 deletion(-)

Den@PCLite MINGW64 ~/gitTRPP (branch1)
$ nano projekt.html

Den@PCLite MINGW64 ~/gitTRPP (branch1)
$ git add projekt.html

Den@PCLite MINGW64 ~/gitTRPP (branch1)
$ git commit -m "Commit1"
[branch1 7a0ecf3] Commit1
1 file changed, 1 insertion(+), 1 deletion(-)

Den@PCLite MINGW64 ~/gitTRPP (branch1)
$ nano projekt.html

Den@PCLite MINGW64 ~/gitTRPP (branch1)
$ git add projekt.html

Den@PCLite MINGW64 ~/gitTRPP (branch1)
$ git commit -m "Commit1"
[branch1 4b9c684] Commit1
1 file changed, 1 insertion(+), 1 deletion(-)

Den@PCLite MINGW64 ~/gitTRPP (branch1)
$
```

Рисунок 54 - Выполнение 3-х коммитов в branch1

Шаг 8. Повторное клонирование репозитория в другую директорию

```
Den@PCLite MINGW64 ~/gitTRPP (branch1)
$ cd HTML

Den@PCLite MINGW64 ~/gitTRPP/HTML (main)
$ git clone https://github.com/Klarnold/HTML.git
Cloning into 'HTML'...
remote: Enumerating objects: 34, done.
remote: Counting objects: 100% (34/34), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 34 (delta 3), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (34/34), 2.89 MiB | 4.07 MiB/s, done.
Resolving deltas: 100% (3/3), done.

Den@PCLite MINGW64 ~/gitTRPP/HTML (main)
$
```

Рисунок 55 – Повторное клонирование репозитория в другую директорию

Шаг 9. Выполнение 3-х коммитов в ветку branch1

Аналогично шагу 7.

Шаг 10. Выгрузка всех изменений в удаленный репозиторий

```
Den@PCLite MINGW64 ~/gitTRPP/HTML (main)
$ git push origin main
Everything up-to-date
```

Рисунок 56 - Выгрузка всех изменений в удаленный репозиторий

Шаг 11. Выгрузка изменения из старого репозитория с опцией –force

```
Den@PCLite MINGW64 ~/gitTRPP/HTML (main)
$ git push --force origin main
Everything up-to-date
```

Рисунок 57 - Выгрузка изменения из старого репозитория с опцией – force

Шаг 12. Получение всех изменения в новом репозитории

```
Den@PCLite MINGW64 ~/gitTRPP/HTML (main)
$ git pull https://github.com/Klarnold/HTML.git
From https://github.com/Klarnold/HTML
* branch          HEAD      -> FETCH_HEAD
Already up to date.

Den@PCLite MINGW64 ~/gitTRPP/HTML (main)
$ |
```

Рисунок 58 - Получение всех изменений в новом репозитории

Часть 4. Индивидуальное задание №1

```
Den@PCLite MINGW64 ~/TRPP (main)
$ git clone https://github.com/Klarnold/project.git
Cloning into 'project'...
remote: Enumerating objects: 45, done.
remote: Counting objects: 100% (45/45), done.
remote: Compressing objects: 100% (31/31), done.
remote: Total 45 (delta 7), reused 39 (delta 6), pack-reused 0 (from 0)
Receiving objects: 100% (45/45), 5.60 KiB | 1.87 MiB/s, done.
Resolving deltas: 100% (7/7), done.
```

Рисунок 59 – Клонировем удалённый репозиторий


```

Den@PCLite MINGW64 ~/TRPP/project (main)
$ git branch br1

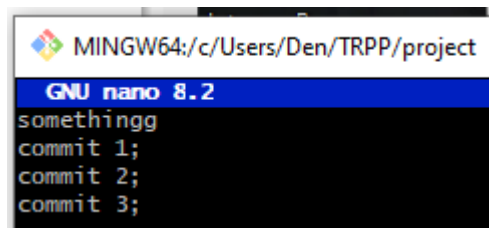
Den@PCLite MINGW64 ~/TRPP/project (main)
$ git branch br2

