



МИНОБРНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное учреждение
высшего образования*

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Отчет по выполнению практической работы №1

Тема:

СИСТЕМЫ КОНТРОЛЯ ВЕРСИЙ

Дисциплина: «Технология разработки программных приложений»

Выполнил студент: Враженко Д.О.

Группа: ИКБО-50-23

Вариант: 6

Москва – 2025

ОГЛАВЛЕНИЕ

ЦЕЛЬ РАБОТЫ.....	4
ХОД РАБОТЫ.....	5
Часть 1. Основные команды Git.....	5
Шаг 1. Устанавливаем Git на локальную машину.....	5
Шаг 2. Настройка Git.....	5
Шаг 3. Начало работы – создание папки и файлов.....	6
Шаг 4. Создание репозитория.....	6
Шаг 5. Добавление файла в репозиторий и добавление первого коммита....	7
Шаг 6. Индексация изменений.....	8
Шаг 6*. Индексация и коммит (не обязательный) – для ознакомления.....	8
Шаг 7. Коммиты нескольких изменений.....	8
Шаг 8. Просмотр истории коммитов.....	10
Шаг 9. Получение старых версий.....	12
Шаг 10. Отмена локальных изменений (до индексации).....	12
Шаг 11. Отмена локальных изменений (после индексации и до коммита).	13
Шаг 12. Отмена коммита.....	14
Часть 2. Системы управления репозиториями.....	15
Личный вариант: 6.....	15
Шаг 1. Создание аккаунта на GitHub.....	15
Шаг 2. Создание SSH-ключа для авторизации.....	15
Шаг 3. Создание нового репозитория для своего проекта.....	16
Шаг 4. Связываем локальный и удаленный репозитории.....	16
Шаг 5. Создание веток и переключение между ними.....	16
Шаг 6. Слияние веток.....	16
Часть 3. Работа с ветвлением и оформление кода.....	17
Личный вариант: 6.....	17
Шаг 1. Форк репозитория.....	17

Шаг 2. Клонирование репозитория на локальную машину.....	17
Шаг 3. Создание двух веток в master'e.....	17
Шаг 4. 3 коммита в каждую из веток, которые меняют один и тот же.....	18
кусочек файла.....	18
Шаг 5. Слияние ветки branch1 в ветку branch2 с разрешением.....	18
конфликтов.....	18
Шаг 6. Выгрузка всех изменений во всех ветках в удаленный.....	18
репозиторий.....	18
Шаг 7. Выполнение 3-х коммитов в branch1.....	18
Шаг 8. Повторное клонирование репозитория в другую директорию.....	19
Шаг 9. Выполнение 3-х коммитов в ветку branch1.....	19
Шаг 10. Выгрузка всех изменений в удаленный репозиторий.....	19
Шаг 11. Выгрузка изменения из старого репозитория с опцией —force.....	19
Шаг 12. Получение всех изменения в новом репозитории.....	20
КОНТРОЛЬНЫЕ ВОПРОСЫ.....	21
ВЫВОД.....	22

ЦЕЛЬ РАБОТЫ

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

Часть 1. Основные команды Git

Часть 2. Системы управления репозиториями

Часть 3. Работа с ветвлением и оформление кода

ХОД РАБОТЫ

Часть 1. Основные команды Git

Шаг 1. Устанавливаем Git на локальную машину

```
danil@danil-VirtualBox:~$ sudo apt install git
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
  git-man liberror-perl
Предлагаемые пакеты:
  git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
Следующие НОВЫЕ пакеты будут установлены:
  git git-man liberror-perl
Обновлено 0 пакетов, установлено 3 новых пакетов, для удаления отмечено 0 пакетов, и 171 пакетов не обновлено.
Необходимо скачать 4 120 кВ/4 147 кВ архивов.
После данной операции объем занятого дискового пространства возрастет на 21,0 МВ.
Хотите продолжить? [Д/н] Д
Пол:1 http://ru.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git-man all 1:2.34.1-1ubuntu1.12 [955 kB]
Пол:2 http://ru.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git amd64 1:2.34.1-1ubuntu1.12 [3 165 kB]
Получено 4 120 кВ за 25с (162 kB/s)
Выбор ранее не выбранного пакета liberror-perl.
(Чтение базы данных ... на данный момент установлен 206101 файл и каталог.)
Подготовка к распаковке .../liberror-perl_0.17029-1_all.deb ...
Распаковывается liberror-perl (0.17029-1) ...
Выбор ранее не выбранного пакета git-man.
Подготовка к распаковке .../git-man_1%3a2.34.1-1ubuntu1.12_all.deb ...
Распаковывается git-man (1:2.34.1-1ubuntu1.12) ...
Выбор ранее не выбранного пакета git.
Подготовка к распаковке .../git_1%3a2.34.1-1ubuntu1.12_amd64.deb ...
Распаковывается git (1:2.34.1-1ubuntu1.12) ...
Настраивается пакет liberror-perl (0.17029-1) ...
Настраивается пакет git-man (1:2.34.1-1ubuntu1.12) ...
Настраивается пакет git (1:2.34.1-1ubuntu1.12) ...
Обрабатываются триггеры для man-db (2.10.2-1) ...
danil@danil-VirtualBox:~$ git --version
git version 2.34.1
```

Рисунок 1 - Установка в Linux и Unix

Шаг 2. Настройка Git

```
danil@danil-VirtualBox:~$ git config --global user.name "DrTECHNIC"
danil@danil-VirtualBox:~$ git config --global user.email "dr.technic27111985@gmail.com"
```

Рисунок 2 - Логин и почта

```
danil@danil-VirtualBox:~$ git config --global core.autocrlf input
danil@danil-VirtualBox:~$ git config --global core.safecrlf warn
```

Рисунок 3 - Команды для Unix/Mac

```
danil@danil-VirtualBox:~$ git config --global core.quotepath off
danil@danil-VirtualBox:~$ git config --list
user.name=DrTECHNIC
user.email=dr.technic27111985@gmail.com
core.autocrlf=input
core.safecrlf=warn
core.quotepath=off
```

Рисунок 4 - Список результатов

Шаг 3. Начало работы – создание папки и файлов

```
danil@danil-VirtualBox:~$ ls
snap  vrazhenko  Видео  Документы  Загрузки  Изображения  Музыка  Общедоступные  'Рабочий стол'  Шаблоны
danil@danil-VirtualBox:~$ mkdir TRPP
danil@danil-VirtualBox:~$ ls
snap  TRPP  vrazhenko  Видео  Документы  Загрузки  Изображения  Музыка  Общедоступные  'Рабочий стол'  Шаблоны
danil@danil-VirtualBox:~$ cd TRPP/
danil@danil-VirtualBox:~/TRPP$ touch project.html
danil@danil-VirtualBox:~/TRPP$ ls
project.html
danil@danil-VirtualBox:~/TRPP$ nano project.html
```

