

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

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №1.3**  
**дисциплины**  
**«Основы кроссплатформенного программирования»**

Выполнил:  
Арзютов Иван  
Владиславович 1 курс,  
группа ИТС-б-о-22-1,  
11.03.02 «Инфокоммуникационные  
технологии и системы связи»,  
направленность (профиль)  
«Инфокоммуникационные системы и  
сети», заочная форма обучения

---

(подпись)

Проверил:  
Воронкин Р. А., доцент кафедры  
инфокоммуникаций

---

(подпись)

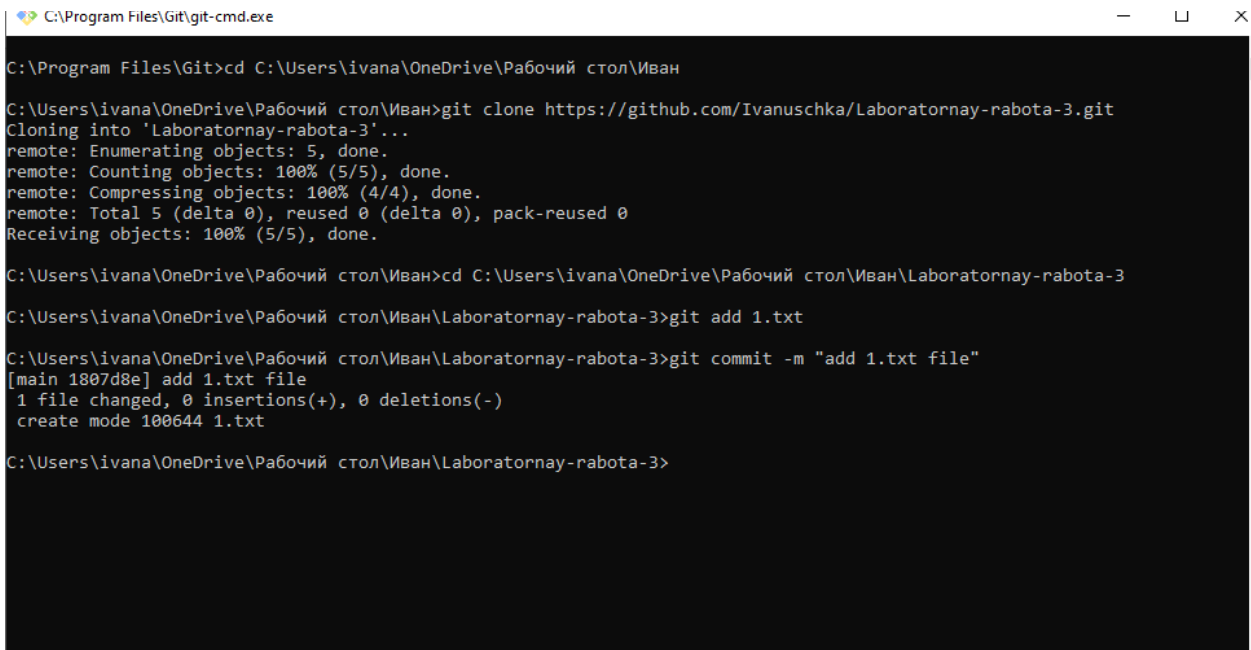
Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2023 г.

**Тема:** Основы ветвления Git.

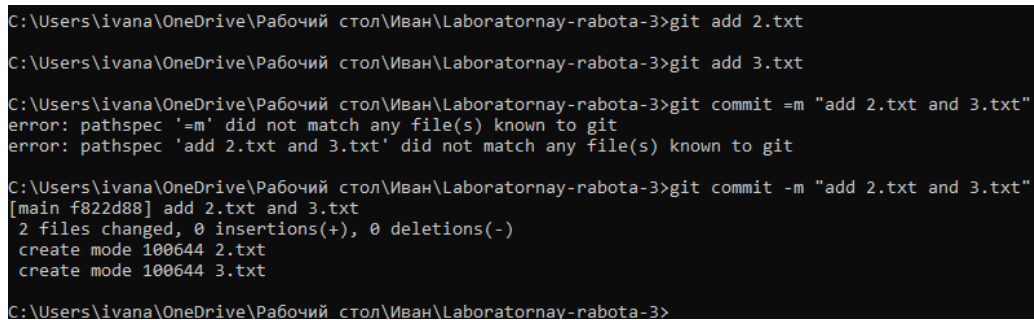
**Цель:** Исследование базовых возможностей по работе с

локальными и удаленными ветками Git.