Den@PCLite MINGW64 ~/TRPP/project (main)
$ git branch -l
  br1
  br2
* main

Den@PCLite MINGW64 ~/TRPP/project (main)
$

```

Рисунок 60 – Создаём две ветви



```

MINGW64:/c/Users/Den/TRPP/project
GNU nano 8.2
somethingg
commit 1;
commit 2;
commit 3;

```

Рисунок 61 – Итог трёх коммитов

```

commit 6c4355c4314bc1d19b375f9c9d92104aa36bfb5a (HEAD -> br1)
Author: Lev <for.inst.techies@mail.ru>
Date: Sat Feb 22 22:02:13 2025 +0300

    third

commit a00779b0e8e36d593e44ec7ba23f9c55f738827b
Author: Lev <for.inst.techies@mail.ru>
Date: Sat Feb 22 22:00:57 2025 +0300

    second

commit c7a2dbb3e75cf4808963fb3785dfbf62492db676
Author: Lev <for.inst.techies@mail.ru>
Date: Sat Feb 22 22:00:43 2025 +0300

    first

commit c87c8134f0f1ea7798eb10a6afaab3febcb56991 (origin/main, origin/HEAD, main, br2)
Author: Klarnold <141218904+Klarnold@users.noreply.github.com>
Date: Sat Feb 22 10:00:02 2025 +0300

```

Рисунок 62 – Просмотр версий br1

```

Den@PCLite MINGW64 ~/TRPP/project (br1)
$ git checkout br2
Switched to branch 'br2'

Den@PCLite MINGW64 ~/TRPP/project (br2)
$ git cherry-pick c7a2dbb3e75cf4808963fb3785dfbf62492db676
[br2 d07030b] first
Date: Sat Feb 22 22:00:43 2025 +0300
1 file changed, 1 insertion(+)

Den@PCLite MINGW64 ~/TRPP/project (br2)
$ |

```

Рисунок 63 – Переводим первый коммит с br1 на br2

```

Den@PCLite MINGW64 ~/TRPP/project (br2)
$ git log
commit d07030b005e79264c516cbf4ddc06b7f6af01eb5 (HEAD -> br2)
Author: Lev <for.inst.techies@mail.ru>
Date: Sat Feb 22 22:00:43 2025 +0300

    first

```

Рисунок 64 – Проверка версий br2

```

Den@PCLite MINGW64 ~/TRPP/project (br2)
$ git push origin br2
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 20 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 350 bytes | 350.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'br2' on GitHub by visiting:
remote:   https://github.com/Klarnold/project/pull/new/br2
remote:
To https://github.com/Klarnold/project.git
 * [new branch]      br2 -> br2

Den@PCLite MINGW64 ~/TRPP/project (br2)
$ git checkout br1
Switched to branch 'br1'

Den@PCLite MINGW64 ~/TRPP/project (br1)
$ git push origin br1
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 20 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 984 bytes | 984.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'br1' on GitHub by visiting:
remote:   https://github.com/Klarnold/project/pull/new/br1
remote:
To https://github.com/Klarnold/project.git
 * [new branch]      br1 -> br1

Den@PCLite MINGW64 ~/TRPP/project (br1)
$ |

```

Рисунок 65 – Выгрузка изменений в удалённый репозиторий

```

GNU nano 8.2      smth
somethingg
commit 1;
commit 2;
commit 3;

```

Рисунок 66 – Файл smth в br1

```

GNU nano 8.2      smth
somethingg
commit 1;

```

Рисунок 67 – Файл smth в br2

```

GNU nano 8.2      smth
somethingg

```

Рисунок 68 – Файл smth в main

```

Den@PCLite MINGW64 ~/TRPP/project (br1)
$ nano smth

Den@PCLite MINGW64 ~/TRPP/project (br1)
$ git checkout br2
Switched to branch 'br2'

Den@PCLite MINGW64 ~/TRPP/project (br2)
$ nano smth

Den@PCLite MINGW64 ~/TRPP/project (br2)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Den@PCLite MINGW64 ~/TRPP/project (main)
$ nano smth

Den@PCLite MINGW64 ~/TRPP/project (main)
$

```

Рисунок 69 – Вывод содержимого файла smth в различных ветвях

```

Den@PCLite MINGW64 ~/TRPP/project (main)
$ git merge br1
bash: git: command not found

Den@PCLite MINGW64 ~/TRPP/project (main)
$ git merge br1
Updating c87c813..6c4355c
Fast-forward
 smth | 3 +++
 1 file changed, 3 insertions(+)

Den@PCLite MINGW64 ~/TRPP/project (main)
$ git merge br2
Auto-merging smth
CONFLICT (content): Merge conflict in smth
Automatic merge failed; fix conflicts and then commit the result.