Рисунок 5 - Создание папки для репозитория и файла

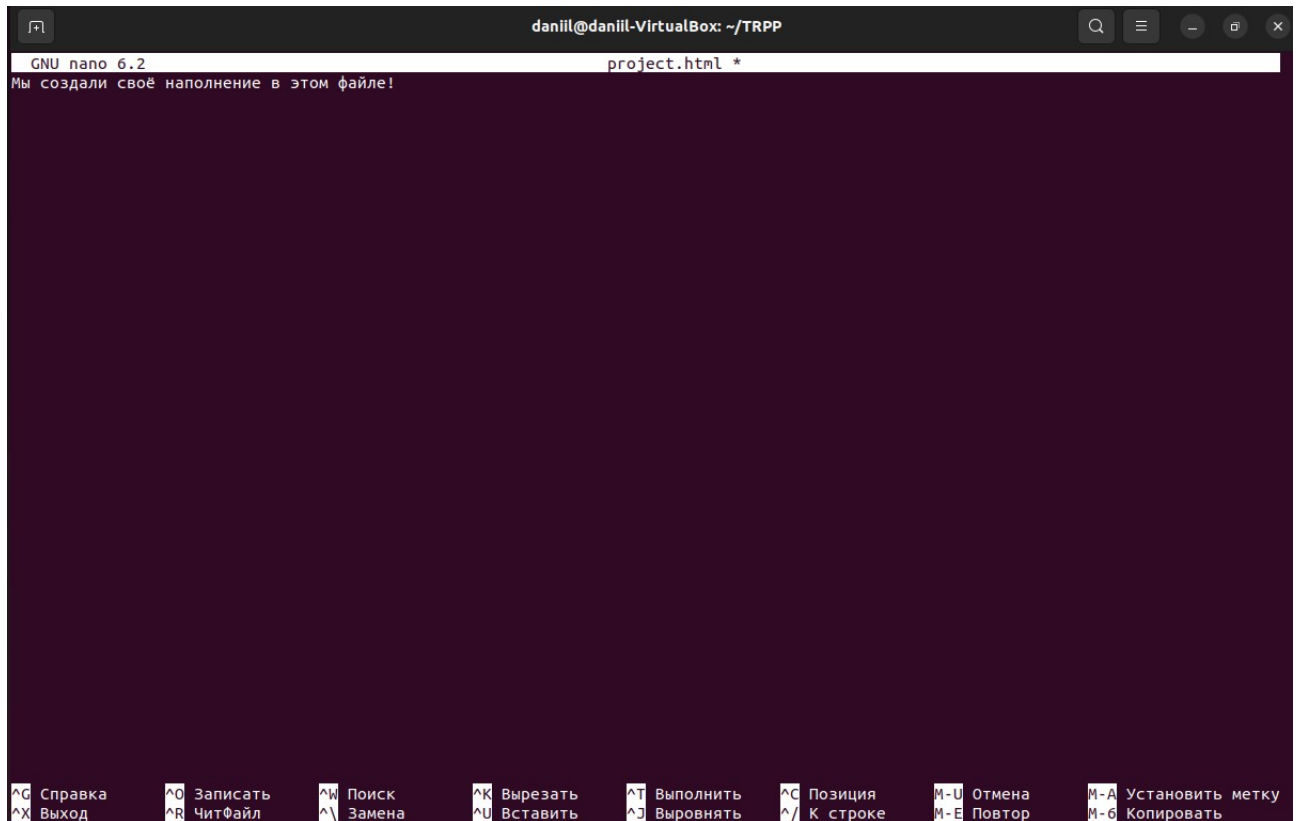


Рисунок 6 - Наполнение файла

Шаг 4. Создание репозитория

```
danil@danil-VirtualBox:~/TRPP$ git init
подсказка: Using 'master' as the name for the initial branch. This default branch name
подсказка: is subject to change. To configure the initial branch name to use in all
подсказка: of your new repositories, which will suppress this warning, call:
подсказка:
подсказка:     git config --global init.defaultBranch <name>
подсказка:
подсказка: Names commonly chosen instead of 'master' are 'main', 'trunk' and
подсказка: 'development'. The just-created branch can be renamed via this command:
подсказка:
подсказка:     git branch -m <name>
Инициализирован пустой репозиторий Git в /home/danil/TRPP/.git/
```

Рисунок 7 - Инициализация репозитория

Шаг 5. Добавление файла в репозиторий и добавление первого коммита

```
danil@danil-VirtualBox:~/TRPP$ git add project.html
danil@danil-VirtualBox:~/TRPP$ git commit -m "Добавление первого коммита"
[master (корневой коммит) b61e48f] Добавление первого коммита
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 project.html
danil@danil-VirtualBox:~/TRPP$ git status
Текущая ветка: master
нечего коммитить, нет изменений в рабочем каталоге
danil@danil-VirtualBox:~/TRPP$ nano project.html
```

Рисунок 8 - Создание первого коммита

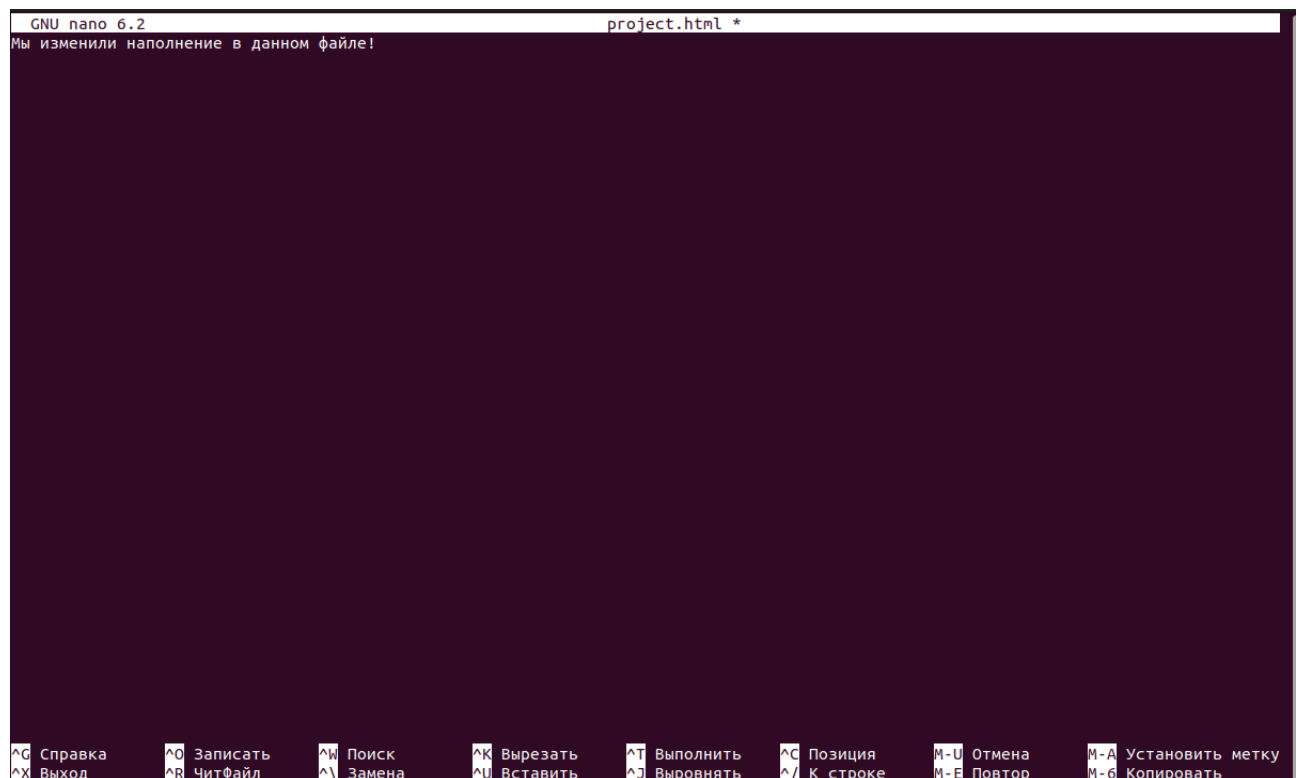


Рисунок 9 - Изменённое наполнение файла

```
danil@danil-VirtualBox:~/TRPP$ git status
Текущая ветка: master
Изменения, которые не в индексе для коммита:
  (используйте «git add <файл>...», чтобы добавить файл в индекс)
  (используйте «git restore <файл>...», чтобы отменить изменения в рабочем каталоге)
    изменено:      project.html
индекс пуст (используйте «git add» и/или «git commit -a»)
```

Рисунок 10 - Результат git status

Шаг 6. Индексация изменений

```
daniil@daniil-VirtualBox:~/TRPP$ git add project.html
daniil@daniil-VirtualBox:~/TRPP$ git status
Текущая ветка: master
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    изменено:      project.html
```

Рисунок 11 - Индексация изменений

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

```
daniil@daniil-VirtualBox:~/TRPP$ touch a.html
daniil@daniil-VirtualBox:~/TRPP$ touch b.html
daniil@daniil-VirtualBox:~/TRPP$ touch c.html
daniil@daniil-VirtualBox:~/TRPP$ nano a.html
daniil@daniil-VirtualBox:~/TRPP$ nano b.html
daniil@daniil-VirtualBox:~/TRPP$ nano c.html
daniil@daniil-VirtualBox:~/TRPP$ git add a.html
daniil@daniil-VirtualBox:~/TRPP$ git add b.html
daniil@daniil-VirtualBox:~/TRPP$ git commit -m "Changes for a and b"
[master 0885a09] Changes for a and b
 3 files changed, 3 insertions(+)
 create mode 100644 a.html
 create mode 100644 b.html
daniil@daniil-VirtualBox:~/TRPP$ git add c.html
daniil@daniil-VirtualBox:~/TRPP$ git commit -m "Underlated change to c"
[master e97ab50] Underlated change to c
 1 file changed, 1 insertion(+)
 create mode 100644 c.html
```

Рисунок 12 - Необязательный пункт

Шаг 7. Коммиты нескольких изменений

```
GNU nano 6.2
<html>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

Рисунок 13 - Изменения файла project.html


```

GNU nano 6.2
<html>
  <head>
</head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>

```

Рисунок 14 - Повторные изменения в project.html

```

daniil@daniil-VirtualBox:~/TRPP$ nano project.html
daniil@daniil-VirtualBox:~/TRPP$ git add project.html
daniil@daniil-VirtualBox:~/TRPP$ nano project.html
daniil@daniil-VirtualBox:~/TRPP$ git status
Текущая ветка: master
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    изменено:      project.html

Изменения, которые не в индексе для коммита:
  (используйте «git add <файл>...», чтобы добавить файл в индекс)
  (используйте «git restore <файл>...», чтобы отменить изменения в рабочем каталоге)
    изменено:      project.html

daniil@daniil-VirtualBox:~/TRPP$ git commit -m "Added standard HTML page tags"
[master 045b83c] Added standard HTML page tags
 1 file changed, 5 insertions(+), 1 deletion(-)
daniil@daniil-VirtualBox:~/TRPP$ git status
Текущая ветка: master
Изменения, которые не в индексе для коммита:
  (используйте «git add <файл>...», чтобы добавить файл в индекс)
  (используйте «git restore <файл>...», чтобы отменить изменения в рабочем каталоге)
    изменено:      project.html

индекс пуст (используйте «git add» и/или «git commit -a»)

```

Рисунок 15 - Результат добавления коммита