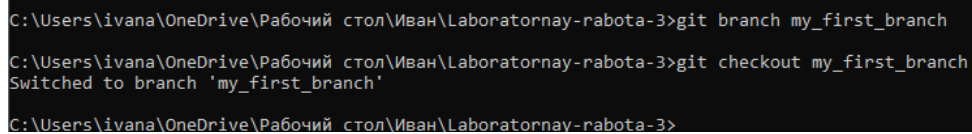
```
C:\Program Files\Git\git-cmd.exe
C:\Program Files\Git>cd C:\Users\ivana\OneDrive\Рабочий стол\Иван
C:\Users\ivana\OneDrive\Рабочий стол\Иван>git clone https://github.com/Ivanuschka/Laboratornay-rabota-3.git
Cloning into 'Laboratornay-rabota-3'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
C:\Users\ivana\OneDrive\Рабочий стол\Иван>cd C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git add 1.txt
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git commit -m "add 1.txt file"
[main 1807d8e] add 1.txt file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>
```

Рисунок 1. Добавил, проиндексировал и закомитил файл 1.txt



```
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git add 2.txt
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git add 3.txt
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git commit -m "add 2.txt and 3.txt"
error: pathspec '=m' did not match any file(s) known to git
error: pathspec 'add 2.txt and 3.txt' did not match any file(s) known to git
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git commit -m "add 2.txt and 3.txt"
[main f822d88] add 2.txt and 3.txt
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 2.txt
create mode 100644 3.txt
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>
```

Рисунок 2. Также добавил 2 файла



```
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git branch my_first_branch
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git checkout my_first_branch
Switched to branch 'my_first_branch'
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>
```