Den@PCLite MINGW64 ~/TRPP/project (main|MERGING)
$ git checkout --theirs
fatal: '--ours/--theirs' needs the paths to check out

Den@PCLite MINGW64 ~/TRPP/project (main|MERGING)
$ git checkout --theirs smth
Updated 1 path from the index

Den@PCLite MINGW64 ~/TRPP/project (main|MERGING)
$ git add smth

Den@PCLite MINGW64 ~/TRPP/project (main|MERGING)
$ git commit -m "added smth"
[main 4d5b92b] added smth

Den@PCLite MINGW64 ~/TRPP/project (main)
$

```

Рисунок 70 – Слияние веток br1 и br2 с main

Часть 5. Индивидуальное задание №2

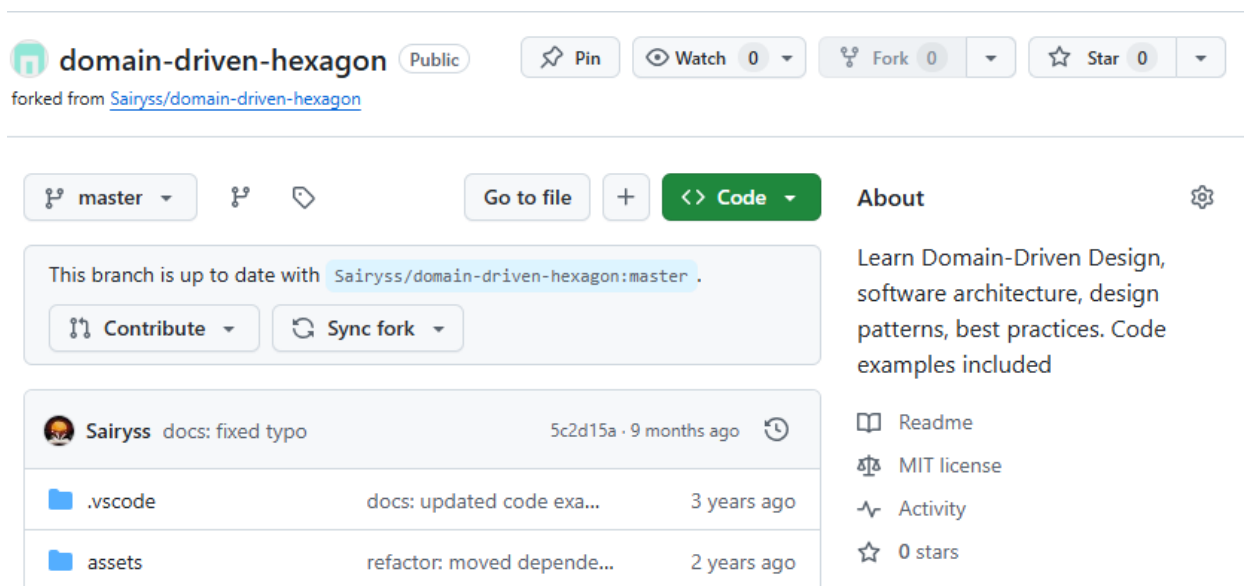


Рисунок 71 – Форк репозитория согласно варианту

```

Den@PCLite MINGW64 ~/TRPP (main)
$ git clone https://github.com/Klarnold/domain-driven-hexagon.git
Cloning into 'domain-driven-hexagon'...
remote: Enumerating objects: 3360, done.
remote: Counting objects: 100% (592/592), done.
remote: Compressing objects: 100% (109/109), done.
remote: Total 3360 (delta 520), reused 483 (delta 483), pack-reused 2768 (from 2)
Receiving objects: 100% (3360/3360), 7.61 MiB | 4.88 MiB/s, done.
Resolving deltas: 100% (1949/1949), done.