```

daniil@daniil-VirtualBox:~/TRPP$ git add .
daniil@daniil-VirtualBox:~/TRPP$ git status
Текущая ветка: master
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    изменено:      project.html

daniil@daniil-VirtualBox:~/TRPP$ git commit -m "Added HTML header"
[master b1cf5e2] Added HTML header
 1 file changed, 2 insertions(+)

```

Рисунок 16 - Коммит второго изменения

Шаг 8. Просмотр истории коммитов

```
danil@danil-VirtualBox:~/TRPP$ git log
commit b1cf5e23b9a230d31a30631c141f377a2639fc53 (HEAD -> master)
Author: DrTECHNIC <dr.technic27111985@gmail.com>
Date: Sat Feb 15 13:48:02 2025 +0300

    Added HTML header

commit 045b83c5735c9c9370efff6b26f99b41a4f1c8fe
Author: DrTECHNIC <dr.technic27111985@gmail.com>
Date: Sat Feb 15 13:46:03 2025 +0300

    Added standard HTML page tags

commit e97ab50821833375e9329215b419fbb7241c7a7f
Author: DrTECHNIC <dr.technic27111985@gmail.com>
Date: Sat Feb 15 13:37:48 2025 +0300

    Underlated change to c

commit 0885a095eb17cdf38ed4d1c2de54d0e254cbc37c
Author: DrTECHNIC <dr.technic27111985@gmail.com>
Date: Sat Feb 15 13:37:02 2025 +0300

    Changes for a and b

commit b61e48f0a0af916da51051636434a526b743ed92
Author: DrTECHNIC <dr.technic27111985@gmail.com>
Date: Sat Feb 15 13:26:25 2025 +0300

    Добавление первого коммита
```

Рисунок 17 - История коммитов

```
danil@danil-VirtualBox:~/TRPP$ git log --pretty=oneline
b1cf5e23b9a230d31a30631c141f377a2639fc53 (HEAD -> master) Added HTML header
045b83c5735c9c9370efff6b26f99b41a4f1c8fe Added standard HTML page tags
e97ab50821833375e9329215b419fbb7241c7a7f Underlated change to c
0885a095eb17cdf38ed4d1c2de54d0e254cbc37c Changes for a and b
b61e48f0a0af916da51051636434a526b743ed92 Добавление первого коммита
```

Рисунок 18 - История коммитов в одну строку

```

daniil@daniil-VirtualBox:~/TRPP$ git log --pretty=oneline --max-count=2
b1cf5e23b9a230d31a30631c141f377a2639fc53 (HEAD -> master) Added HTML header
045b83c5735c9c9370efff6b26f99b41a4f1c8fe Added standard HTML page tags
daniil@daniil-VirtualBox:~/TRPP$ git log --pretty=oneline --since='5 minutes ago'
b1cf5e23b9a230d31a30631c141f377a2639fc53 (HEAD -> master) Added HTML header
daniil@daniil-VirtualBox:~/TRPP$ git log --pretty=oneline --until='5 minutes ago'
045b83c5735c9c9370efff6b26f99b41a4f1c8fe Added standard HTML page tags
e97ab50821833375e9329215b419fbb7241c7a7f Underlated change to c
0885a095eb17cdf38ed4d1c2de54d0e254cbc37c Changes for a and b
b61e48f0a0af916da51051636434a526b743ed92 Добавление первого коммита
daniil@daniil-VirtualBox:~/TRPP$ git log --pretty=oneline --author=DrTECHNIC
b1cf5e23b9a230d31a30631c141f377a2639fc53 (HEAD -> master) Added HTML header
045b83c5735c9c9370efff6b26f99b41a4f1c8fe Added standard HTML page tags
e97ab50821833375e9329215b419fbb7241c7a7f Underlated change to c
0885a095eb17cdf38ed4d1c2de54d0e254cbc37c Changes for a and b
b61e48f0a0af916da51051636434a526b743ed92 Добавление первого коммита
daniil@daniil-VirtualBox:~/TRPP$ git log --pretty=oneline --all
b1cf5e23b9a230d31a30631c141f377a2639fc53 (HEAD -> master) Added HTML header
045b83c5735c9c9370efff6b26f99b41a4f1c8fe Added standard HTML page tags
e97ab50821833375e9329215b419fbb7241c7a7f Underlated change to c
0885a095eb17cdf38ed4d1c2de54d0e254cbc37c Changes for a and b
b61e48f0a0af916da51051636434a526b743ed92 Добавление первого коммита

```

Рисунок 19 - История коммитов с разными параметрами

```

daniil@daniil-VirtualBox:~/TRPP$ git log --pretty=format:"%h %ad | %s%d [%an]" --graph --date=short
* b1cf5e2 2025-02-15 | Added HTML header (HEAD -> master) [DrTECHNIC]
* 045b83c 2025-02-15 | Added standard HTML page tags [DrTECHNIC]
* e97ab50 2025-02-15 | Underlated change to c [DrTECHNIC]
* 0885a09 2025-02-15 | Changes for a and b [DrTECHNIC]
* b61e48f 2025-02-15 | Добавление первого коммита [DrTECHNIC]

```

Рисунок 20 - История коммитов с укороченным хэшем коммита, дополнением коммита, датой коммита, комментарием, именем автора, отображением дерева коммитов в виде ASCII-графика и сохранением формата даты коротким

Шаг 9. Получение старых версий

```
danil@danil-VirtualBox:~/TRPP$ git log --pretty=format:"%h %ad | %s%d [%an]" --graph --date=short
* b1cf5e2 2025-02-15 | Added HTML header (HEAD -> master) [DrTECHNIC]
* 045b83c 2025-02-15 | Added standard HTML page tags [DrTECHNIC]
* e97ab50 2025-02-15 | Underlated change to c [DrTECHNIC]
* 0885a09 2025-02-15 | Changes for a and b [DrTECHNIC]
* b61e48f 2025-02-15 | Добавление первого коммита [DrTECHNIC]
danil@danil-VirtualBox:~/TRPP$ git checkout b61e48f
Примечание: переключение на «b61e48f».
```

Вы сейчас в состоянии «отсоединённого указателя HEAD». Можете осмотреться, внести экспериментальные изменения и зафиксировать их, также можете отменить любые коммиты, созданные в этом состоянии, не затрагивая другие ветки, переключившись обратно на любую ветку.

Если хотите создать новую ветку для сохранения созданных коммитов, можете сделать это (сейчас или позже), используя команду `switch` с параметром `-c`. Например:

```
git switch -c <новая-ветка>
```

Или отмените эту операцию с помощью:

```
git switch -
```

Отключите этот совет, установив переменную конфигурации `advice.detachedHead` в значение `false`

HEAD сейчас на b61e48f Добавление первого коммита

Рисунок 21 - Получаем старую версию

Так как я случайно сохранил пустой файл в первый коммит, то у меня ничего не вывелось.

```
danil@danil-VirtualBox:~/TRPP$ cat project.html
danil@danil-VirtualBox:~/TRPP$ git checkout master
Предыдущая позиция HEAD была b61e48f Добавление первого коммита
Переключились на ветку «master»
danil@danil-VirtualBox:~/TRPP$ cat project.html
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

Рисунок 22 - Сравнение вывода в разных версиях

Шаг 10. Отмена локальных изменений (до индексации)

```
danil@danil-VirtualBox:~/TRPP$ git checkout master
Уже на «master»
danil@danil-VirtualBox:~/TRPP$ nano project.html
```

Рисунок 23 - Уже на ветке "master"


```

GNU nano 6.2
<html>
<head>
</head>
<body>
<h1>Мы создали своё наполнение в данном файле!</h1>
<!-- Это плохой комментарий. Мы должны убрать его -->
</body>
</html>

```

Рисунок 24 - Изменения в project.html

```

daniil@daniil-VirtualBox:~/TRPP$ git status
Текущая ветка: master
Изменения, которые не в индексе для коммита:
  (используйте «git add <файл>...», чтобы добавить файл в индекс)
  (используйте «git restore <файл>...», чтобы отменить изменения в рабочем каталоге)
    изменено:      project.html

индекс пуст (используйте «git add» и/или «git commit -a»)
daniil@daniil-VirtualBox:~/TRPP$ git checkout project.html
Updated 1 path from the index
daniil@daniil-VirtualBox:~/TRPP$ git status
Текущая ветка: master
ничего коммитить, нет изменений в рабочем каталоге
daniil@daniil-VirtualBox:~/TRPP$ cat project.html
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>

```

Рисунок 25 - Отмена изменений

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

```

GNU nano 6.2
<html>
<head>
<!-- Это нежелательный комментарий! -->
</head>
<body>
<h1>Мы создали своё наполнение в данном файле!</h1>
</body>
</html>

```

Рисунок 26 - Изменений в project.html

```

daniil@daniil-VirtualBox:~/TRPP$ nano project.html
daniil@daniil-VirtualBox:~/TRPP$ git add project.html
daniil@daniil-VirtualBox:~/TRPP$ git status
Текущая ветка: master
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    изменено:      project.html

daniil@daniil-VirtualBox:~/TRPP$ git reset HEAD project.html
Непроиндексированные изменения после сброса:
M       project.html
daniil@daniil-VirtualBox:~/TRPP$ git checkout project.html
Updated 1 path from the index
daniil@daniil-VirtualBox:~/TRPP$ git status
Текущая ветка: master
ничего коммитить, нет изменений в рабочем каталоге