Рисунок 3. Создал и перешёл на новую ветку

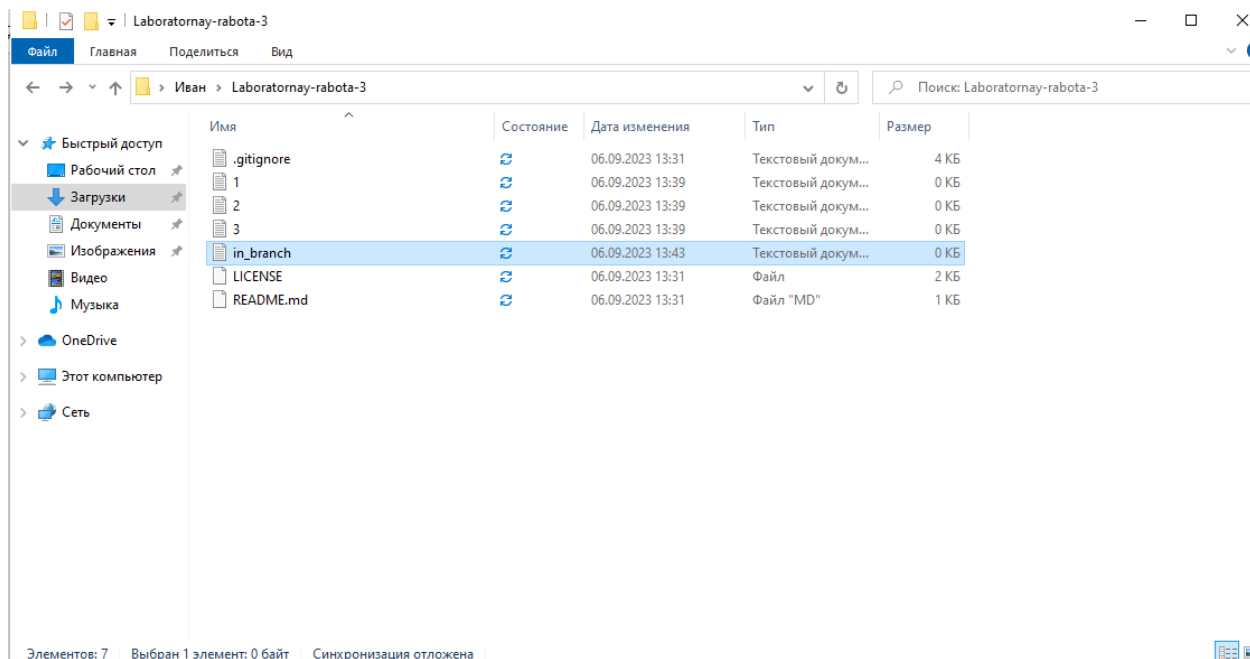


Рисунок 4. Создал файл в новой ветке

```
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git checkout -b new_branch
Switched to a new branch 'new_branch'

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>
```

Рисунок 5. Создал и сразу перешёл на новую ветку

```
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git commit -m "new_branch"
[new_branch ec66c8a] new_branch
1 file changed, 1 insertion(+)

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>
```

Рисунок 6. Внёс изменения в файл на ветке new\_branch

```
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git merge my_first_branch
Updating f822d88..a98089c
Fast-forward
 in_branch.txt | 0
1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 in_branch.txt

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git merge new_branch
Merge made by the 'ort' strategy.
 1.txt | 1 +
1 file changed, 1 insertion(+)

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>
```

Рисунок 7. Слил ветки с веткой main

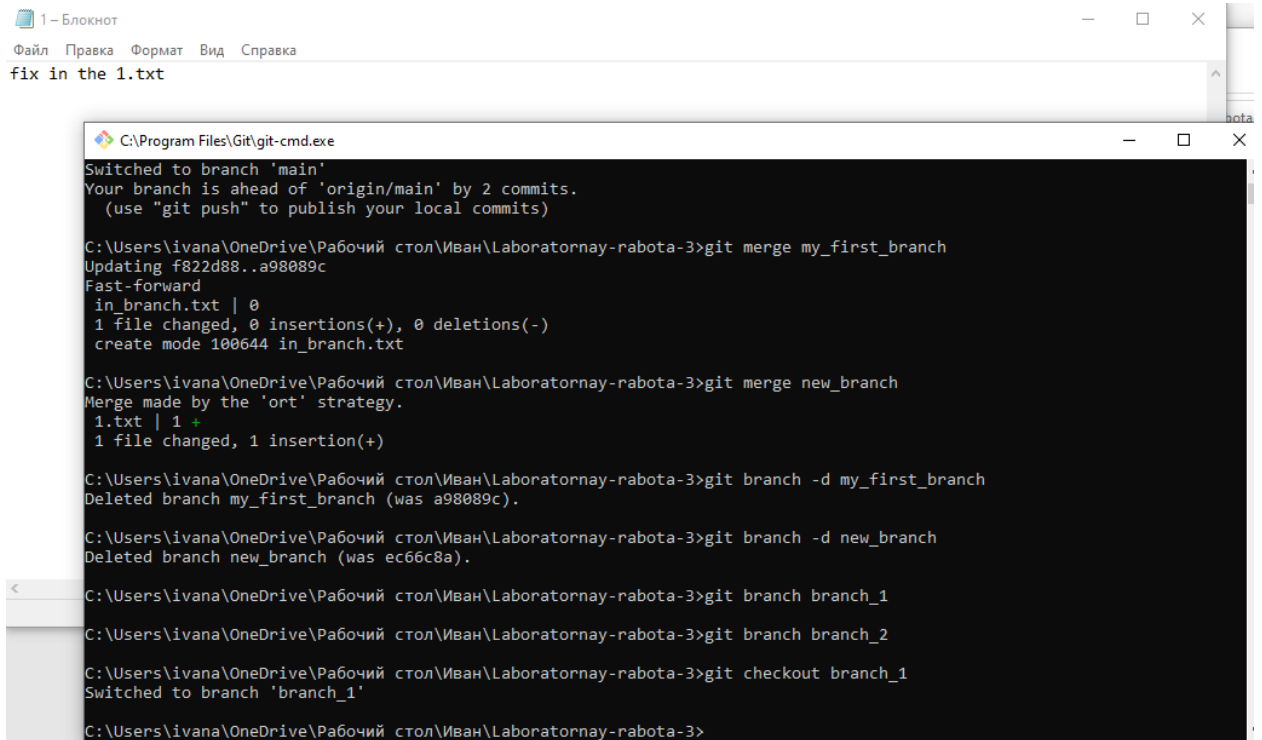
```
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git merge new_branch
Merge made by the 'ort' strategy.
 1.txt | 1 +
1 file changed, 1 insertion(+)

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git branch -d my_first_branch
Deleted branch my_first_branch (was a98089c).

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git branch -d new_branch
Deleted branch new_branch (was ec66c8a).

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>
```