Den@PCLite MINGW64 ~/TRPP (main)
$ |

```

Рисунок 72 – Клонирование форка на локальную машину

```

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (master)
$ git branch branch1

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (master)
$ git branch branch2

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (master)
$ git branch -l
  branch1
  branch2
* master

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (master)
$

```

Рисунок 73 – Создание веток branch1 и branch2

```

GNU nano 8.2                                README.md
# Domain-Driven Hexagon

**Check out my other repositories**:

- [Backend best practices](https://github.com/Sairyss/backend-
- [System Design Patterns](https://github.com/Sairyss/system-d
- [Full Stack starter template](https://github.com/Sairyss/fu

```

Рисунок 74 – Изначальный вид файла README.md

```

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1)
$ git commit -m "second"
[branch1 db336fd] second
1 file changed, 1 insertion(+), 1 deletion(-)

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1)
$ nano README.md

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1)
$ git add README.md

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1)
$ git commit -m "third"
[branch1 0588f61] third
1 file changed, 1 insertion(+), 1 deletion(-)

```

Рисунок 75 – Осуществление коммитов

```

MINGW64:/c/Users/Den/TRPP/domain-driven-hexagon
GNU nano 8.2 README.md
# Domain-Driven Hexagon
commit1; commit2; commit3;
**Check out my other repositories**:

```

Рисунок 76 – Результат 3 коммитов для файла README.md в ветке branch1

```

GNU nano 8.2 README.md
# Domain-Driven Hexagon
commit1; commit2; commit3; chages
**Check out my other repositories**:

```

Рисунок 77 – Результат 3 коммитов для файла README.md в ветке branch2

```

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1)
$ git merge branch2
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.

```

Рисунок 78 – Возникновение конфликта

```

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch2)
$ git checkout branch1
Switched to branch 'branch1'

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1)
$ git merge branch2
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1|MERGING)
$ git checkout --theirs README.md
Updated 1 path from the index

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1|MERGING)
$ git add README.md

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1|MERGING)
$ git commit -m "added theirs"
[branch1 d18758a] added theirs

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1)
$ nano README.md

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1)
$

```

Рисунок 79 – Решение конфликта слияния

```

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1)
$ git push origin branch1
Everything up-to-date