```

Рисунок 27 - Отмена локальных изменений

Шаг 12. Отмена коммита

```

GNU nano 6.2
<html>
<head>
<!-- Это нежелательный комментарий! -->
</head>
<body>
<h1>Мы создали своё наполнение в данном файле!</h1>
</body>
</html>

```

Рисунок 28 - Содержимое файла project.html

```

daniil@daniil-VirtualBox:~/TRPP$ nano project.html
daniil@daniil-VirtualBox:~/TRPP$ git add project.html
daniil@daniil-VirtualBox:~/TRPP$ git commit -m "Ой, нам не нужен данный коммит"
[master d43d522] Ой, нам не нужен данный коммит
 1 file changed, 6 insertions(+), 5 deletions(-)
daniil@daniil-VirtualBox:~/TRPP$ git config --global core.editor nano
daniil@daniil-VirtualBox:~/TRPP$ git revert HEAD
[master 06df141] Revert "Ой, нам не нужен данный коммит"
 1 file changed, 5 insertions(+), 6 deletions(-)
daniil@daniil-VirtualBox:~/TRPP$ git revert HEAD --no-edit
[master e19ba42] Revert "Revert "Ой, нам не нужен данный коммит""
 Date: Sat Feb 15 14:31:01 2025 +0300
 1 file changed, 6 insertions(+), 5 deletions(-)
daniil@daniil-VirtualBox:~/TRPP$ git log --pretty=format:"%h %ad | %s%d [%an]" --graph --date=short
* e19ba42 2025-02-15 | Revert "Revert "Ой, нам не нужен данный коммит"" (HEAD -> master) [DrTECHNIC]
* 06df141 2025-02-15 | Revert "Ой, нам не нужен данный коммит" [DrTECHNIC]
* d43d522 2025-02-15 | Ой, нам не нужен данный коммит [DrTECHNIC]
* b1cf5e2 2025-02-15 | Added HTML header [DrTECHNIC]
* 045b83c 2025-02-15 | Added standard HTML page tags [DrTECHNIC]
* e97ab50 2025-02-15 | Underlated change to c [DrTECHNIC]
* 0885a09 2025-02-15 | Changes for a and b [DrTECHNIC]
* b61e48f 2025-02-15 | Добавление первого коммита [DrTECHNIC]

```

Рисунок 29 - Отмена коммита

Часть 2. Системы управления репозиториями

Личный вариант: 6

- 1) Клонировать непустой удаленный репозиторий на локальную машину
- 2) Создать новую ветку и вывести список всех веток
- 3) Произвести 3 коммита в новой ветке в новых файлах
- 4) Выгрузить изменения в удаленный репозиторий
- 5) Произвести какие-нибудь изменения в файле который существует в ветке master, но не коммитить их
- 6) Выполнить git stash
- 7) Переключиться на ветку master
- 8) Выполнить git stash pop
- 9) Вывести в консоли различия между веткой master и новой веткой
- 10) Слить новую ветку с master при помощи merge

Шаг 1. Создание аккаунта на GitHub

У меня уже был аккаунт на GitHub.

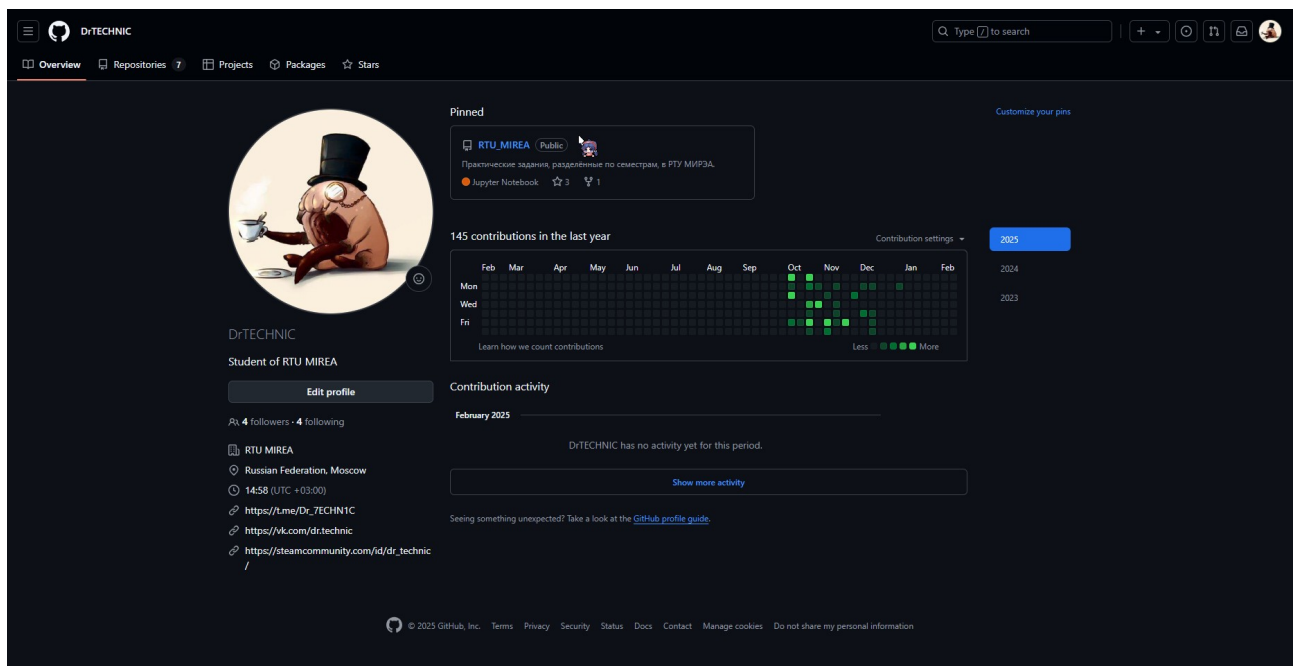


Рисунок 30 - Аккаунт на GitHub

Шаг 2. Создание SSH-ключа для авторизации

```
danil@danil-VirtualBox:~/TRPP$ cd ~/.ssh
danil@danil-VirtualBox:~/.ssh$ ssh-keygen -t rsa -b 4096 -C "dr.technic2711985@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/danil/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/danil/.ssh/id_rsa
Your public key has been saved in /home/danil/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:6l6j0aJSgE88PaKa0smJHZ2hjooV2FiFXZNzgTBAv8w dr.technic2711985@gmail.com
The key's randomart image is:
+---[RSA 4096]-----+
| .o++ooo..          |
| .o..+..           |
| o... o            |
| .=*o+.           |
| o+o*E+  S         |
| . .+.  o          |
| +*oo  + +         |
| O++  o = .        |
| *. ...+          |
+---[SHA256]-----+
danil@danil-VirtualBox:~/.ssh$ ssh-agent
SSH_AUTH_SOCK=/tmp/ssh-XXXXXXk3Ex0x/agent.5332; export SSH_AUTH_SOCK;
SSH_AGENT_PID=5333; export SSH_AGENT_PID;
echo Agent pid 5333;
```

Рисунок 31 - Создание SSH-ключа

```
danil@danil-VirtualBox:~/.ssh$ eval "$(ssh-agent -s)"
Agent pid 5342
danil@danil-VirtualBox:~/.ssh$ ssh-add ~/.ssh/id_rsa
Identity added: /home/danil/.ssh/id_rsa (dr.technic2711985@gmail.com)
danil@danil-VirtualBox:~/.ssh$ cat ~/.ssh/id_rsa
id_rsa      id_rsa.pub
danil@danil-VirtualBox:~/.ssh$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQD3SPsBm4Y5VwsA5tyBGDg3F0UfRTTrRQGDHeYYCnyvTmfWmPKga0Jcm9eHwEDTkKo4cB/kpEKXKiatqHBvu11pTWXJWtm
00e4TWddsnIE7htJEodwX6NAAeqEQ0NgInuILE4DcAd30SKX04iaL6PW4bY2Aq8X6dfcNRHtZmmjry96y8NLTuEz+gPCrJXheDd7oenQYmkmt2tJ1c082NCLBZQYr4DBfxzA
UGsihqqlikEsk16w/59jVnnRL7ltvLmvLe8AuIV1c5oRxB4BnHuFFSxc2vjdAGxh6ePtZKwpejYwtUd9UHRkqkm9wgKSY8BD22pLEqvJT3P2SI7x1ftLI5jldyhVftkXmn
lnENx2Rp45bmR1Hq/bid7YP4t4r6Y68Gn7Ad9fdeIubw2ksm/xz7qpZjJNKsvL26m4IRHj5V1hjB/jebe10afLt2Nwika/VA959BKx7abHPf9IZgIvwz4nE8u9iTNieBaRv1
JYdQ/oY6NLI8J9qApatnHLUL3Ge0sz4ykjZzu10yg80w08VFP6xY/QI0igypWn9r/tzv9JxxG1GaYCdVZ0g80oeYqTk6gUp5DLxM1sADG1zU30yeYhXiqI32CGok6fl1Rjsi
NqeioYSVgnCWpFu6p/gc2uMCndk+EmN+nGuB/7CUgMD0mro5fjCNFRAIfnN3Iw== dr.technic2711985@gmail.com
```

