

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития  
Кафедра инфокоммуникаций

**«Основы ветвления Git»**

**ОТЧЕТ**  
**по лабораторной работе №3**  
**дисциплины**  
**«Основы программной инженерии»**

Выполнила:

Кувшин Ирина Анатольевна

2 курс, группа ПИЖ-б-о-21-1,

09.03.04 «Программная инженерия»,

направленность (профиль) «Разработка

и сопровождение программного

обеспечения», очная форма обучения

---

(подпись)

Проверил:

---

(подпись)

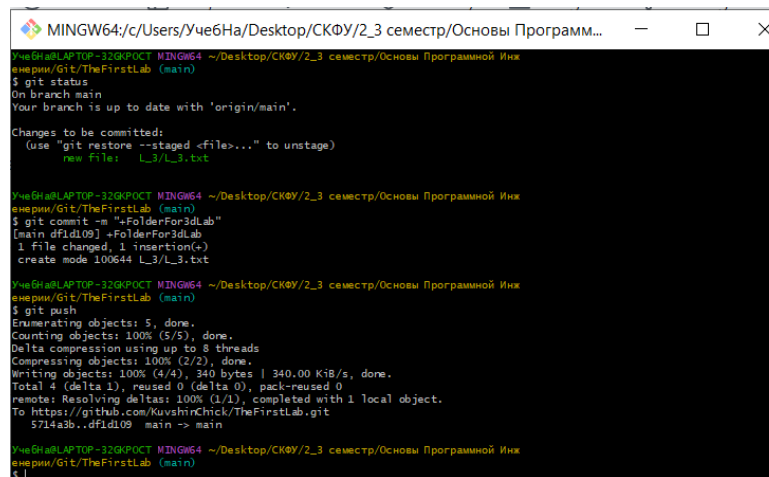
Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2022 г.

**Цель работы:** исследование базовых возможностей по работе с локальными и удаленными ветками Git.

**Ссылка на репозиторий:** <https://github.com/KuvshinChick/TheFirstLab.git>

**Ход работы:**



```
MINGW64~/Desktop/СКФУ/2_3 семестр/Основы Программной Инж
eиeиrиrи/Git/TheFirstLab (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   L_3/L_3.txt

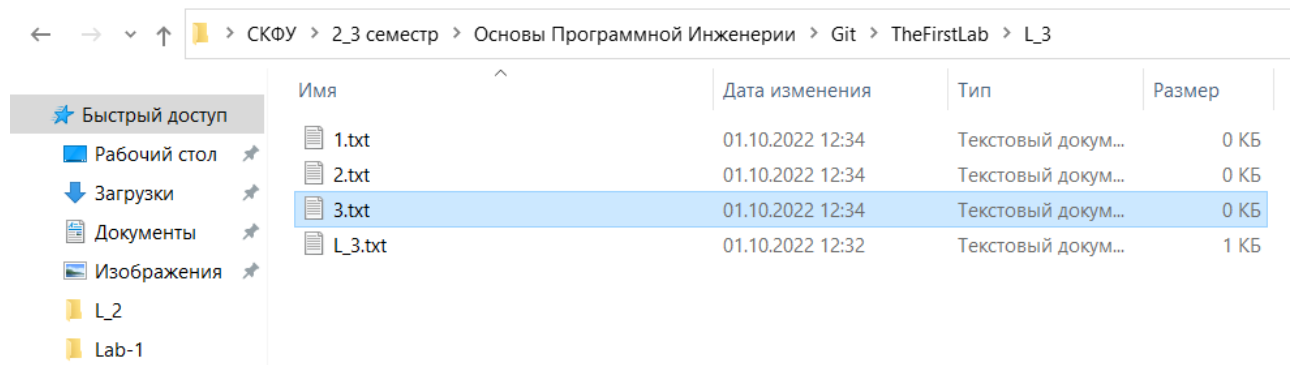
MINGW64~/Desktop/СКФУ/2_3 семестр/Основы Программной Инж
eиeиrиrи/Git/TheFirstLab (main)
$ git commit -m "+FolderFor3dLab"
[main df1d109] +FolderFor3dLab
1 file changed, 1 insertion(+)
create mode 100644 L_3/L_3.txt

MINGW64~/Desktop/СКФУ/2_3 семестр/Основы Программной Инж
eиeиrиrи/Git/TheFirstLab (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 340 bytes | 340.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/KuvshinChick/TheFirstLab.git
 5714a3b..df1d109  main -> main

MINGW64~/Desktop/СКФУ/2_3 семестр/Основы Программной Инж
eиeиrиrи/Git/TheFirstLab (main)
$
```

Рисунок 3.1 – Создание папки для ЛР\_3

3. Создать три файла: 1.txt, 2.txt, 3.txt.
4. Проиндексировать первый файл и сделать коммит с комментарием "add 1.txt file".
5. Проиндексировать второй и третий файлы.
6. Перезаписать уже сделанный коммит с новым комментарием "add 2.txt and 3.txt."