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1)
$

```

Рисунок 80 – Выгрузка изменений в ветке branch1 в удалённый репозиторий

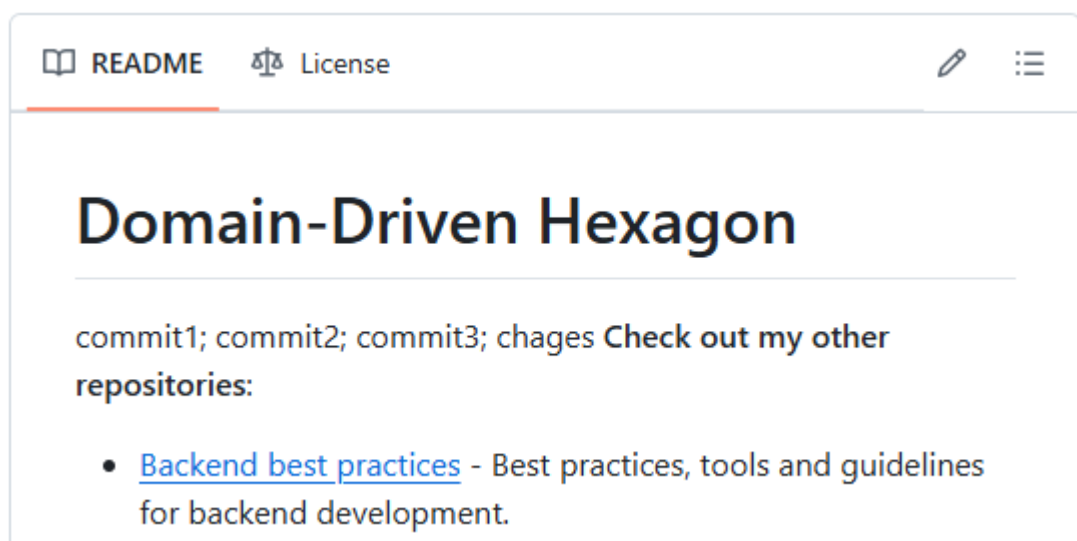


Рисунок 81 – Изменения в ветке branch2

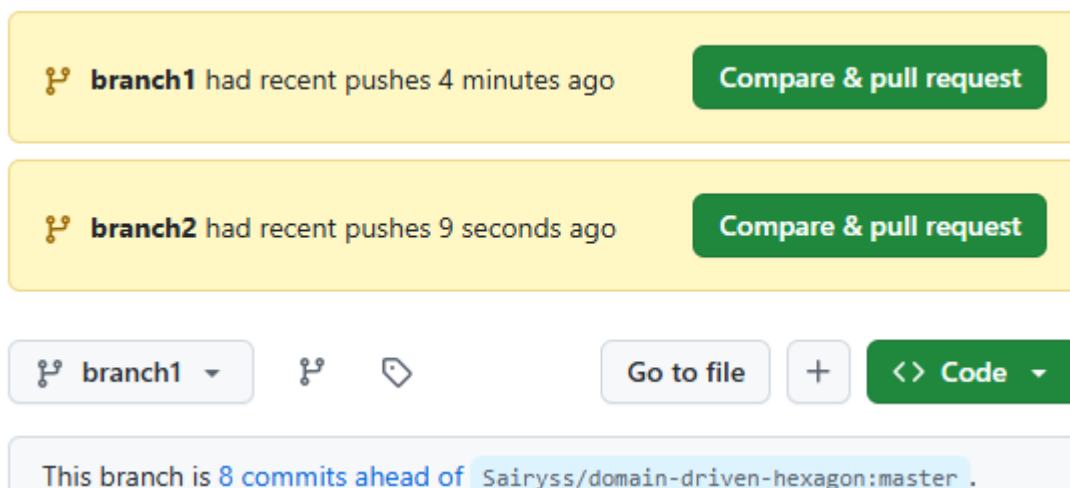


Рисунок 82 – Наличие веток branch1 и branch2 в удалённом репозитории

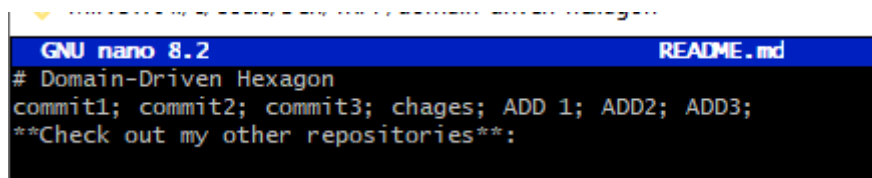


Рисунок 83 – Результат 3 коммитов для файла README.md в ветке branch1


```

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1)
$ nano README.md

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1)
$ git add README.md

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1)
$ git commit -m "new first"
[branch1 a31c9da] new first
1 file changed, 1 insertion(+), 1 deletion(-)

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1)
$ nano README.md

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1)
$ git add README.md

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1)
$ git commit -m "new second"
[branch1 acf3183] new second
1 file changed, 1 insertion(+), 1 deletion(-)

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1)
$ nano README.md

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1)
$ git add README.md

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1)
$ git commit -m "new third"
[branch1 98aad46] new third
1 file changed, 1 insertion(+), 1 deletion(-)

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1)
$ |

```

Рисунок 84 – Осуществление коммитов в branch1

```

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1)
$ cd ..

Den@PCLite MINGW64 ~/TRPP (main)
$ mkdir exp

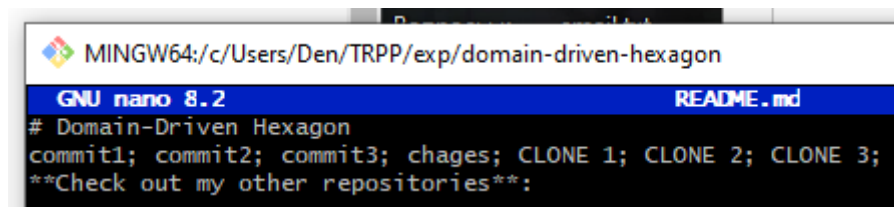
Den@PCLite MINGW64 ~/TRPP (main)
$ cd exp

Den@PCLite MINGW64 ~/TRPP/exp (main)
$ git clone https://github.com/Klarnold/domain-driven-hexagon.git
Cloning into 'domain-driven-hexagon'...
remote: Enumerating objects: 3376, done.
remote: Counting objects: 100% (608/608), done.
remote: Compressing objects: 100% (113/113), done.
remote: Total 3376 (delta 532), reused 499 (delta 495), pack-reused 2768 (from 2)
Receiving objects: 100% (3376/3376), 7.61 MiB | 5.00 MiB/s, done.
Resolving deltas: 100% (1961/1961), done.