Рисунок 32 - Получение SSH-ключа

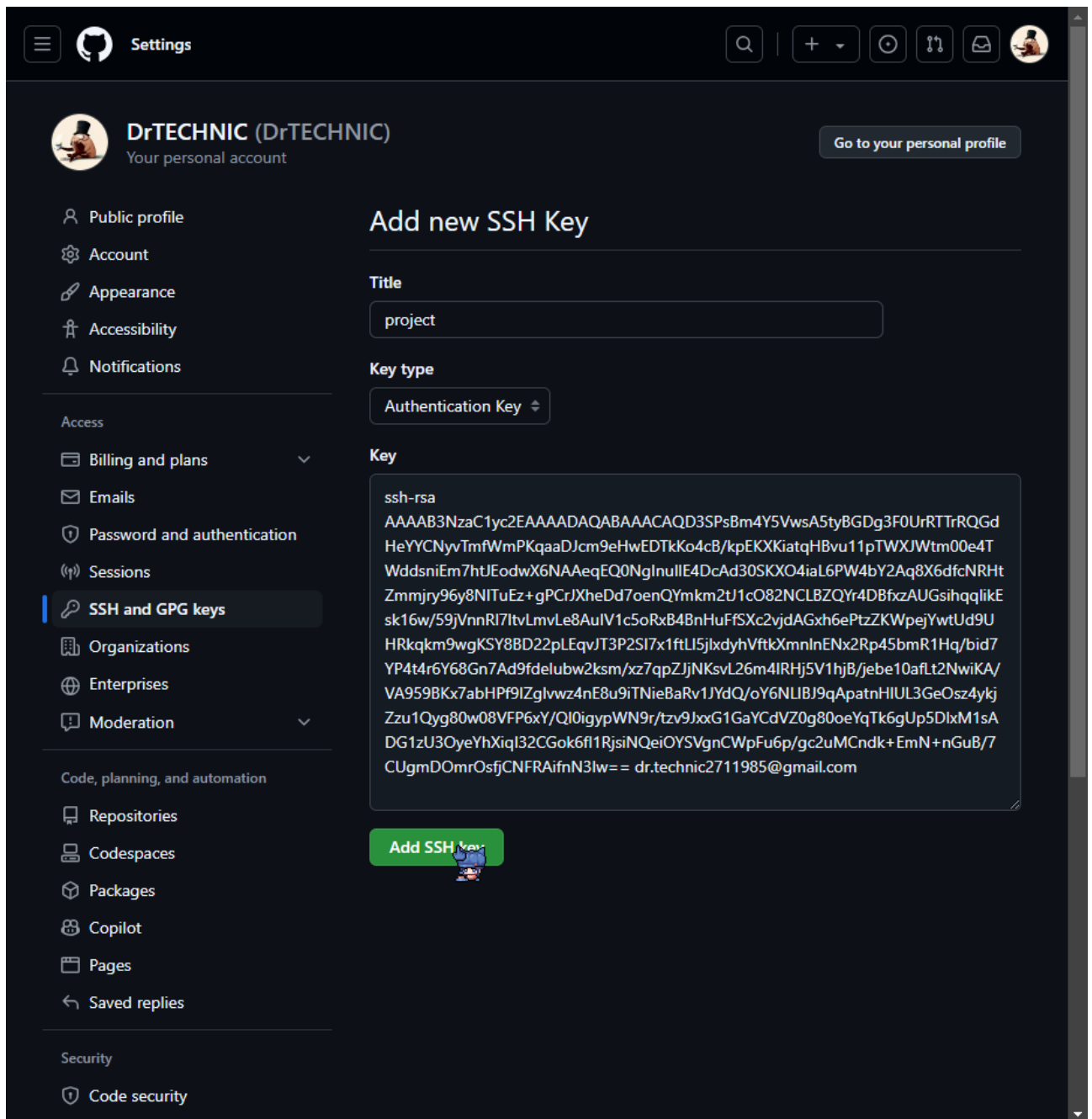


Рисунок 33 - Связь локального и удалённого репозитория

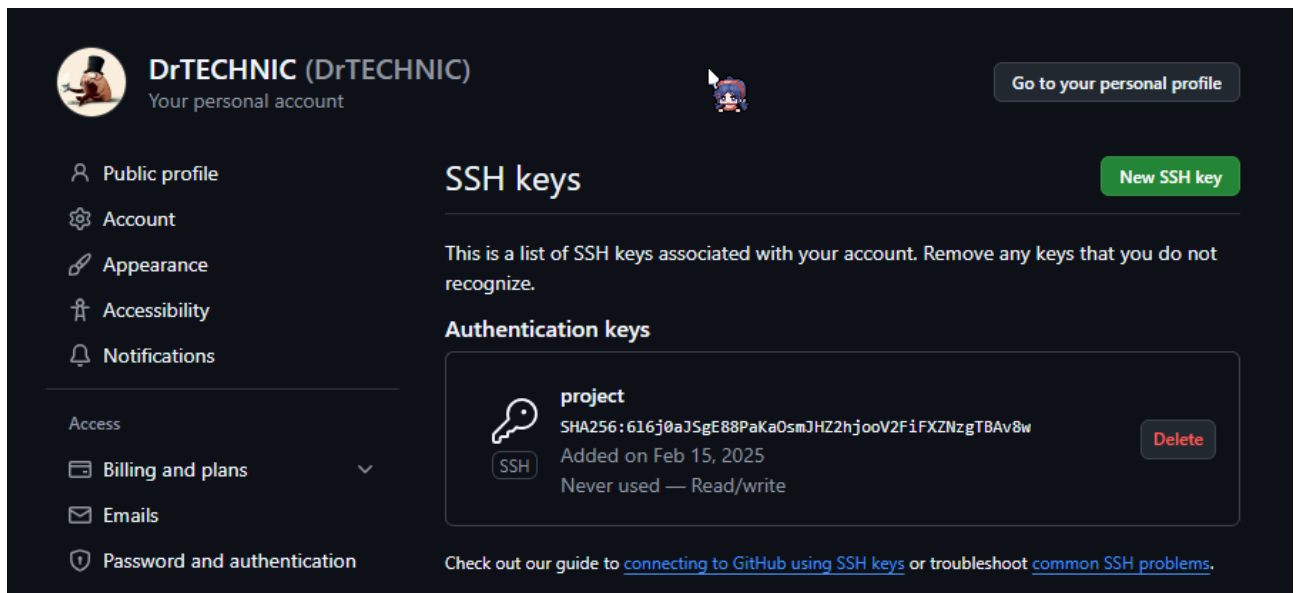


Рисунок 34 - SSH-ключ в GitHub

Шаг 3. Создание нового репозитория для своего проекта

```
danil@danil-VirtualBox:~$ mkdir project
danil@danil-VirtualBox:~$ ls
project  snap  TRPP  vrazhenko  Видео  Документы  Загрузки  Изображения  Музыка  Общедоступные  'Рабочий стол'  Шаблоны
danil@danil-VirtualBox:~$ cd project/
danil@danil-VirtualBox:~/project$ ls
danil@danil-VirtualBox:~/project$ nano readme.txt
danil@danil-VirtualBox:~/project$ nano project.html
danil@danil-VirtualBox:~/project$ git init
подсказка: Using 'master' as the name for the initial branch. This default branch name
подсказка: is subject to change. To configure the initial branch name to use in all
подсказка: of your new repositories, which will suppress this warning, call:
подсказка:
подсказка:     git config --global init.defaultBranch <name>
подсказка:
подсказка: Names commonly chosen instead of 'master' are 'main', 'trunk' and
подсказка: 'development'. The just-created branch can be renamed via this command:
подсказка:
подсказка:     git branch -m <name>
Инициализирован пустой репозиторий Git в /home/danil/project/.git/
```