← → ▾ ▴ 📁 > СКФУ > 2_3 семестр > Основы Программной Инженерии > Git > TheFirstLab > L_3				
Имя		Дата изменения	Тип	Размер
1.txt		01.10.2022 12:34	Текстовый докум...	0 КБ
2.txt		01.10.2022 12:34	Текстовый докум...	0 КБ
3.txt		01.10.2022 12:34	Текстовый докум...	0 КБ
L_3.txt		01.10.2022 12:32	Текстовый докум...	1 КБ

Рисунок 3.2 – Создание файлов

```
MINGW64:/c/Users/УчебНа/Desktop/СКФУ/2_3 семестр/Основы Программ...
Программной Инженерии/Git/TheFirstLab (main)
$ git add L_3/1.txt

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы
Программной Инженерии/Git/TheFirstLab (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   L_3/1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    L_3/2.txt
    L_3/3.txt

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы
Программной Инженерии/Git/TheFirstLab (main)
$ git commit -m "add 1.txt file"
[main 094da1d] add 1.txt file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 L_3/1.txt

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы
Программной Инженерии/Git/TheFirstLab (main)
$
```

Рисунок 3.3 – Индексация и коммит первого файла

```
УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы
Программной Инженерии/Git/TheFirstLab (main)
$ git add L_3/2.txt

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab
(main)
$ git add L_3/3.txt

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab
(main)
$
```

Рисунок 3.4 – Индексация второго и третьего файлов

```
УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab
(main)
$ git commit --amend -m "add 2.txt and 3.txt"
[main c9fb955] add 2.txt and 3.txt
Date: Sat Oct 1 12:40:05 2022 +0300
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 L_3/1.txt
create mode 100644 L_3/2.txt
create mode 100644 L_3/3.txt

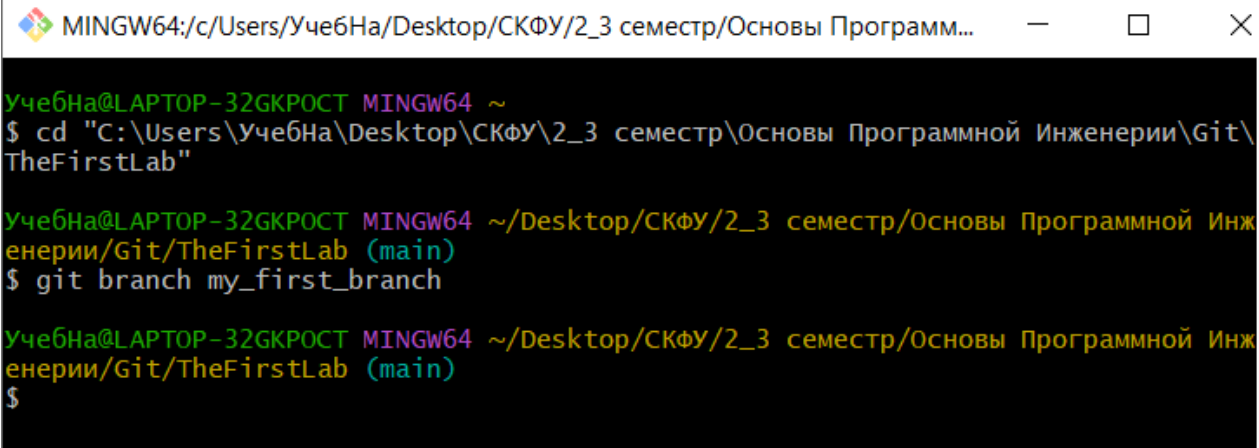
УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab
(main)
$ git log -p -1
commit c9fb955201fa429c06bfff8c31e83217645ac318 (HEAD -> main)
Author: KuvshinChick <kuvshinirina75@gmail.com>
Date: Sat Oct 1 12:40:05 2022 +0300

    add 2.txt and 3.txt

diff --git a/L_3/1.txt b/L_3/1.txt
new file mode 100644
index 0000000..e69de29
diff --git a/L_3/2.txt b/L_3/2.txt
new file mode 100644
index 0000000..e69de29
diff --git a/L_3/3.txt b/L_3/3.txt
```

Рисунок 3.5 – Перезапись коммита

7. Создать новую ветку my\_first\_branch.
8. Перейти на ветку и создать новый файл in\_branch.txt, закоммитить изменения.
9. Вернуться на ветку master.