Den@PCLite MINGW64 ~/TRPP/exp (main)
$ |

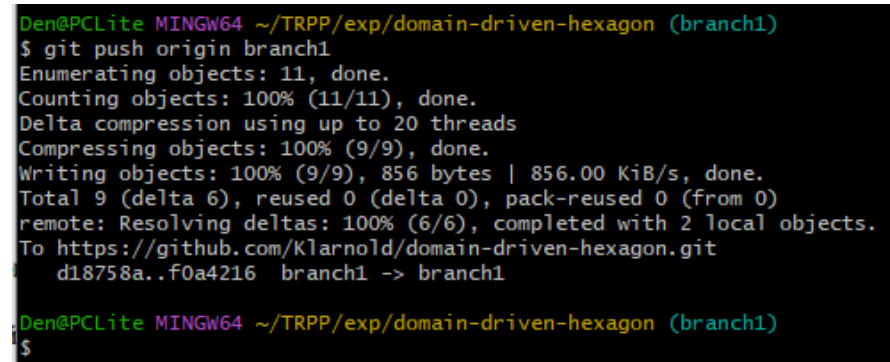
```

Рисунок 85 – Клонирование удалённого репозитория в новую папку



```
MINGW64:/c/Users/Den/TRPP/exp/domain-driven-hexagon
GNU nano 8.2 README.md
# Domain-Driven Hexagon
commit1; commit2; commit3; chages; CLONE 1; CLONE 2; CLONE 3;
**Check out my other repositories**:
```

Рисунок 86 – Результат коммитов в branch1 для файла README.md



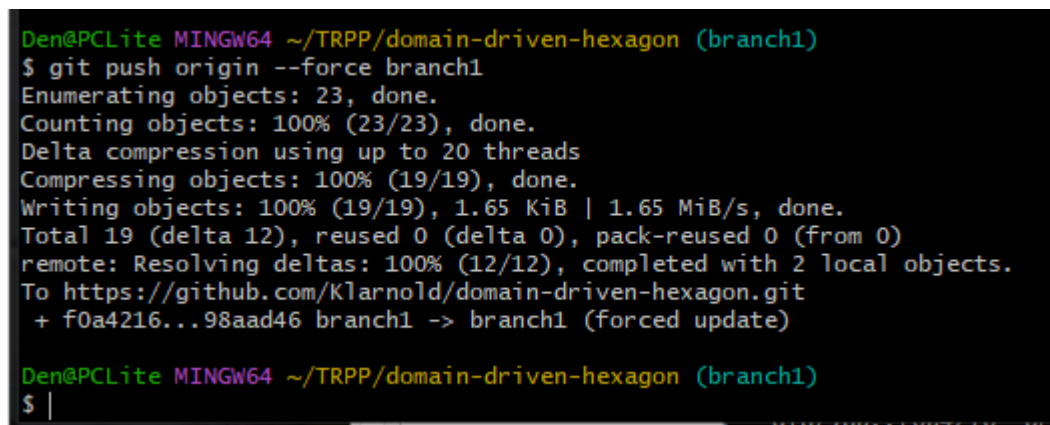
```
Den@PCLite MINGW64 ~/TRPP/exp/domain-driven-hexagon (branch1)
$ git push origin branch1
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 20 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 856 bytes | 856.00 KiB/s, done.
Total 9 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (6/6), completed with 2 local objects.
To https://github.com/Klarnold/domain-driven-hexagon.git
   d18758a..f0a4216  branch1 -> branch1

Den@PCLite MINGW64 ~/TRPP/exp/domain-driven-hexagon (branch1)
$
```

Рисунок 87 – Выгрузка изменений с нового клона в удалённый репозиторий



Рисунок 88 – Отображение изменений в удалённом репозитории