Рисунок 35 - Новый репозиторий

Шаг 4. Связываем локальный и удаленный репозитории

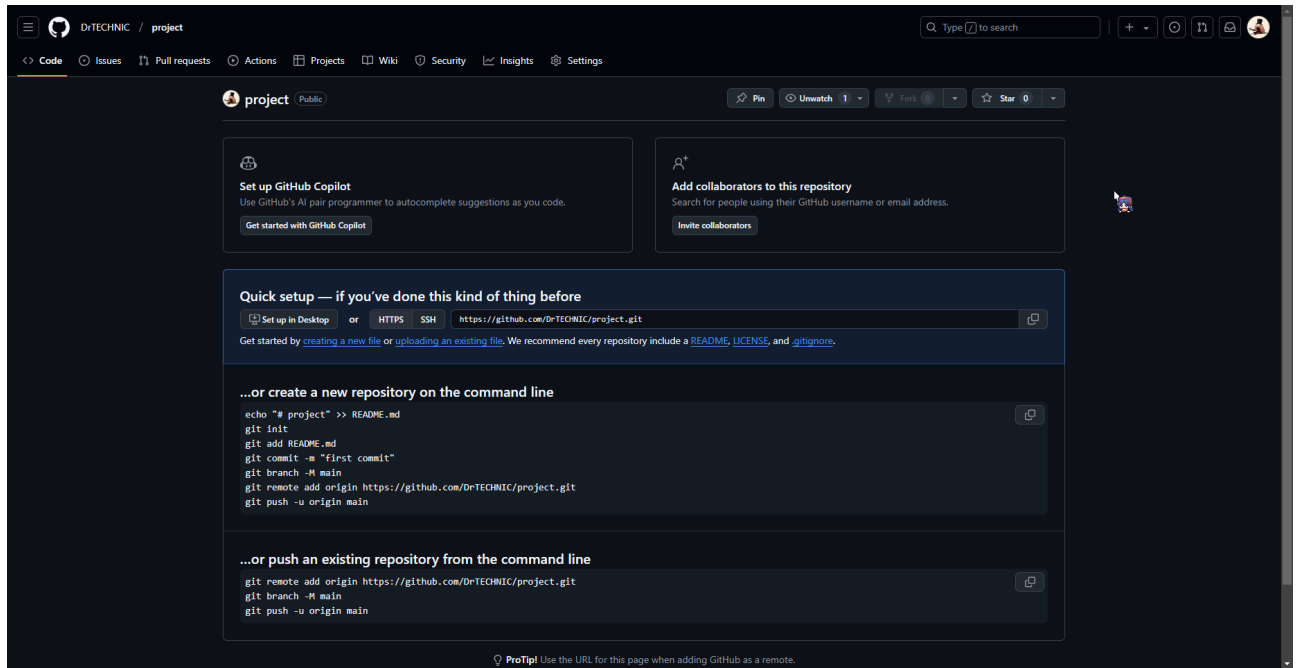


Рисунок 36 - Пустой репозиторий на GitHub

```
danil@danil-VirtualBox:~/project$ git remote add project git@github.com:DrTECHNIC/project.git
```

Рисунок 37 - Связь двух репозиторияев

Шаг 5. Создание веток и переключение между ними

```
danil@danil-VirtualBox:~/project$ git checkout -b main
Переключились на новую ветку «main»
danil@danil-VirtualBox:~/project$ git status
Текущая ветка: main

Еще нет коммитов

Неотслеживаемые файлы:
(используйте «git add <файл>...», чтобы добавить в то, что будет включено в коммит)
    project.html
    readme.txt

индекс пуст, но есть неотслеживаемые файлы
(используйте «git add», чтобы проиндексировать их)
danil@danil-VirtualBox:~/project$ git add project.html
danil@danil-VirtualBox:~/project$ git add readme.txt
danil@danil-VirtualBox:~/project$ git status
Текущая ветка: main

Еще нет коммитов

Изменения, которые будут включены в коммит:
(используйте «git rm --cached <файл>...», чтобы убрать из индекса)
    новый файл:    project.html
    новый файл:    readme.txt

danil@danil-VirtualBox:~/project$ git commit -m "first commit"
[main (корневой коммит) d15cd99] first commit
2 files changed, 7 insertions(+)
create mode 100644 project.html
create mode 100644 readme.txt
```

Рисунок 38 - Коммитирование изменений

Шаг 6. Слияние веток

Перед этим действием я создал пустой файл, который закоммитил в ветке main, а после слил из main в master.

```
danil@danil-VirtualBox:~/project$ git merge main
Обновление d15cd99..3905107
Fast-forward
 empty_file.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 empty_file.txt
```

Рисунок 39 - Слияние из main в master

Задание по варианту:

- 1) Клонировать непустой удаленный репозиторий на локальную машину

```
daniil@daniil-VirtualBox:~$ git clone https://github.com/DrTECHNIC/Repository_For_Tests
Клонирование в «Repository_For_Tests»...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Получение объектов: 100% (3/3), готово.
```

Рисунок 40 - Клонирование репозитория

- 2) Создайте новую ветку и выведите список всех веток

```
daniil@daniil-VirtualBox:~/Repository_For_Tests$ git checkout -b newbranch
Переключились на новую ветку «newbranch»
daniil@daniil-VirtualBox:~/Repository_For_Tests$ git branch -a
  main
* newbranch
remotes/origin/HEAD -> origin/main
remotes/origin/branch-1
remotes/origin/main
remotes/origin/secondBranch
```

Рисунок 41 - Названия всех веток

- 3) Произведите 3 коммита в новой ветке в новых файлах

```
daniil@daniil-VirtualBox:~/Repository_For_Tests$ touch text_file.txt
daniil@daniil-VirtualBox:~/Repository_For_Tests$ git add text_file.txt
daniil@daniil-VirtualBox:~/Repository_For_Tests$ git commit -m "first_commit"
[newbranch 9c4b8de] first_commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 text_file.txt
daniil@daniil-VirtualBox:~/Repository_For_Tests$ nano text_file.txt
daniil@daniil-VirtualBox:~/Repository_For_Tests$ git add text_file.txt
daniil@daniil-VirtualBox:~/Repository_For_Tests$ git commit -m "second_commit"
[newbranch e85e8f4] second_commit
1 file changed, 1 insertion(+)
daniil@daniil-VirtualBox:~/Repository_For_Tests$ nano text_file.txt
daniil@daniil-VirtualBox:~/Repository_For_Tests$ git add text_file.txt
daniil@daniil-VirtualBox:~/Repository_For_Tests$ git commit -m "third_commit"
[newbranch 638fb5b] third_commit
1 file changed, 1 insertion(+), 1 deletion(-)
```

Рисунок 42 - Создание 3 коммитов

4) Выгрузите изменения в удаленный репозиторий

```
danil@danil-VirtualBox:~/Repository_For_Tests$ git push -u origin newbranch
Username for 'https://github.com': ghp_cxEyqMivDYopJUMsqtq0htC33E6zX4AMA78
Password for 'https://ghp_cxEyqMivDYopJUMsqtq0htC33E6zX4AMA78@github.com':
Перечисление объектов: 10, готово.
Подсчет объектов: 100% (10/10), готово.
При сжатии изменений используется до 5 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (9/9), 815 байтов | 815.00 КиБ/с, готово.
Всего 9 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote:
remote: Create a pull request for 'newbranch' on GitHub by visiting:
remote:   https://github.com/DrTECHNIC/Repository_For_Tests/pull/new/newbranch
remote:
To https://github.com/DrTECHNIC/Repository_For_Tests
 * [new branch]   newbranch -> newbranch
Ветка «newbranch» отслеживает внешнюю ветку «newbranch» из «origin».
```

Рисунок 43 - Выгрузка в репозиторий

5) Произведите какие-нибудь изменения в файле который существует в ветке master, но не коммитьте их

Так как в этом репозитории нет ветки master, я буду использовать ветку main.

```
danil@danil-VirtualBox:~/Repository_For_Tests$ nano README.md
danil@danil-VirtualBox:~/Repository_For_Tests$ git status
Текущая ветка: main
Эта ветка соответствует «origin/main».

Изменения, которые не в индексе для коммита:
  (используйте «git add <файл>...», чтобы добавить файл в индекс)
  (используйте «git restore <файл>...», чтобы отменить изменения в рабочем каталоге)
        изменено:   README.md

индекс пуст (используйте «git add» и/или «git commit -a»)
```

Рисунок 44 - Незакомиченные изменения

```
GNU nano 6.2                                README.md *
# Repository_For_Tests
В этом репозитории я буду проводить различные тесты всего... ну, как я планирую...
----- Изменения -----
```

Рисунок 45 - Изменения в README.md

6) Выполните git stash

```
danil@danil-VirtualBox:~/Repository_For_Tests$ git stash
Рабочий каталог и состояние индекса сохранены WIP on main: a74254d Создание файла README.md
```

Рисунок 46 - Спрятали изменения

7) Переключитесь на ветку master

```
danil@danil-VirtualBox:~/Repository_For_Tests$ git checkout main
Уже на «main»
Эта ветка соответствует «origin/main».
```

Рисунок 47 - Переключение на main

8) Выполните git stash pop

```
danil@danil-VirtualBox:~/Repository_For_Tests$ git stash pop
Текущая ветка: main
Эта ветка соответствует «origin/main».

Изменения, которые не в индексе для коммита:
  (используйте «git add <файл>...», чтобы добавить файл в индекс)
  (используйте «git restore <файл>...», чтобы отменить изменения в рабочем каталоге)
      изменено:      README.md

индекс пуст (используйте «git add» и/или «git commit -a»)
Отброшено refs/stash@{0} (21c9180514fec4cc9dac2ad56c665dc0544239e8)
```

Рисунок 48 - Достаём спрятанные изменения

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

```
danil@danil-VirtualBox:~/Repository_For_Tests$ git diff main newbranch
diff --git a/text_file.txt b/text_file.txt
new file mode 100644
index 0000000..83f9fcf
--- /dev/null
+++ b/text_file.txt
@@ -0,0 +1 @@
+txt
```

Рисунок 49 - Различия ветвей

10) Слейте новую ветку с master при помощи merge

```
danil@danil-VirtualBox:~/Repository_For_Tests$ git merge newbranch
Обновление a74254d..638fb5b
Fast-forward
 text_file.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 text file.txt
```