Рисунок 8. Удалил ветки из репозитория



```
1 - Блокнот
Файл Правка Формат Вид Справка
fix in the 1.txt

C:\Program Files\Git\git-cmd.exe
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 2 commits.
(use "git push" to publish your local commits)

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git merge my_first_branch
Updating f822d88..a98089c
Fast-forward
 in_branch.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 in_branch.txt

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git merge new_branch
Merge made by the 'ort' strategy.
 1.txt | 1 +
 1 file changed, 1 insertion(+)

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git branch -d my_first_branch
Deleted branch my_first_branch (was a98089c).

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git branch -d new_branch
Deleted branch new_branch (was ec66c8a).

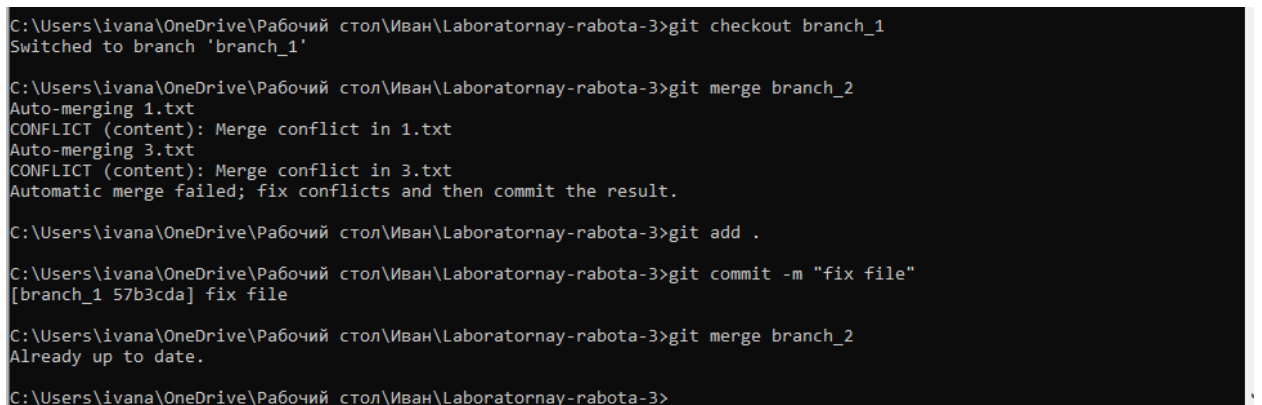
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git branch branch_1

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git branch branch_2

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git checkout branch_1
Switched to branch 'branch_1'

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>
```