```
Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1)
$ git push origin --force branch1
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 20 threads
Compressing objects: 100% (19/19), done.
Writing objects: 100% (19/19), 1.65 KiB | 1.65 MiB/s, done.
Total 19 (delta 12), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (12/12), completed with 2 local objects.
To https://github.com/Klarnold/domain-driven-hexagon.git
 + f0a4216...98aad46 branch1 -> branch1 (forced update)

Den@PCLite MINGW64 ~/TRPP/domain-driven-hexagon (branch1)
$
```

Рисунок 89 – Обновление удалённого репозитория при помощи --force

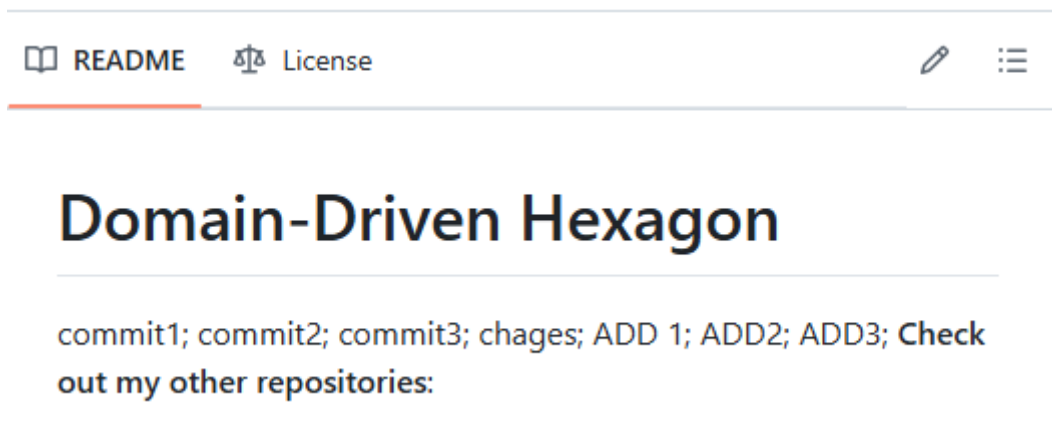


Рисунок 90 - Отображение изменений в удалённом репозитории

Часть 6. Ответы на вопросы

4. Что такое коммит?

Коммит – это фиксация изменений, произведённых в текущем репозитории.

5. Что такое ветка в репозитории Git?

Ветка в Git – это набор изменений, сохраняющий хронологический порядок. Можно сказать, что изначально репозиторий состоит из одной ветки. Ветки можно сливать друг с другом, а также переходить от коммита к коммита.

7. Что такое слияние двух веток?

Слияние двух веток – это перенос изменений с одной ветки на другую. Если мы говорим о слиянии branch1 с branch2, то в таком случае все изменения с branch1 перейдут в branch2.

8. Что такое конфликт в Git? Как его решить и почему они бывают?

Конфликт в Git – это когда при слиянии двух веток изменения одной ветки противоречат другой – то есть разные изменения в одинаковых файлах и система не может решить, какие изменения стоит записать. Решить это проблему можно при помощи выбора вручную нужней версии или при

устранении нежелаемого коммита в самом файле с удалением разметкой, обозначающей два разных коммита.

9. Как отменить слияние веток, если произошел конфликт?

Отменить слияние можно при помощи команды `git merge --abort`.

13. Что делает команда `git log`?

Команда `git log` выводит на экран коммиты текущей ветви в хронологическом порядке (новейшие коммиты находятся сверху) с отображением положения ветвей на определённых коммитах.

25. Как отменить действие команды "`git add`" на файл?

Отменить действие команды `git add` можно при помощи команд `git reset <наименование файла>` (`git reset` – отменяет действие команды для всех файлов).

32. Что такое GitHub?

GitHub - веб-сервис для хостинга IT-проектов и их совместной разработки. По сути, это визуальный интерфейс для работы с системой контроля версий Git. Также GitHub предоставляет глобальное взаимодействие между репозиториями, позволяя пользователям комментировать их и обращаться к ним.

Вывод

В ходе я ознакомился с такой прекрасным инструментом как Git, который позволяет вести контроль и сохранение старых версий репозитория, что значительно облегчает работу программиста. Кроме ознакомления, я на практике закрепил полученные знания и обрёл навыки, которые непременно пригодятся мне в дальнейшей карьере программиста.