



ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ

ГОРОДА МОСКВЫ

Государственное бюджетное профессиональное
образовательное учреждение города Москвы

«Колледж малого бизнеса № 4»

(ГБПОУ КМБ № 4)

Отчёт по лабораторной работе №7

Специальность: 09.02.07 Информационные системы и программирование

Форма обучения: очная

Студент: Межибор Ярослав Евгеньевич

Группа: ИПО-21.24

Проверил: Рыбаков Александр Сергеевич

Москва, 2025 г.

Цель работы

Изучить механизмы ветвления в Git как инструмента для параллельной разработки. Научиться создавать, переключать и удалять ветки, а также освоить способы объединения изменений и разрешения возникающих при этом конфликтов.

Задачи

1. Отработать команды просмотра веток (git branch, git branch -v) и навигации между ними (git checkout).
2. Практически освоить создание новых веток для решения локальных задач (например, ветки iss53 и hotfix).
3. Изучить механизм слияния веток с помощью команды git merge и понять принцип работы метода fast-forward.
4. Смоделировать ситуацию возникновения конфликта при попытке слияния веток, содержащих разные изменения в одних и тех же строках файла.
5. Научиться вручную устранять конфликты в коде, завершать слияние и проводить финальную фиксацию изменений.
6. Выполнить самостоятельную работу по созданию персонального профиля и управлению его версиями через ветки в удаленном репозитории GitHub.

Ход работы

```
yarom@Huawei_laptop MINGW64 /d/ProGit
$ cd D:/ProGit/branches

yarom@Huawei_laptop MINGW64 /d/ProGit/branches
$ git init
Initialized empty Git repository in D:/ProGit/branches/.git/

yarom@Huawei_laptop MINGW64 /d/ProGit/branches (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

Рис. 1. Создание новой папки, инициализация репозитория и проверка статуса

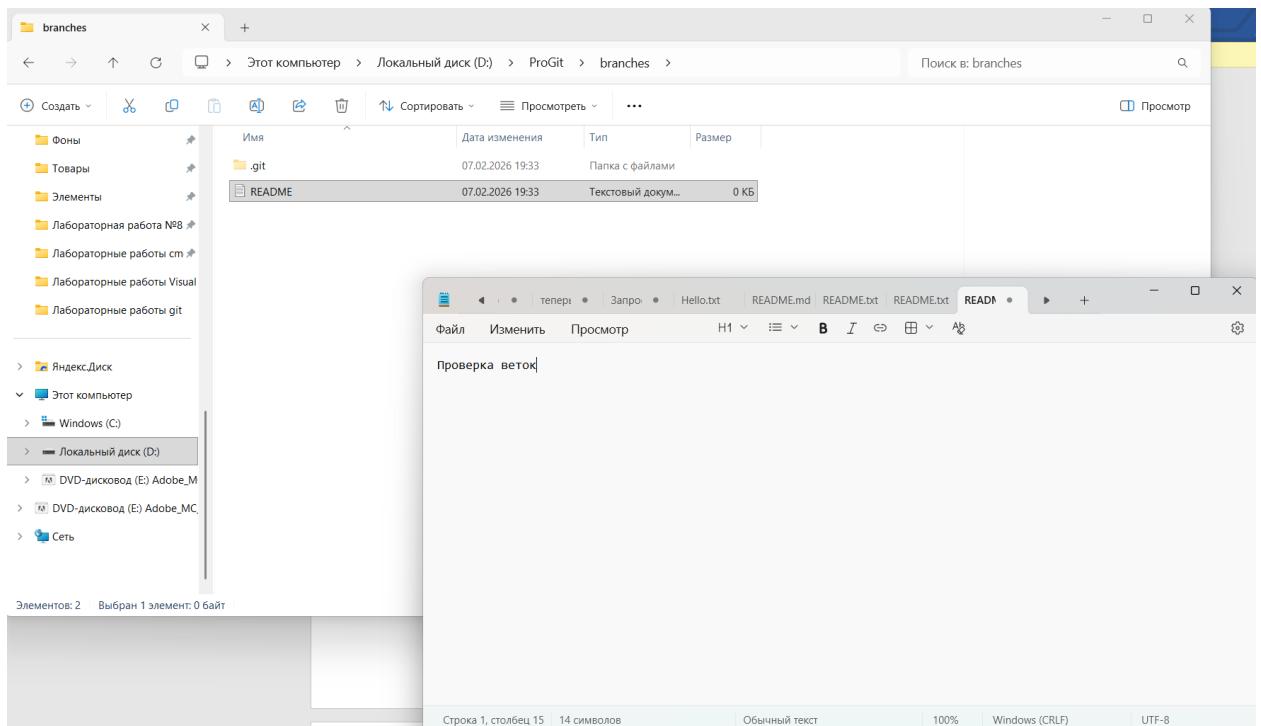


Рис. 2. Создание файла в созданной папке-репозитории и его редактирование

```
yarom@Huawei_laptop MINGW64 /d/ProGit/branches (master)
$ git add .