Рисунок 9. Внёс изменения в файл 1.txt



```
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git checkout branch_1
Switched to branch 'branch_1'

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git merge branch_2
Auto-merging 1.txt
CONFLICT (content): Merge conflict in 1.txt
Auto-merging 3.txt
CONFLICT (content): Merge conflict in 3.txt
Automatic merge failed; fix conflicts and then commit the result.

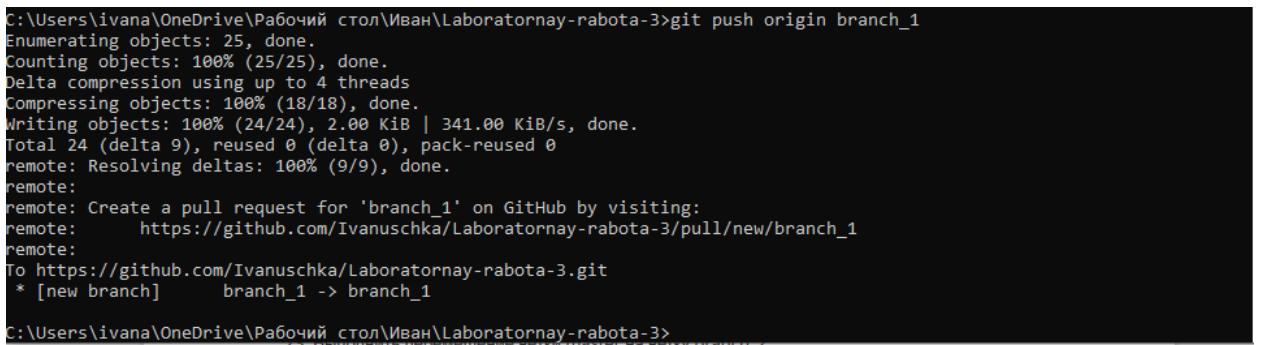
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git add .

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git commit -m "fix file"
[branch_1 57b3cda] fix file

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git merge branch_2
Already up to date.

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>
```

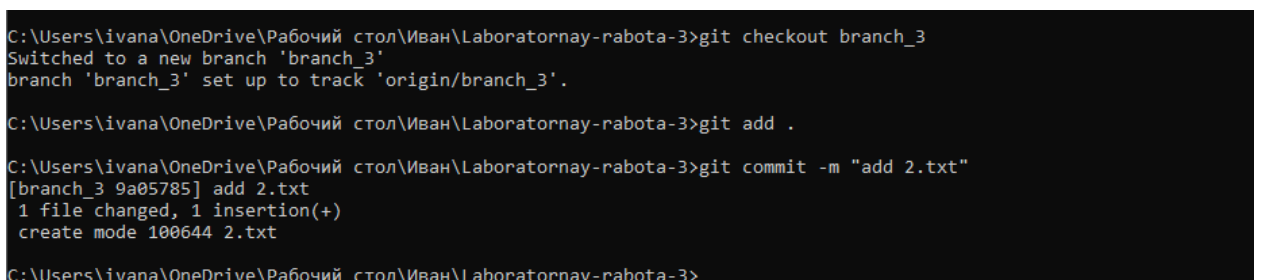
Рисунок 10. Решил конфликт слияния и соединил ветки branch\_1 и branch\_2



```
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git push origin branch_1
Enumerating objects: 25, done.
Counting objects: 100% (25/25), done.
Delta compression using up to 4 threads
Compressing objects: 100% (18/18), done.
Writing objects: 100% (24/24), 2.00 KiB | 341.00 KiB/s, done.
Total 24 (delta 9), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (9/9), done.
remote:
remote: Create a pull request for 'branch_1' on GitHub by visiting:
remote:   https://github.com/Ivanuschka/Laboratornay-rabota-3/pull/new/branch_1
remote:
To https://github.com/Ivanuschka/Laboratornay-rabota-3.git
 * [new branch]      branch_1 -> branch_1

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>
```