Рисунок 50 - Слияние ветвей

Часть 3. Работа с ветвлением и оформление кода

Личный вариант: 6

<https://github.com/airbnb/javascript>

Шаг 1. Форк репозитория

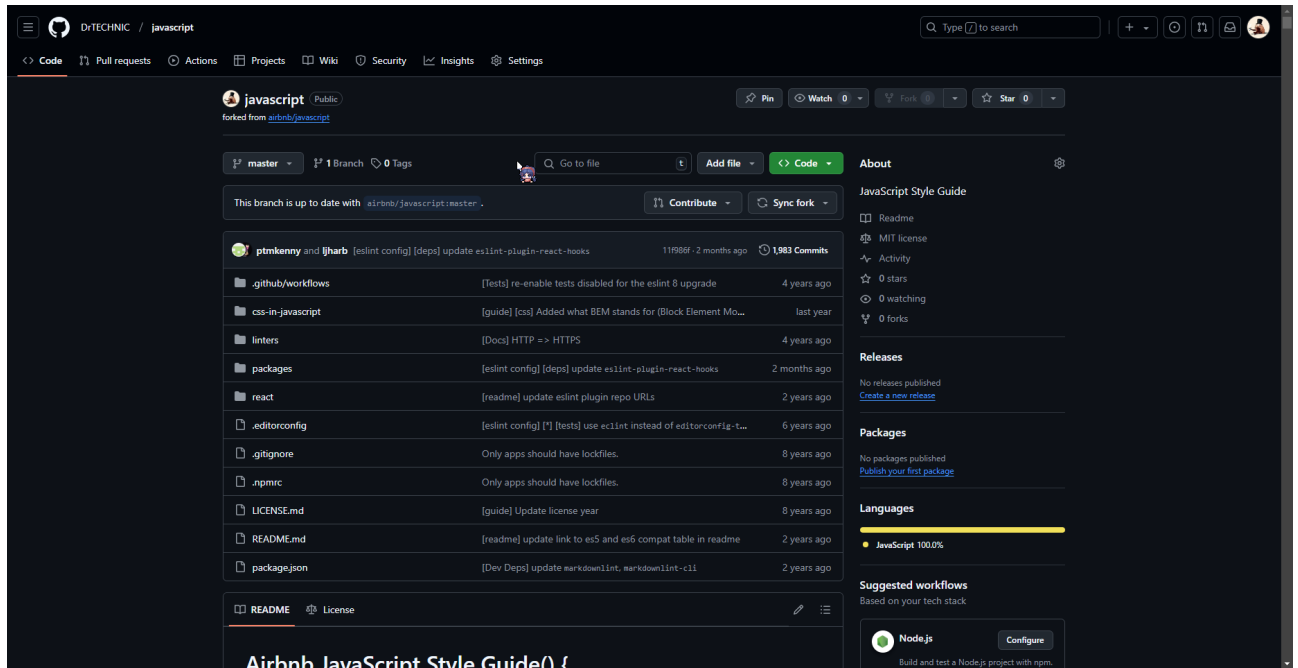


Рисунок 51 - Форк репозитория

Шаг 2. Клонирование репозитория на локальную машину

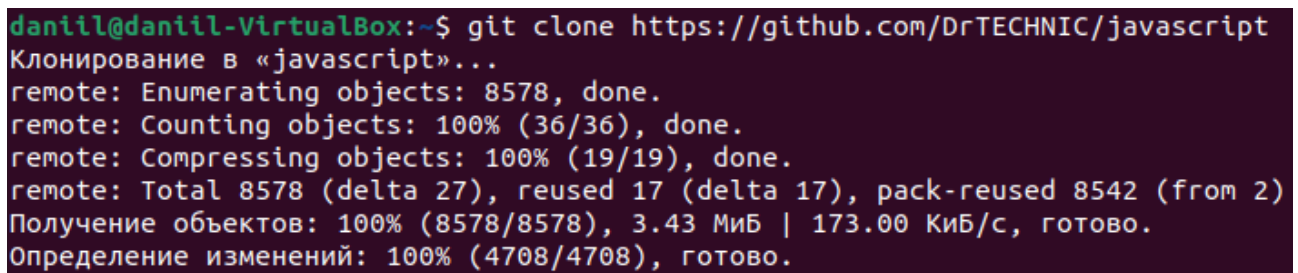


Рисунок 52 - Клонирование репозитория

Шаг 3. Создание двух веток в master'е

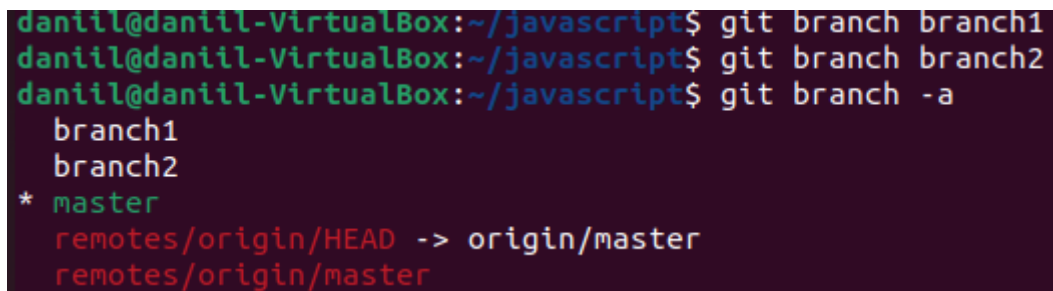


Рисунок 53 - Создание 2 веток

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

```
daniil@daniil-VirtualBox:~/javascript$ git checkout branch1
Переключились на ветку «branch1»
daniil@daniil-VirtualBox:~/javascript$ ls
css-in-javascript  LICENSE.md  linters  package.json  packages  react  README.md
daniil@daniil-VirtualBox:~/javascript$ nano README.md
daniil@daniil-VirtualBox:~/javascript$ git add README.md
daniil@daniil-VirtualBox:~/javascript$ git commit -m "my_first_commit"
[branch1 42c521b] my_first_commit
1 file changed, 1 insertion(+), 1 deletion(-)
daniil@daniil-VirtualBox:~/javascript$ nano README.md
daniil@daniil-VirtualBox:~/javascript$ git add README.md
daniil@daniil-VirtualBox:~/javascript$ git commit -m "my_second_commit"
[branch1 f09ce62] my_second_commit
1 file changed, 1 insertion(+), 1 deletion(-)
daniil@daniil-VirtualBox:~/javascript$ nano README.md
daniil@daniil-VirtualBox:~/javascript$ git add README.md
daniil@daniil-VirtualBox:~/javascript$ git commit -m "my_third_commit"
[branch1 5634966] my_third_commit
1 file changed, 1 insertion(+), 1 deletion(-)
daniil@daniil-VirtualBox:~/javascript$ git checkout branch2
Переключились на ветку «branch2»
daniil@daniil-VirtualBox:~/javascript$ nano README.md
daniil@daniil-VirtualBox:~/javascript$ git add README.md
daniil@daniil-VirtualBox:~/javascript$ git commit -m "start_commit"
[branch2 90fbc62] start_commit
1 file changed, 1 insertion(+), 1 deletion(-)
daniil@daniil-VirtualBox:~/javascript$ nano README.md
daniil@daniil-VirtualBox:~/javascript$ git add README.md
daniil@daniil-VirtualBox:~/javascript$ git commit -m "middle_commit"
[branch2 5bf97f3] middle_commit
1 file changed, 1 insertion(+), 1 deletion(-)
daniil@daniil-VirtualBox:~/javascript$ nano README.md
daniil@daniil-VirtualBox:~/javascript$ git add README.md
daniil@daniil-VirtualBox:~/javascript$ git commit -m "final_commit"
[branch2 281cf6a] final_commit
1 file changed, 1 insertion(+), 1 deletion(-)
```

Рисунок 54 - По 3 коммита в новых ветках

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

```
daniil@daniil-VirtualBox:~/javascript$ git merge branch1
Автослияние README.md
КОНФЛИКТ (содержимое): Конфликт слияния в README.md
Сбой автоматического слияния; исправьте конфликты, затем зафиксируйте результат.
daniil@daniil-VirtualBox:~/javascript$ nano README.md
daniil@daniil-VirtualBox:~/javascript$ nano README.md
daniil@daniil-VirtualBox:~/javascript$ git add README.md
daniil@daniil-VirtualBox:~/javascript$ git commit -m "Разрешение конфликтов между branch1 и branch2"
[branch2 69c3fc6] Разрешение конфликтов между branch1 и branch2
daniil@daniil-VirtualBox:~/javascript$ git status
Текущая ветка: branch2
ничего коммитить, нет изменений в рабочем каталоге
```

Рисунок 55 - Слияние branch1 в branch2

```
<<<<<< HEAD
*Третьи изменения в этой ветке*
=====
*Мои изменения* *Ещё одни изменения* *Последние изменения*
>>>>>> branch1
```

Рисунок 56 - Конфликты

```
# Airbnb JavaScript Style Guide() {
*Мои изменения* *Ещё одни изменения* *Последние изменения*
*A mostly reasonable approach to JavaScript*
```