yarom@Huawei_laptop MINGW64 /d/ProGit/branches (master)
$ git commit -m "Создал и изменил файл"
[master (root-commit) 3abaa50] Создал и изменил файл
 1 file changed, 1 insertion(+)
 create mode 100644 README.txt

yarom@Huawei_laptop MINGW64 /d/ProGit/branches (master)
$ git status
On branch master
nothing to commit, working tree clean
```

Рис. 3. Добавил и закоммитил его

```
yarom@Huawei_laptop MINGW64 /d/ProGit/branches (master)
$ git branch testing

yarom@Huawei_laptop MINGW64 /d/ProGit/branches (master)
$ git checkout testing
Switched to branch 'testing'
```

Рис. 4. Создал ветку и перешел на нее

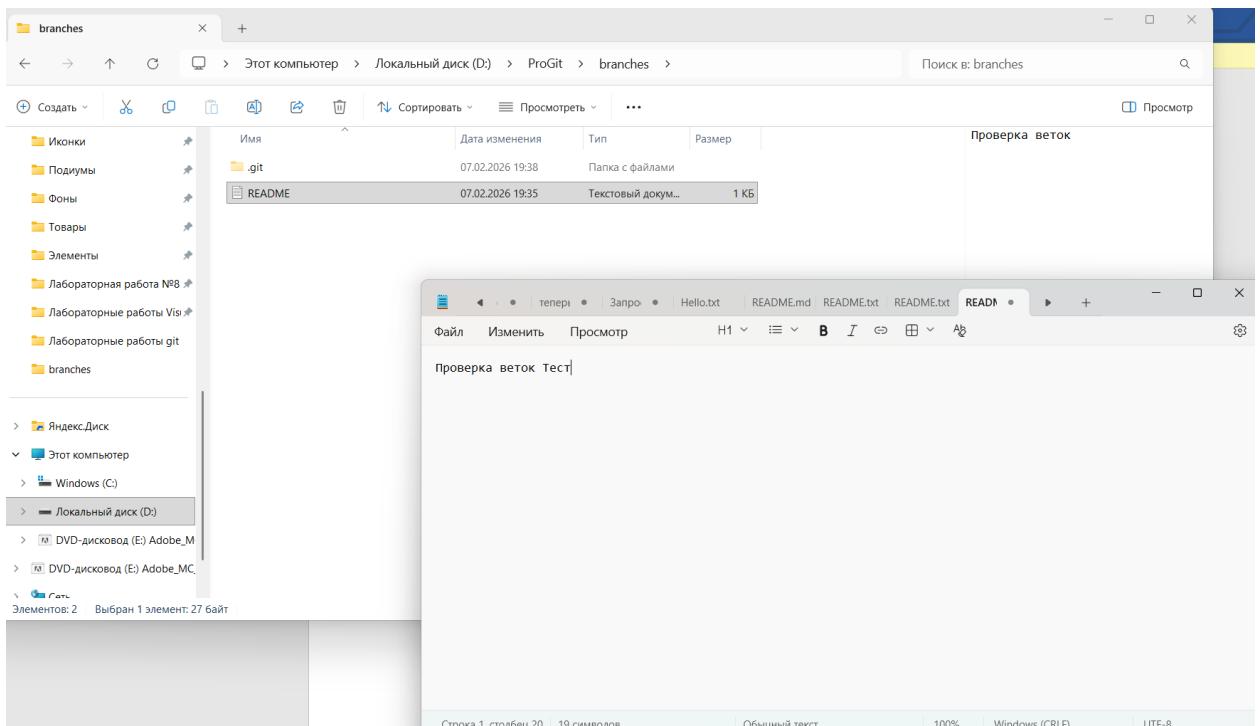


Рис. 5. Новое изменение файла

```

yarom@Huawei_laptop MINGW64 /d/ProGit/branches (testing)
$ git add .

yarom@Huawei_laptop MINGW64 /d/ProGit/branches (testing)
$ git commit -m "Коммит в ветке testing"
[testing bb343ed] Коммит в ветке testing
 1 file changed, 1 insertion(+), 1 deletion(-)

yarom@Huawei_laptop MINGW64 /d/ProGit/branches (testing)
$ git log
commit bb343ed1bba7eeb222688f72c7fbaf279c8cc805 (HEAD -> testing)
Author: Yarik <yaromezh@gmail.com>
Date:   Sat Feb 7 19:40:31 2026 +0300

    Коммит в ветке testing

commit 3a6a505b6ec0ee4cbe6ffd863505e0afb50cce96b (master)
Author: Yarik <yaromezh@gmail.com>
Date:   Sat Feb 7 19:36:08 2026 +0300