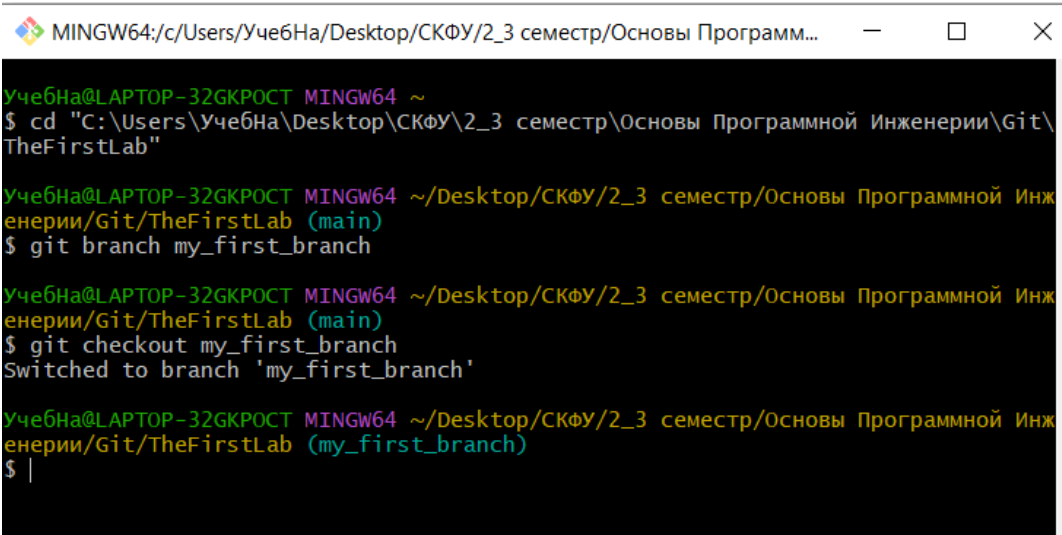


```
MINGW64:/c/Users/УчебНа/Desktop/СКФУ/2_3 семестр/Основы Программ...
УчебНа@LAPTOP-32GKPOCT MINGW64 ~
$ cd "C:\Users\УчебНа\Desktop\СКФУ\2_3 семестр\Основы Программной Инженерии\Git\TheFirstLab"

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab (main)
$ git branch my_first_branch

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab (main)
$
```

Рисунок 3.6 – Создание новой ветки



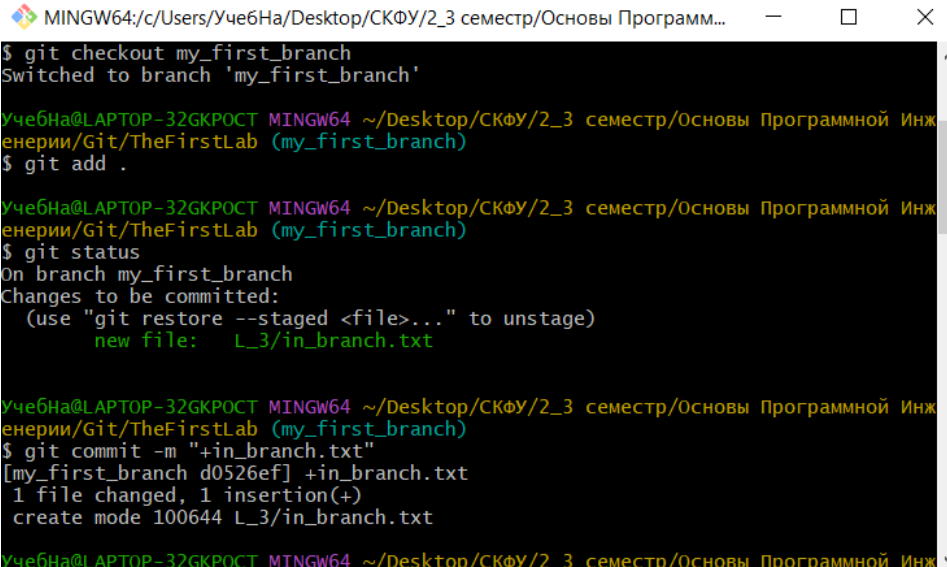
```
MINGW64:/c/Users/УчебНа/Desktop/СКФУ/2_3 семестр/Основы Программ...
УчебНа@LAPTOP-32GKPOCT MINGW64 ~
$ cd "C:\Users\УчебНа\Desktop\СКФУ\2_3 семестр\Основы Программной Инженерии\Git\TheFirstLab"

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab (main)
$ git branch my_first_branch

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab (main)
$ git checkout my_first_branch
Switched to branch 'my_first_branch'

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab (my_first_branch)
$ |
```