Рисунок 57 - Разрешение конфликта

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

```
danil@danil-VirtualBox:~/javascript$ git push -u origin master
Username for 'https://github.com': DrTECHNIC
Password for 'https://DrTECHNIC@github.com':
Ветка «master» отслеживает внешнюю ветку «master» из «origin».
Everything up-to-date
danil@danil-VirtualBox:~/javascript$ git push -u origin branch1
Username for 'https://github.com': DrTECHNIC
Password for 'https://DrTECHNIC@github.com':
Перечисление объектов: 11, готово.
Подсчет объектов: 100% (11/11), готово.
При сжатии изменений используется до 5 потоков
Сжатие объектов: 100% (9/9), готово.
Запись объектов: 100% (9/9), 896 байтов | 896.00 КиБ/с, готово.
Всего 9 (изменений 6), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (6/6), completed with 2 local objects.
remote:
remote: Create a pull request for 'branch1' on GitHub by visiting:
remote:   https://github.com/DrTECHNIC/javascript/pull/new/branch1
remote:
To https://github.com/DrTECHNIC/javascript
 * [new branch]   branch1 -> branch1
Ветка «branch1» отслеживает внешнюю ветку «branch1» из «origin».
danil@danil-VirtualBox:~/javascript$ git push -u origin branch2
Username for 'https://github.com': DrTECHNIC
Password for 'https://DrTECHNIC@github.com':
Перечисление объектов: 14, готово.
Подсчет объектов: 100% (14/14), готово.
При сжатии изменений используется до 5 потоков
Сжатие объектов: 100% (10/10), готово.
Запись объектов: 100% (10/10), 1.12 КиБ | 1.12 МиБ/с, готово.
Всего 10 (изменений 6), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (6/6), completed with 2 local objects.
remote:
remote: Create a pull request for 'branch2' on GitHub by visiting:
remote:   https://github.com/DrTECHNIC/javascript/pull/new/branch2
remote:
To https://github.com/DrTECHNIC/javascript
 * [new branch]   branch2 -> branch2
Ветка «branch2» отслеживает внешнюю ветку «branch2» из «origin».
```

Рисунок 58 - Выгрузка всех изменений

Шаг 7. Выполнение 3-х коммитов в branch1

```
danil@danil-VirtualBox:~/javascript$ git checkout branch1
Переключились на ветку «branch1»
Эта ветка соответствует «origin/branch1».
danil@danil-VirtualBox:~/javascript$ touch text1.txt
danil@danil-VirtualBox:~/javascript$ git add text1.txt
danil@danil-VirtualBox:~/javascript$ git commit "1more"
error: pathspec '1more' did not match any file(s) known to git
danil@danil-VirtualBox:~/javascript$ git commit -m "1more"
[branch1 77b04b2] 1more
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 text1.txt
danil@danil-VirtualBox:~/javascript$ touch text2.txt
danil@danil-VirtualBox:~/javascript$ touch nottext2
danil@danil-VirtualBox:~/javascript$ git add text2.txt
danil@danil-VirtualBox:~/javascript$ git add nottext2
danil@danil-VirtualBox:~/javascript$ git commit -m "2more"
[branch1 76aec99] 2more
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 nottext2
create mode 100644 text2.txt
danil@danil-VirtualBox:~/javascript$ nano text2.txt
danil@danil-VirtualBox:~/javascript$ nano text1.txt
danil@danil-VirtualBox:~/javascript$ git add text1.txt
danil@danil-VirtualBox:~/javascript$ git add text2.txt
danil@danil-VirtualBox:~/javascript$ git commit -m "3more"
[branch1 3dd3e5e] 3more
2 files changed, 2 insertions(+)
```

Рисунок 59 - Выполнение 3 коммитов в branch1

Шаг 8. Повторное клонирование репозитория в другую директорию

```
danil@danil-VirtualBox:~/javascript$ cd ..
danil@danil-VirtualBox:~$ ls
javascript  Repository_For_Tests  TRPP      Видео      Загрузки  Музыка      'Рабочий стол'
project     snap                  vrazhenko  Документы  Изображения  Общедоступные  Шаблоны
danil@danil-VirtualBox:~$ mkdir onemorerep
danil@danil-VirtualBox:~$ cd onemorerep/
danil@danil-VirtualBox:~/onemorerep$ git clone https://github.com/DrTECHNIC/javascript
Клонирование в «javascript»...
remote: Enumerating objects: 8597, done.
remote: Counting objects: 100% (35/35), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 8597 (delta 25), reused 22 (delta 17), pack-reused 8562 (from 3)
Получение объектов: 100% (8597/8597), 3.46 Миб | 170.00 КиБ/с, готово.
Определение изменений: 100% (4718/4718), готово.
```

Рисунок 60 - Повторное клонирование репозитория

Шаг 9. Выполнение 3-х коммитов в ветку branch1

```
danil@danil-VirtualBox:~/onemorerep/javascript$ git checkout -b branch1
Переключились на новую ветку «branch1»
danil@danil-VirtualBox:~/onemorerep/javascript$ touch 123
danil@danil-VirtualBox:~/onemorerep/javascript$ git add 123
danil@danil-VirtualBox:~/onemorerep/javascript$ git commit -m "1"
[branch1 fa2d1ab] 1
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 123
danil@danil-VirtualBox:~/onemorerep/javascript$ nano 123
danil@danil-VirtualBox:~/onemorerep/javascript$ touch 246
danil@danil-VirtualBox:~/onemorerep/javascript$ git add 123
danil@danil-VirtualBox:~/onemorerep/javascript$ git add 246
danil@danil-VirtualBox:~/onemorerep/javascript$ git commit -m "2"
[branch1 4e9b3eb] 2
2 files changed, 1 insertion(+)
create mode 100644 246
danil@danil-VirtualBox:~/onemorerep/javascript$ touch 369
danil@danil-VirtualBox:~/onemorerep/javascript$ nano 369
danil@danil-VirtualBox:~/onemorerep/javascript$ git add 369
danil@danil-VirtualBox:~/onemorerep/javascript$ git commit -m "3"
[branch1 1e77b67] 3
1 file changed, 1 insertion(+)
create mode 100644 369
```

Рисунок 61 - Выполнение 3 коммитов

Шаг 10. Выгрузка всех изменений в удаленный репозиторий

```
danil@danil-VirtualBox:~/onemorerep/javascript$ git pull origin branch1 --rebase
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 9 (delta 3), reused 9 (delta 3), pack-reused 0 (from 0)
Распаковка объектов: 100% (9/9), 754 байта | 188.00 КиБ/с, готово.
Из https://github.com/DrTECHNIC/javascript
* branch          branch1      -> FETCH_HEAD
+ 1e77b67...3dd3e5e branch1     -> origin/branch1 (принудительное обновление)
Успешно перемещён и обновлён refs/heads/branch1.
danil@danil-VirtualBox:~/onemorerep/javascript$ git push origin branch1
Username for 'https://github.com': DrTECHNIC
Password for 'https://DrTECHNIC@github.com':
Everything up-to-date
```

Рисунок 62 - Выгрузка изменений

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

```
danil@danil-VirtualBox:~/javascript$ git push --force origin branch1
Username for 'https://github.com': DrTECHNIC
Password for 'https://DrTECHNIC@github.com':
Перечисление объектов: 10, готово.
Подсчет объектов: 100% (10/10), готово.
При сжатии изменений используется до 5 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (9/9), 774 байта | 387.00 КиБ/с, готово.
Всего 9 (изменений 3), повторно использовано 1 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To https://github.com/DrTECHNIC/javascript
+ 1e77b67...3dd3e5e branch1 -> branch1 (forced update)
```

Рисунок 63 - Принудительная выгрузка

Шаг 12. Получение всех изменения в новом репозитории

```
danil@danil-VirtualBox:~/onemorerep/javascript$ git pull origin master
Из https://github.com/DrTECHNIC/javascript
* branch          master      -> FETCH_HEAD
Уже актуально.
```

Рисунок 64 - Получение всех изменений

КОНТРОЛЬНЫЕ ВОПРОСЫ

3. Что такое репозиторий Git?

Репозиторий Git — это хранилище, в котором находятся файлы проекта и вся его история изменений. Он содержит коммиты, ветки и другие метаданные, необходимые для отслеживания версий кода. Репозиторий может быть локальным (на вашем компьютере) или удалённым (например, на GitHub, GitLab, Bitbucket).

4. Что такое коммит?

Коммит (commit) — это сохранённое состояние файлов в репозитории на определённый момент времени. Он фиксирует изменения и добавляет их в историю проекта. У каждого коммита есть уникальный идентификатор (хеш) и комментарий, описывающий суть изменений.

5. Что такое ветка в репозитории Git?

Ветка (branch) — это независимая линия разработки в Git. Она позволяет вносить изменения, не затрагивая основную ветку (обычно main или master). Ветки используются для работы над новыми функциями, исправления багов и тестирования, после чего они могут быть объединены с основной веткой с помощью merge или rebase.

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

Конфликт (merge conflict) возникает, когда два коммита изменяют один и тот же участок кода в разных ветках. Git не может автоматически объединить изменения и требует вмешательства разработчика. Конфликты возникают из-за параллельных изменений в одном файле и могут быть вызваны работой нескольких разработчиков над одним участком кода.

11. Что делает команда git status?

Показывает текущее состояние репозитория.

12. Что делает команда git add?

Добавляет изменения в индекс для коммита.

13. Что делает команда `git log`?

Отображает историю коммитов.

14. Что делает команда `git diff`?

Показывает разницу между версиями файлов.

ВЫВОД

В ходе выполнения практической работы были изучены основные команды Git, принципы работы с системами управления репозиториями, а также механизмы ветвления и оформления кода. Освоены команды для инициализации, коммитов, работы с ветками и разрешения конфликтов. Полученные навыки позволяют эффективно управлять версиями кода и организовывать командную разработку.