    Создал и изменил файл

```

Рис. 6. Добавил, закоммитил и просмотрел историю коммитов файла

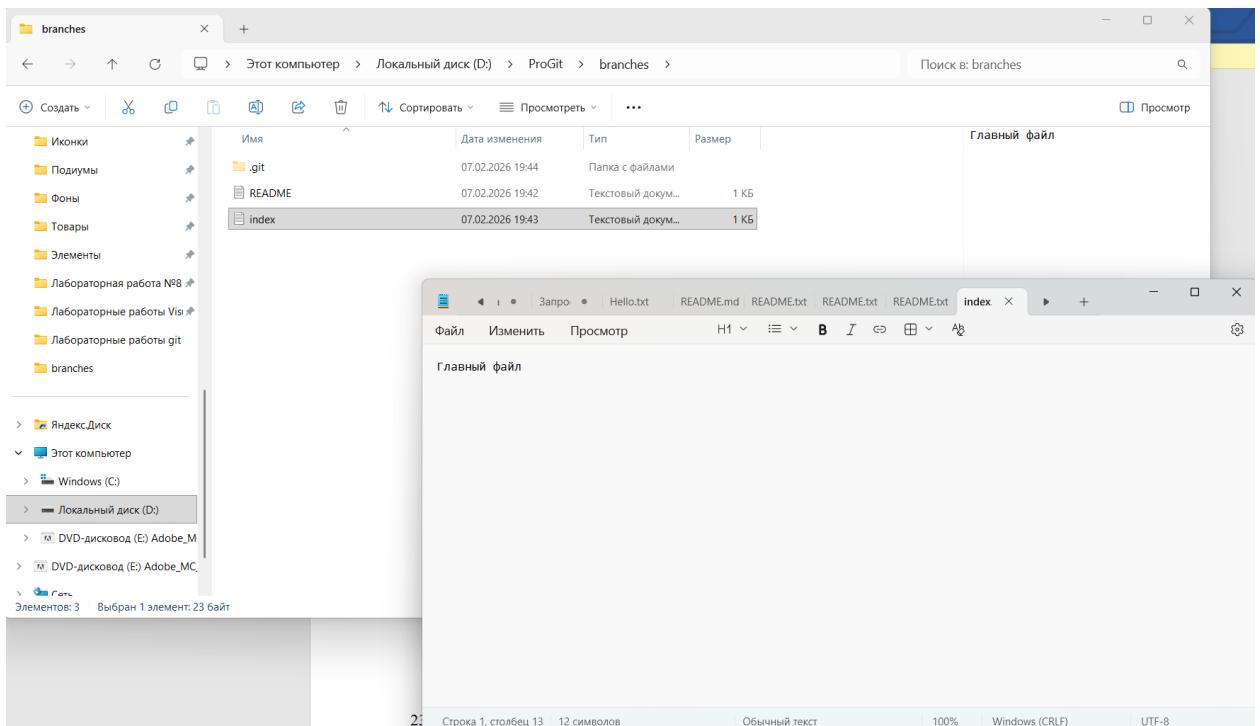


Рис. 7. Создал новый файл и написал в нем

```
yarom@Huawei_laptop MINGW64 /d/ProGit/branches (master)
$ git add .

yarom@Huawei_laptop MINGW64 /d/ProGit/branches (master)
$ git commit -m "Коммит в ветке master"
[master 1fe3231] Коммит в ветке master
 1 file changed, 1 insertion(+)
  create mode 100644 index.txt
```

Рис. 8. Добавил и закоммитил его

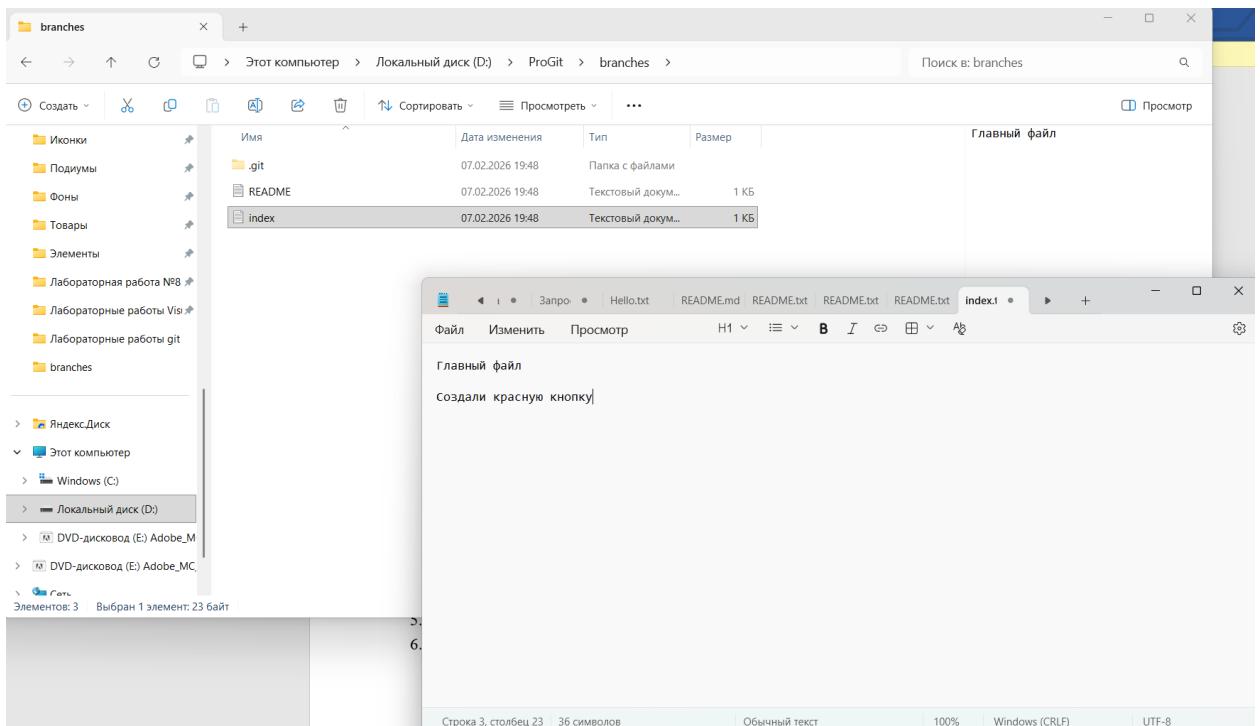


Рис. 9. Изменил файл