Рисунок 3.7 – Переход на новую ветку



```
MINGW64:/c/Users/УчебНа/Desktop/СКФУ/2_3 семестр/Основы Программ...
$ git checkout my_first_branch
Switched to branch 'my_first_branch'

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab (my_first_branch)
$ git add .

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab (my_first_branch)
$ git status
On branch my_first_branch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   L_3/in_branch.txt

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab (my_first_branch)
$ git commit -m "+in_branch.txt"
[my_first_branch d0526ef] +in_branch.txt
1 file changed, 1 insertion(+)
create mode 100644 L_3/in_branch.txt

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab (my_first_branch)
$
```

Рисунок 3.8 – Коммит файла в новую ветку

10. Создать и сразу перейти на ветку new\_branch.
11. Сделать изменения в файле 1.txt, добавить строчку “new row in the 1.txt file”, закоммитить изменения.
12. Перейти на ветку master и слить ветки master и my\_first\_branch, после чего слить ветки master и new\_branch.
13. Удалить ветки my\_first\_branch и new\_branch.

```
Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инж  
нерии/Git/TheFirstLab (my_first_branch)  
$ git checkout -b new_branch  
Switched to a new branch 'new_branch'
```

Рисунок 3.9 – Создание и переход на новую ветку

```
Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инж  
нерии/Git/TheFirstLab (new_branch)  
$ git add .  
  
Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инж  
нерии/Git/TheFirstLab (new_branch)  
$ git commit -m "+new row in the 1.txt|"  
[new_branch 51e0ddb] +new row in the 1.txt  
1 file changed, 1 insertion(+)  
  
Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инж  
нерии/Git/TheFirstLab (new_branch)  
$
```

Рисунок 3.10 – Создание изменений файла и его коммит

```
MINGW64; c:/Users/Учебна/Desktop/СКФУ/2_3 семестр/Основы Программ...  
Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы программной инж  
нерии/Git/TheFirstLab (new_branch)  
$ git checkout main  
Switched to branch 'main'  
Your branch is ahead of 'origin/main' by 1 commit.  
(use "git push" to publish your local commits)  
  
Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы программной инж  
нерии/Git/TheFirstLab (main)  
$ git merge my_first_branch  
Updating c9fb955..d0526ef  
Fast-forward  
 L_3/in_branch.txt | 1 +  
1 file changed, 1 insertion(+)  
create mode 100644 L_3/in_branch.txt  
  
Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы программной инж  
нерии/Git/TheFirstLab (main)  
$ git merge new_branch  
Updating d0526ef..51e0ddb  
Fast-forward  
 L_3/1.txt | 1 +  
1 file changed, 1 insertion(+)
```

Рисунок 3.11 – Слияние веток

```

учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/скфу/2_3 семестр/Основы программной инж
енерии/Git/TheFirstLab (main)
$ git branch -d my_first_branch
Deleted branch my_first_branch (was d0526ef).

учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/скфу/2_3 семестр/Основы программной инженерии/Git
/TheFirstLab (main)
$ git branch -d new_branch
Deleted branch new_branch (was 51e0ddb).

учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/скфу/2_3 семестр/Основы программной инженерии/Git
/TheFirstLab (main)
$

```

Рисунок 3.12 – Удаление веток

14. Создать ветки branch\_1 и branch\_2.
15. Перейти на ветку branch\_1 и изменить файл 1.txt, удалить все содержимое и добавить текст “fix in the 1.txt”, изменить файл 3.txt, удалить все содержимое и добавить текст “fix in the 3.txt”, закоммитить изменения.
16. Перейти на ветку branch\_2 и также изменить файл 1.txt, удалить все содержимое и добавить текст “My fix in the 1.txt”, изменить файл 3.txt, удалить все содержимое и добавить текст “My fix in the 3.txt”, закоммитить изменения.
17. Слить изменения ветки branch\_2 в ветку branch\_1.

```

учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/скфу/2_3 семестр/Основы программной инженерии/Git
/TheFirstLab (main)
$ git branch branch_2

учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/скфу/2_3 семестр/Основы программной инженерии/Git
/TheFirstLab (main)
$ git checkout -b branch_1
Switched to a new branch 'branch_1'

учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/скфу/2_3 семестр/Основы программной инженерии/Git
/TheFirstLab (branch_1)
$ |