Рисунок 11. Отправил ветку на репозиторий



```
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git checkout branch_3
Switched to a new branch 'branch_3'
branch 'branch_3' set up to track 'origin/branch_3'.

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git add .

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git commit -m "add 2.txt"
[branch_3 9a05785] add 2.txt
 1 file changed, 1 insertion(+)
 create mode 100644 2.txt

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>
```

Рисунок 12. Создал удалённую ветку репозитория и добавил файл 2.txt

```
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git checkout branch_2
Switched to branch 'branch_2'

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git git checkout main
git: 'git' is not a git command. See 'git --help'.

The most similar command is
  init

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 5 commits.
(use "git push" to publish your local commits)

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git rebase branch_2
Successfully rebased and updated refs/heads/main.

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>
```

Рисунок 13. Переместил ветку main на ветку branch\_2

```
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Ivanuschka/Laboratornay-rabota-3.git
  7675be6..252e6f1  main -> main

C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornay-rabota-3>
```

Рисунок 14. Переместил изменения на удалённый репозиторий

### Контрольные вопросы:

1. Ветка в Git это параллельная версия проекта, которая может быть использована для разработки новых фич, исправления багов или проведения экспериментов, не влияющих на основную версию проекта.
2. HEAD в Git это указатель на текущую ветку или коммит, на котором находится пользователь или репозиторий.
3. Способы создания веток в Git:

- Создание новой ветки с помощью команды `git branch <имя_ветки>`
  - Создание и переключение на новую ветку с помощью команды `git checkout -b <имя_ветки>`
4. Текущую ветку можно узнать с помощью команды `git branch`, которая выведет список всех существующих веток, при этом звездочка (\*) будет указывать на текущую ветку.
  5. Для переключения между ветками можно использовать команду `git checkout <имя_ветки>`, где `<имя_ветки>` - это имя ветки, на которую нужно переключиться.
  6. Удаленная ветка в Git это ветка, которая находится на удаленном сервере (например, на Bitbucket), а не локально на компьютере пользователя.
  7. Ветка отслеживания (tracking branch) в Git это локальная ветка, которая связана с удаленной веткой и служит для отслеживания изменений в удаленной ветке и синхронизации их с локальной веткой.
  8. Создать ветку отслеживания можно с помощью команды `git checkout -b <имя_ветки> <имя_удаленной_ветки>`.
  9. Чтобы отправить изменения из локальной ветки в удаленную ветку, нужно выполнить команду `git push <имя_удаленного_репозитория> <имя_локальной_ветки>:<имя_удаленной_ветки>`.
  10. Основное отличие между командами `git fetch` и `git pull` заключается в том, что `git fetch` загружает изменения из удаленного репозитория, но не объединяет их с локальной веткой, в то время как `git pull` загружает изменения и сразу же объединяет их с локальной веткой.
  11. Удалить локальную ветку можно с помощью команды `git branch -d <имя_ветки>`, а удаленную ветку - с помощью команды `git push <имя_удаленного_репозитория> --delete <имя_удаленной_ветки>`.

12. Git-flow это модель ветвления в Git, которая предлагает определенный набор стандартов и практик, которые помогают организовать процесс совместной разработки. Она состоит из двух основных веток: master и develop, а также из дополнительных веток типа feature, release и hotfix.

13. Основные типы веток в git-flow:

- master - главная ветка, в которой находится последняя стабильная версия проекта;
- develop - ветка, в которой происходит разработка новых фич;
- feature - ветка, создаваемая для разработки новой функциональности;
- release - ветка, создаваемая для подготовки выпуска новой версии проекта;
- hotfix - ветка, создаваемая для исправления критических ошибок в готовой версии проекта.

14. Инструменты для работы с ветками Git, предоставляемые средствами с GUI могут включать в себя: возможность создания, удаления и переключения между ветками с помощью графического интерфейса, просмотр списка всех доступных веток, отслеживание изменений и синхронизация с удаленным репозиторием. Некоторые популярные GUI-клиенты для работы с Git включают в себя GitHub Desktop, GitKraken, Sourcetree и другие.