```

yarom@Huawei_laptop MINGW64 /d/ProGit/branches (testing)
$ git branch hotfix

yarom@Huawei_laptop MINGW64 /d/ProGit/branches (testing)
$ git checkout master
Switched to branch 'master'

yarom@Huawei_laptop MINGW64 /d/ProGit/branches (master)
$ git status
On branch master
nothing to commit, working tree clean

yarom@Huawei_laptop MINGW64 /d/ProGit/branches (master)
$ git add .

yarom@Huawei_laptop MINGW64 /d/ProGit/branches (master)
$ git commit -m "Создали красную кнопку"
[master c04bc26] Создали красную кнопку
 1 file changed, 3 insertions(+), 1 deletion(-)

```

Рис. 10. Создал новую ветку, добавил и закоммитил все

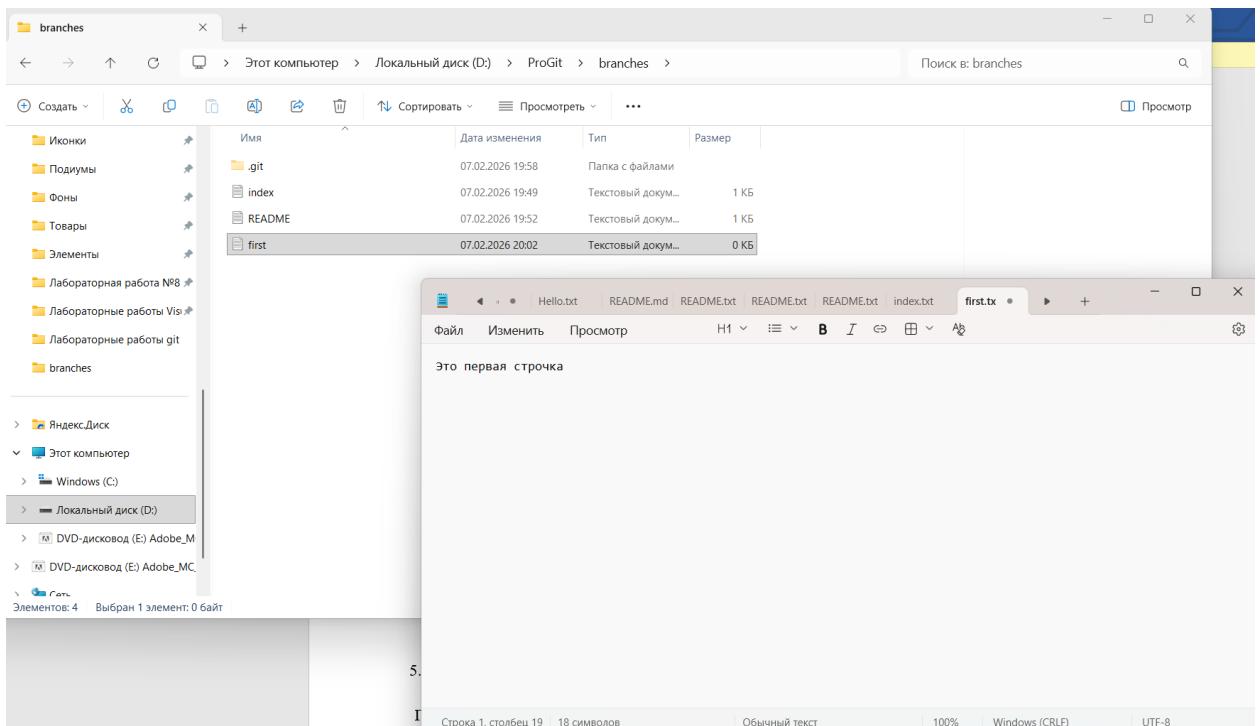


Рис. 11. Создание нового файла и его редактирование

```
yarom@Huawei_laptop MINGW64 /d/ProGit/branches (master|MERGING)
$ git add .