```

Рисунок 3.13 – Создание веток branch\_1 и branch\_2

```

учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/скфу/2_3 семестр/Основы программной инженерии/Git
/TheFirstLab (branch_1)
$ git status
On branch branch_1
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:   L_3/1.txt
        modified:   L_3/3.txt

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

учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/скфу/2_3 семестр/Основы программной инженерии/Git
/TheFirstLab (branch_1)
$ git add .

учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/скфу/2_3 семестр/Основы программной инженерии/Git
/TheFirstLab (branch_1)
$ git commit -m "+fix in the files"
[branch_1 d7c6320] +fix in the files
2 files changed, 2 insertions(+), 1 deletion(-)

```

Рисунок 3.14 – Коммит измененных файлов в ветке branch\_1

```

учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы программной инженерии/Git
/TheFirstLab (branch_2)
$ git status
on branch branch_2
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:   L_3/1.txt
        modified:   L_3/3.txt

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

учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы программной инженерии/Git
/TheFirstLab (branch_2)
$ git add .

учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы программной инженерии/Git
/TheFirstLab (branch_2)
$ git commit -m "My fix in the files"
[branch_2 ac4d328] My fix in the files
2 files changed, 2 insertions(+), 1 deletion(-)

```

Рисунок 3.15 – Коммит измененных файлов в ветке branch\_2

```

учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы программной инженерии/Git
/TheFirstLab (branch_2)
$ git merge branch_1
Auto-merging L_3/1.txt
CONFLICT (content): Merge conflict in L_3/1.txt
Auto-merging L_3/3.txt
CONFLICT (content): Merge conflict in L_3/3.txt
Automatic merge failed; fix conflicts and then commit the result.

```

Рисунок 3.16 – Попытка слияния

18. Решить конфликт файла 1.txt в ручном режиме, а конфликт 3.txt используя команду git mergetool с помощью одной из доступных утилит, например Meld.  
(i - нажимаем в самом файле для редактирования текста (если нужно) ctrl+c - завершаем редактирование (если начали через i) shift+z, shift+z - ДВА РАЗА сохраняем и выходим из редактора.)



```

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инж
енерии/Git/TheFirstLab/L_3 (branch_2|MERGING)
$ git mergetool

This message is displayed because 'merge.tool' is not configured.
See 'git mergetool --tool-help' or 'git help config' for more details.
'git mergetool' will now attempt to use one of the following tools:
opendiff kdiff3 tkdiff xxdiff meld tortoisemerge gvimdiff diffmerge ecme
rge p4merge araxis bc codecompare smerge emerge vimdiff nvimdiff
Merging:
L_3/1.txt
L_3/3.txt

Normal merge conflict for 'L_3/1.txt':
{local}: modified file
{remote}: modified file
Hit return to start merge resolution tool (vimdiff): meld
4 files to edit

Normal merge conflict for 'L_3/3.txt':
{local}: modified file
{remote}: modified file
Hit return to start merge resolution tool (vimdiff):
4 files to edit

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инж
енерии/Git/TheFirstLab/L_3 (branch_2|MERGING)
$ git merge branch_1
fatal: You have not concluded your merge (MERGE_HEAD exists).
Please, commit your changes before you merge.

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инж
енерии/Git/TheFirstLab/L_3 (branch_2|MERGING)
$ commit -m "try_to_merge"
bash: commit: command not found

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инж
енерии/Git/TheFirstLab/L_3 (branch_2|MERGING)

```

```

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инж
енерии/Git/TheFirstLab/L_3 (branch_2|MERGING)
$ commit -m "try_to_merge"
bash: commit: command not found

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инж
енерии/Git/TheFirstLab/L_3 (branch_2|MERGING)
$ git commit -m "try-to-merge"
[branch_2 b1dc08a] try-to-merge

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инж
енерии/Git/TheFirstLab/L_3 (branch_2)
$ git merge branch_1
Already up to date.

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инж
енерии/Git/TheFirstLab/L_3 (branch_2)
$ git branch
  branch_1
* branch_2
  main

```

Рисунок 3.17 – Использование mergetool

```

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/The
FirstLab/L_3 (branch_2)
$ git branch
  branch_1
* branch_2
  main

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/The
FirstLab/L_3 (branch_2)
$ git branch -d branch_1
Deleted branch branch_1 (was d7c6320).

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/The
FirstLab/L_3 (branch_2)
$

```

Рисунок 3.18 – Удаление ветки

19. Отправить ветку branch\_2 на GitHub.



```

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/The
FirstLab/L_3 (branch_2)
$ git push origin branch_2
Enumerating objects: 29, done.
Counting objects: 100% (29/29), done.
Delta compression using up to 8 threads
Compressing objects: 100% (18/18), done.
Writing objects: 100% (27/27), 2.16 KiB | 737.00 KiB/s, done.
Total 27 (delta 10), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (10/10), completed with 1 local object.
remote:
remote: Create a pull request for 'branch_2' on GitHub by visiting:
remote:   https://github.com/KuvshinChick/TheFirstLab/pull/new/branch_2
remote:
To https://github.com/KuvshinChick/TheFirstLab.git
 * [new branch]      branch_2 -> branch_2

```

Рисунок 3.19 – Отправка ветки в удаленный реп

20. Создать средствами GitHub удаленную ветку branch\_3.
21. Создать в локальном репозитории ветку отслеживания удаленной ветки branch\_3.
22. Перейти на ветку branch\_3 и добавить файл файл 2.txt строку "the final fantasy in the 4.txt file".
23. Выполнить перемещение ветки master на ветку branch\_2.
24. Отправить изменения веток master и branch\_2 на GitHub.

Create a branch

Branch name

branch\_3

Branch source

branch\_2

Beta Share feedback Create branch

Рисунок 3.20 – Создание ветки branch\_3

```

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы программной инженерии/Git
/TheFirstLab/L_3 (branch_2)
$ git fetch origin
From https://github.com/kuvshinchick/TheFirstLab
 * [new branch]      branch_3 -> origin/branch_3

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы программной инженерии/Git
/TheFirstLab/L_3 (branch_2)
$ git branch
* branch_2
  main

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы программной инженерии/Git
/TheFirstLab/L_3 (branch_2)
$ git checkout -b branch_3 origin/branch_3
Switched to a new branch 'branch_3'
branch 'branch_3' set up to track 'origin/branch_3'.

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы программной инженерии/Git
/TheFirstLab/L_3 (branch_3)
$ git branch
* branch_3
  branch_2
  main

```

Рисунок 3.21 – Создание ветки отслеживания branch\_3

```

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirst
Lab/L_3 (branch_2|REBASE 2/2)
$ git rebase --continue
hint: Waiting for your editor to close the file... unix2dos: converting file C:/Users/УчебНа/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab/.git/COMMIT_EDITMSG to DOS format..
.
dos2unix: converting file C:/Users/УчебНа/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirstLab/.git/COMMIT_EDITMSG to Unix format...
[detached HEAD cbd69d7] +fix in the files
2 files changed, 4 insertions(+), 2 deletions(-)
Successfully rebased and updated refs/heads/branch_2.

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirst
Lab/L_3 (branch_2)
$

```

Рисунок 3.22 – Перемещение ветки master на ветку branch\_2

```

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirst
Lab/L_3 (branch_2|REBASE 2/2)
$ git push origin main
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/KuvshinChick/TheFirstLab.git
   df1d109..51e0ddb  main -> main

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirst
Lab/L_3 (branch_2|REBASE 2/2)
$ git push origin branch_2
Everything up-to-date

УчебНа@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инженерии/Git/TheFirst
Lab/L_3 (branch_2|REBASE 2/2)
$

```

Рисунок 3.23 – Отправка веток

## Вопросы:

### 1. Что такое ветка?

Ветка в Git'e — это просто легковесный подвижный указатель на один из КОММИТОВ.

### 2. Что такое HEAD?

HEAD — это указатель, задача которого ссылаться на определенный коммит в репозитории.

HEAD — это указатель на коммит в вашем репозитории, который станет родителем следующего коммита.

### 3. Способы создания веток.

- Командой `git branch`
- > `git branch testing`
- Командой `git checkout -b`
- > `git checkout -b iss53`

### 4. Как узнать текущую ветку?

```
Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инж
нерии/Git/TheFirstLab/L_3 (branch_2)
$ git branch
* branch_2
  branch_3
  main
```

## 5. Как переключаться между ветками?

\$ git checkout <Имя ветки>

```
Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инж
нерии/Git/TheFirstLab/L_3 (branch_2)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Учебна@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_3 семестр/Основы Программной Инж
нерии/Git/TheFirstLab/L_3 (main)
$ git branch
  branch_2
  branch_3
* main
```

## 6. Что такое удаленная ветка?

Удалённые ветки — это ссылки на состояние веток в удаленных репозиториях.