yarom@Huawei_laptop MINGW64 /d/ProGit/branches (master|MERGING)
$ git commit -m "Создали файл"
[master d8d9640] Создали файл

yarom@Huawei_laptop MINGW64 /d/ProGit/branches (master)
```

Рис. 12. Добавил и закоммитил файл

```
yarom@Huawei_laptop MINGW64 /d/ProGit/branches (master)
$ git branch feature/1

yarom@Huawei_laptop MINGW64 /d/ProGit/branches (master)
$ git checkout -b 'feature/1'
fatal: a branch named 'feature/1' already exists

yarom@Huawei_laptop MINGW64 /d/ProGit/branches (master)
$ git checkout feature/1
Switched to branch 'feature/1'
```

Рис. 13. Создал новую ветку и переключился на нее

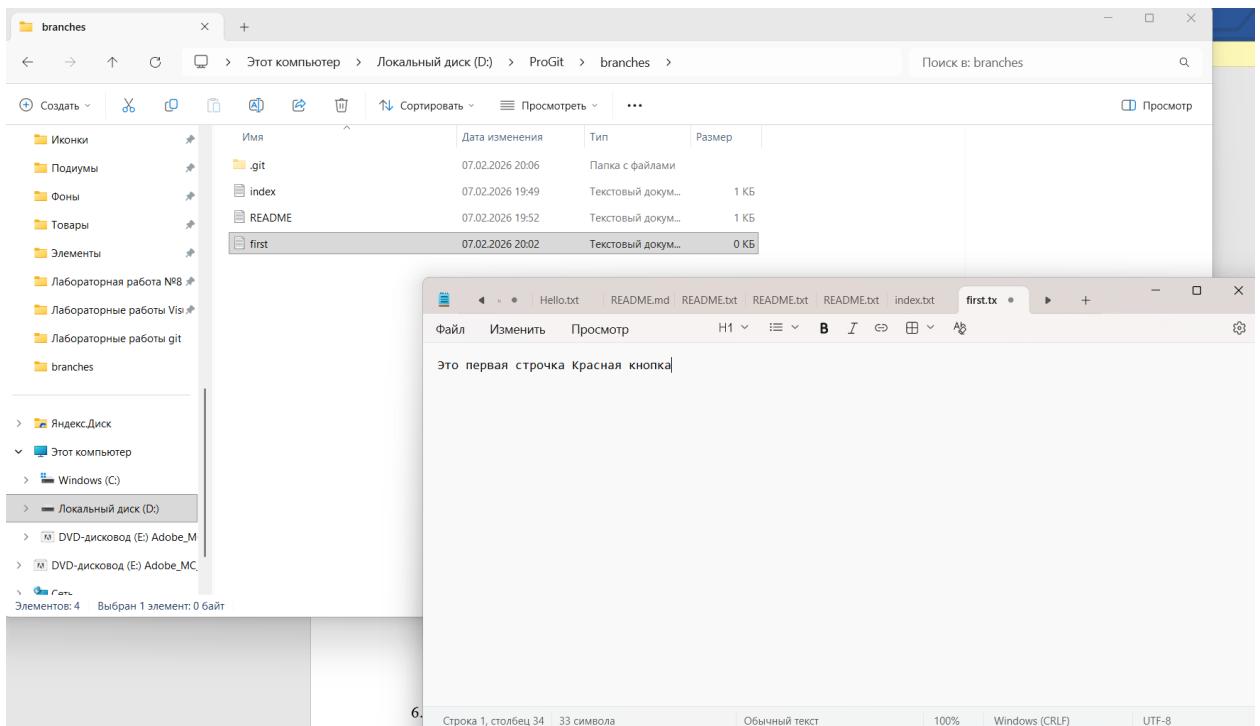


Рис. 14. Изменил файл

```
yarom@Huawei_laptop MINGW64 /d/ProGit/branches (feature/1)
$ git add .

yarom@Huawei_laptop MINGW64 /d/ProGit/branches (feature/1)
$ git commit -m "Создали красную кнопку"
[feature/1 a051db1] Создали красную кнопку
 1 file changed, 1 insertion(+)
```

Рис. 15. Добавил и закоммитил

```
yarom@Huawei_laptop MINGW64 /d/ProGit/branches (master)
$ git branch feature/2

yarom@Huawei_laptop MINGW64 /d/ProGit/branches (master)
$ git checkout feature/2
Switched to branch 'feature/2'
```

Рис. 16. Создал новую ветку и переключился на нее

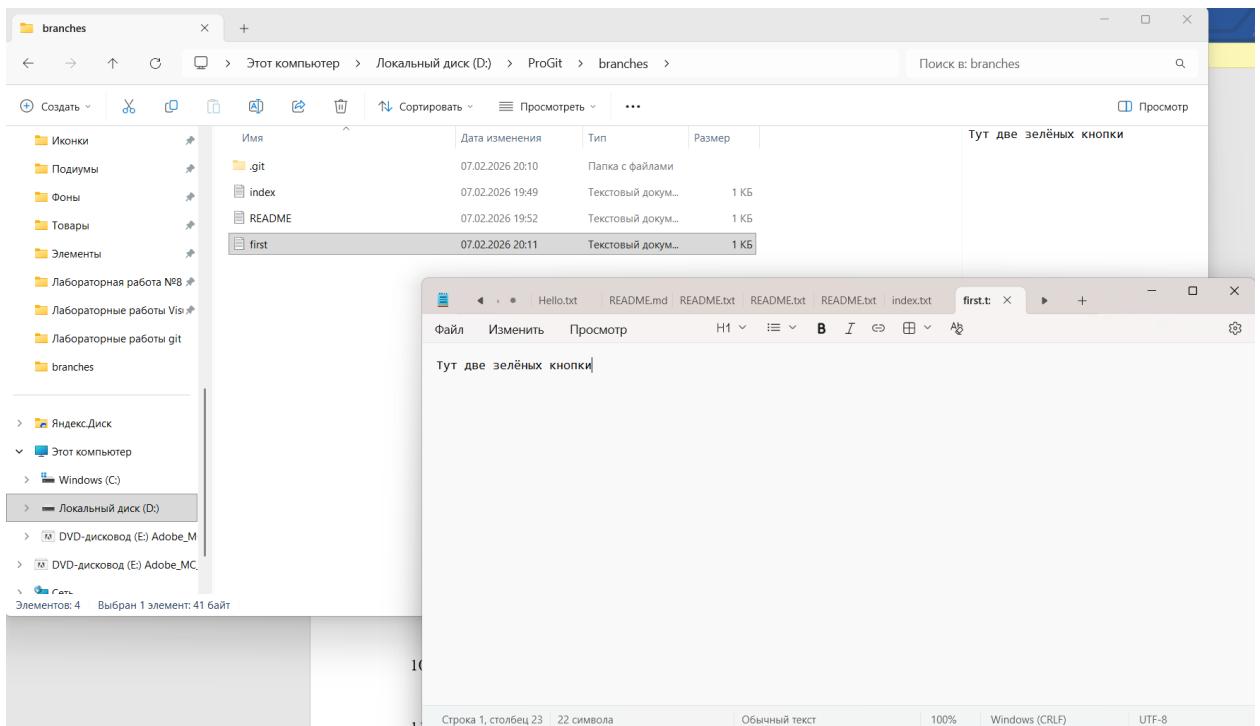


Рис. 17. Изменил файл

```
yarom@Huawei_laptop MINGW64 /d/ProGit/branches (feature/2)
$ git add .

yarom@Huawei_laptop MINGW64 /d/ProGit/branches (feature/2)
$ git commit -m "Создали две зеленые кнопки"
[feature/2 988a020] Создали две зеленые кнопки
 1 file changed, 1 insertion(+)
```

Рис. 18. Добавил и закоммитил

```
yarom@Huawei_laptop MINGW64 /d/ProGit/branches (feature/2)
$ git checkout feature/1
Switched to branch 'feature/1'

yarom@Huawei_laptop MINGW64 /d/ProGit/branches (feature/1)
$ git merge feature/2
Auto-merging first.txt
CONFLICT (content): Merge conflict in first.txt
Automatic merge failed; fix conflicts and then commit the result.
```

Рис. 19. Переключился на ветку и попытался слить ее с другой

Файл Изменить Просмотр H1 B I A Строки 2, столбец 10 | 96 символов | Обычный текст | 100% | Windows (CRLF) | UTF-8

```
<<<<< HEAD
Это первая строчка Красная кнопка
=====
Тут две зелёных кнопки
>>>> feature/2
```

Рис. 20. Что-то непонятное произошло

```
yarom@Huawei_laptop MINGW64 /d/ProGit/branches (feature/1|MERGING)
$ git add .