## 7. Что такое ветка отслеживания?

Ветки слежения — это ссылки на определённое состояние удалённых веток.

## 8. Как создать ветку отслеживания?

Для синхронизации ваших изменений с удаленным сервером выполните команду git fetch (в нашем случае git fetch origin ).

> git fetch origin

> git checkout -b <Имя локальной ветки> origin/<Имя удаленной ветки>

## 9. Как отправить изменения из локальной ветки в удаленную ветку?

> git push origin <Имя ветки>

## 10. В чем отличие команд git fetch и git pull ?

Для синхронизации ваших изменений с удаленным сервером выполните команду git fetch (в нашем случае git fetch origin ). Эта команда определяет какому серверу соответствует “origin”(в нашем случае это git.ourcompany.com), извлекает оттуда данные, которых у вас ещё нет, и обновляет локальную базу данных, сдвигая указатель origin/master на новую позицию.

Git fetch это команда, которая говорит: "обновите мою локальную копию удаленного репозитория". git pull говорит: "перенесите изменения в удаленный репозиторий туда, где я храню свой собственный код".

## 11. Как удалить локальную и удаленную ветки?

Удаленная ветка

```
git push origin --delete <Имя ветки>
```

Локальная ветка

```
git branch -d <Имя ветки>
```

12. Изучить модель ветвления git-flow (использовать материалы статей <https://www.atlassian.com/ru/git/tutorials/comparing-workflows/gitflow-workflow>, <https://habr.com/ru/post/106912/>). Какие основные типы веток присутствуют в модели git-flow? Как организована работа с ветками в модели git-flow? В чем недостатки git-flow?

Git-flow — альтернативная модель ветвления Git, в которой используются функциональные ветки и несколько основных веток.

Под каждую новую функцию нужно выделить собственную ветку, которую можно отправить в центральный репозиторий для создания резервной копии или совместной работы команды. Ветки feature создаются не на основе main, а на основе develop. Когда работа над функцией завершается, соответствующая ветка сливается с веткой develop. Функции не следует отправлять напрямую в ветку main.

Последовательность действий при работе по модели Gitflow:

1. Из ветки main создается ветка develop.
2. Из ветки develop создается ветка release.
3. Из ветки develop создаются ветки feature.
4. Когда работа над веткой feature завершается, она сливается в ветку develop.
5. Когда работа над веткой release завершается, она сливается с ветками develop и main.
6. Если в ветке main обнаруживается проблема, из main создается ветка hotfix.
7. Когда работа над веткой hotfix завершается, она сливается с ветками develop и main.

Первая проблема: авторам приходится использовать ветку develop вместо master, поскольку master зарезервирован для кода, который отправляется в продакшен. Существует сложившийся обычай называть рабочую ветвь по умолчанию master, и делать ответвления и слияния с ней. Большинство инструментов по умолчанию используют это название для

основной ветки и по умолчанию выводят именно ее, и бывает неудобно постоянно переключаться вручную на другую ветку.

Вторая проблема процесса git flow – сложности, возникающие из-за веток для патчей и для релиза. Подобная структура может подойти некоторым организациям, но для абсолютного большинства она просто убийственно излишня. На сегодняшний день большинство компаний практикуют непрерывное развертывание (continuous delivery), что подразумевает, что основная ветвь по умолчанию может быть задеплоена (deploy). А значит, можно избежать использования веток для релиза и патчей, и всех связанных с ними хлопот, например, обратного слияния из веток релизов.

13. На прошлой лабораторной работе было задание выбрать одно из программных средств с GUI для работы с Git. Необходимо в рамках этого вопроса привести описание инструментов для работы с ветками Git, предоставляемых этим средством.