yarom@Huawei_laptop MINGW64 /d/ProGit/branches (feature/1|MERGING)
$ git commit -m "Конфликт решен"
[feature/1 9b0fc22] Конфликт решен
```

Рис. 21. Проиндексировал и закоммитил файл

Домашняя работа

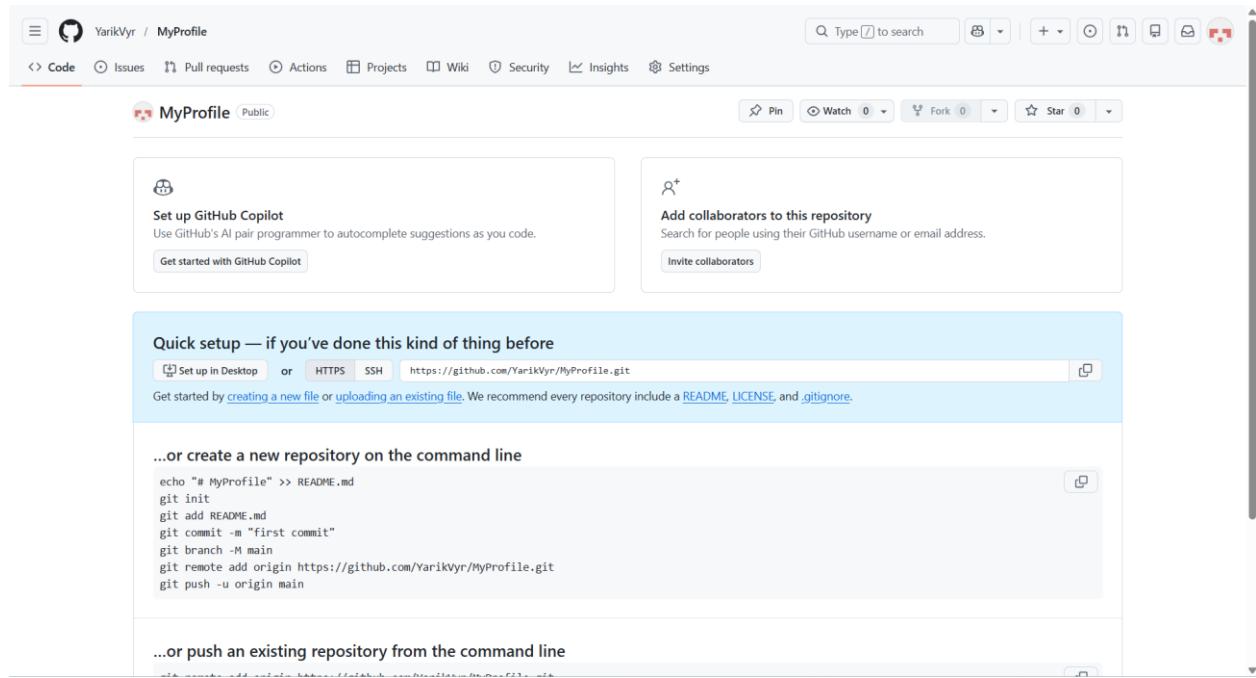


Рис. 22. Создал новый репо

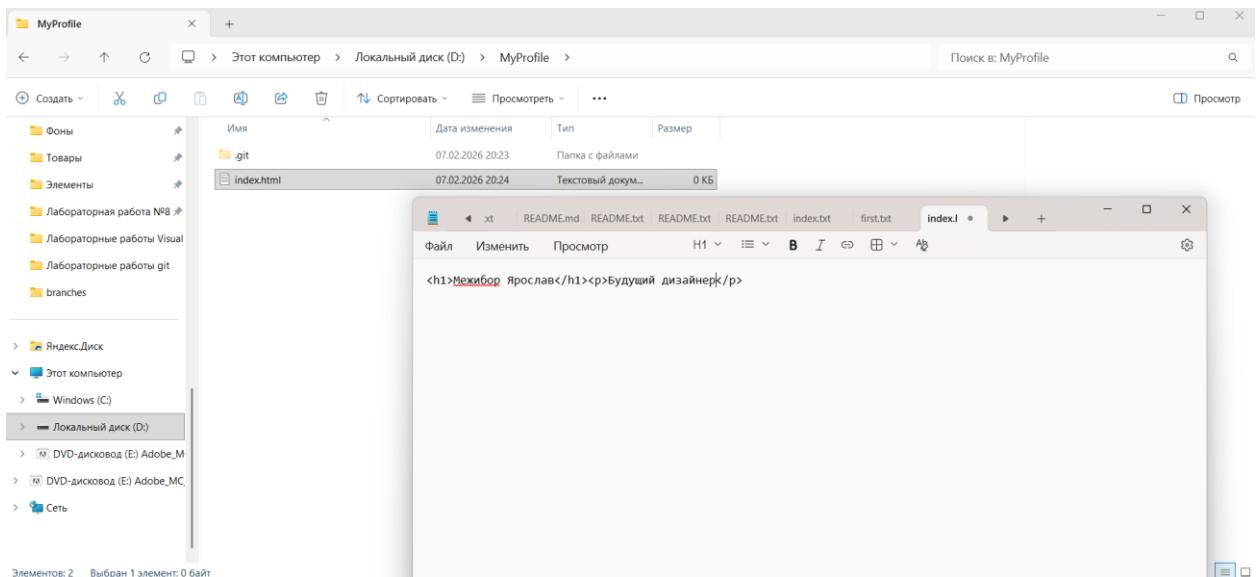


Рис. 23. Создал новый файл и изменил его

```
yarom@Huawei_laptop MINGW64 ~
$ cd D:/MyProfile

yarom@Huawei_laptop MINGW64 /d/MyProfile
$ git init
Initialized empty Git repository in D:/MyProfile/.git/

yarom@Huawei_laptop MINGW64 /d/MyProfile (master)
$ git remote add origin https://github.com/YarikVyr/MyProfile.git
```

Рис. 24. Создал и инициализировал новый репо, указав путь на GitHub