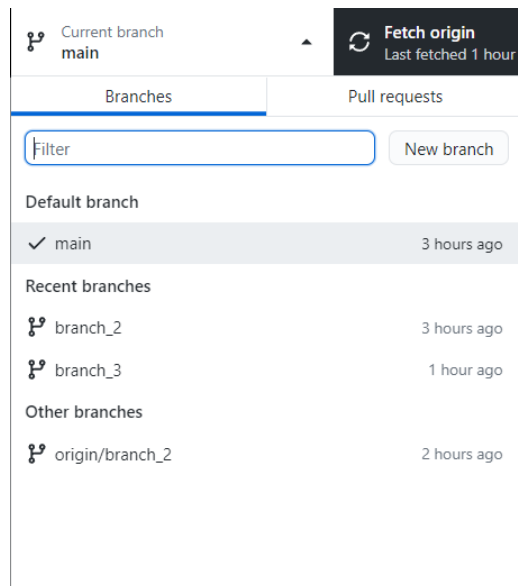


Рисунок – меню веток

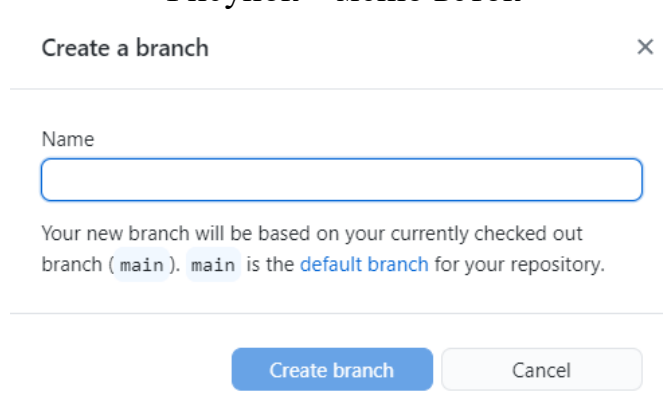


Рисунок – создание ветки

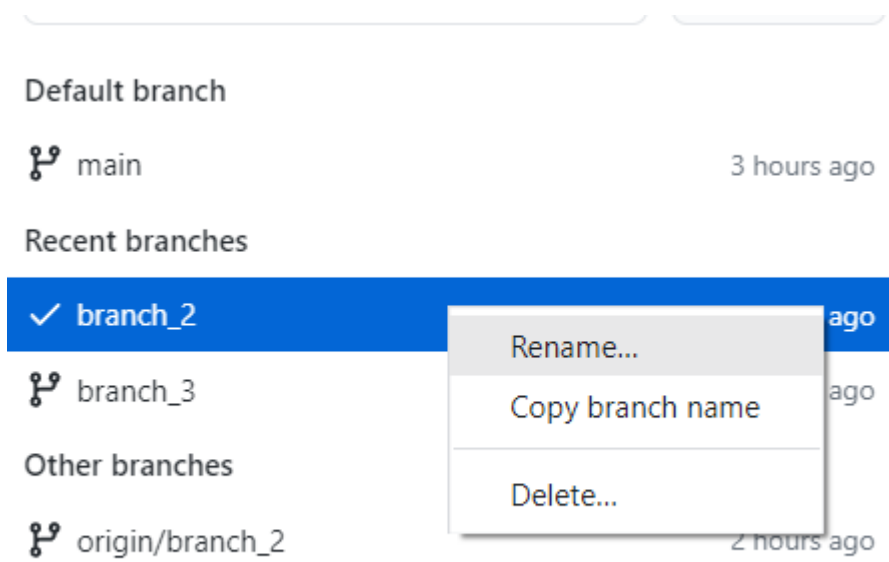


Рисунок – переименовать ветку

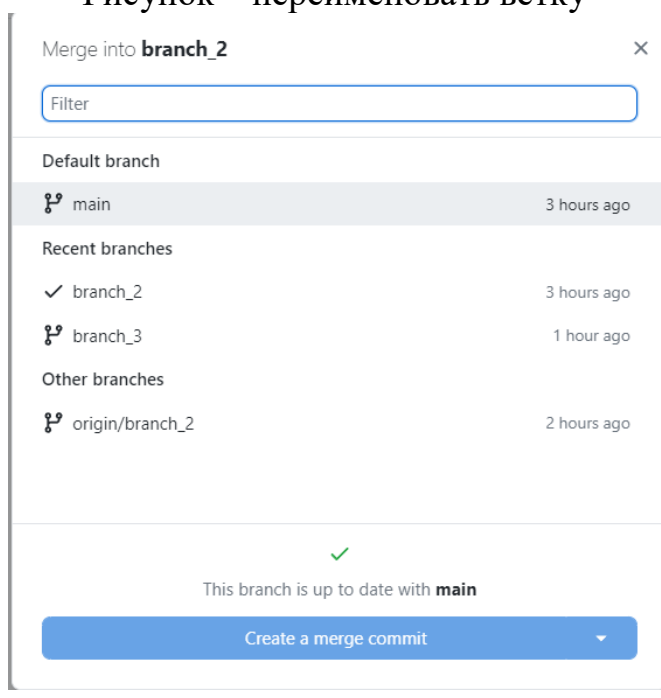


Рисунок – merge ветки