```
yarom@Huawei_laptop MINGW64 /d/MyProfile (master)
$ git branch button

yarom@Huawei_laptop MINGW64 /d/MyProfile (master)
$ git checkout button
Switched to branch 'button'
```

Рис. 25. Создал новую ветку и переключился на нее

```
<h1>Межибор Ярослав</h1><p>Будущий дизайнер</p>
<button>Подписаться на новости</button>
```

Рис. 26. Изменил файл

```
yarom@Huawei_laptop MINGW64 /d/MyProfile (button)
$ git push origin button
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 389 bytes | 389.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'button' on GitHub by visiting:
remote:     https://github.com/YarikVyr/MyProfile/pull/new/button
remote:
To https://github.com/YarikVyr/MyProfile.git
 * [new branch]      button -> button
```

Рис. 27. Запушил изменения на GitHub

```
yarom@Huawei_laptop MINGW64 /d/MyProfile (button)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

yarom@Huawei_laptop MINGW64 /d/MyProfile (master)
$ git merge button
Updating a5667e6..0e837c6
Fast-forward
 index.html.txt | 3 +-+
 1 file changed, 2 insertions(+), 1 deletion(-)

yarom@Huawei_laptop MINGW64 /d/MyProfile (master)
$ git push origin master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/YarikVyr/MyProfile.git
    a5667e6..0e837c6  master -> master
```

Рис. 28. Переключился обратно на ветку, слился с основной и запушил

```
yarom@Huawei_laptop MINGW64 /d/MyProfile (master)
$ git branch color

yarom@Huawei_laptop MINGW64 /d/MyProfile (master)
$ git checkout color
Switched to branch 'color'
```

Рис. 29. Создал новую ветку

```
yarom@Huawei_laptop MINGW64 /d/MyProfile (color)
$ git add .

yarom@Huawei_laptop MINGW64 /d/MyProfile (color)
$ git commit -m "Поменял цвет"
[color faab4a3] Поменял цвет
 1 file changed, 2 insertions(+), 1 deletion(-)
```

Рис. 30. Изменил и закоммитил

```
yarom@Huawei_laptop MINGW64 /d/MyProfile (color)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

yarom@Huawei_laptop MINGW64 /d/MyProfile (master)
$ git branch login

yarom@Huawei_laptop MINGW64 /d/MyProfile (master)
$ git checkout login
Switched to branch 'login'
```

Рис. 31. Переключился на ветку обратно, создал в ней новую и переключился на нее

```
yarom@Huawei_laptop MINGW64 /d/MyProfile (login)
$ git commit -a -m "Добавил вход через соцсети"
[login 50f2611] Добавил вход через соцсети
 1 file changed, 2 insertions(+), 1 deletion(-)
```

Рис. 32. Закоммитил все это

```
yarom@Huawei_laptop MINGW64 /d/MyProfile (master)
$ git merge color
Updating 0e837c6..faab4a3
Fast-forward
 index.html.txt | 3 +-+
 1 file changed, 2 insertions(+), 1 deletion(-)

yarom@Huawei_laptop MINGW64 /d/MyProfile (master)
$ git merge login
```

Рис. 33. Сливаем ветки вместе

```
yarom@Huawei_laptop MINGW64 /d/MyProfile (master)
$ git merge login
Auto-merging index.html.txt
CONFLICT (content): Merge conflict in index.html.txt
Automatic merge failed; fix conflicts and then commit the result.
```

Рис. 34. Уфф, что же это

```
<h1>Межибор Ярослав</h1><p>Будущий дизайнер</p>
<button>Подписаться на новости</button>
style="background: red;
<p>Форма входа через VK</p>
```

Рис. 35. Пришло исправлять вручную

```
yarom@Huawei_laptop MINGW64 /d/MyProfile (master|MERGING)
$ git add .

yarom@Huawei_laptop MINGW64 /d/MyProfile (master|MERGING)
$ git commit -m "я все починил"
[master 7162111] я все починил

yarom@Huawei_Laptop MINGW64 /d/MyProfile (master)
$ git push origin master
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 910 bytes | 910.00 KiB/s, done.
Total 9 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To https://github.com/YarikVyr/MyProfile.git
  0e837c6..7162111  master -> master
```

Рис. 36. Починил и запушил

Контрольные вопросы

1. Ветка master.
2. С помощью команды `git branch [имя_ветки]`.
3. HEAD - это указатель на текущую активную ветку и коммит. Команда `git checkout` перемещает этот указатель на выбранную ветку, делая её рабочей.
4. Нужно перейти в ветку, которая принимает изменения и выполнить команду `git merge [имя_сливаемой_ветки]`.
5. Это тип слияния, при котором целевая ветка просто перемещается вперед на коммит сливаемой ветки без создания нового узла, так как истории не разошлись.

Вывод

В ходе выполнения работы я освоил технологию ветвления. Я научился разделять задачи по разным веткам, что позволяет сохранять стабильность основной ветки (master) во время работы над новыми функциями или исправлениями. На практике был изучен процесс разрешения конфликтов слияния, который требует внимательного анализа изменений и ручного редактирования кода.