

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**
**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Институт цифрового развития

ОТЧЁТ

по лабораторной работе №5

Дисциплина: «Основы кроссплатформенного программирования»

Тема: “Исследование базовых возможностей по работе с локальными и удаленными ветками Git”

Выполнил студент группы

ИВТ-б-о-21-1

Уланбековна Айканыш

« » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил доцент

Кафедры инфокоммуникаций, старший
преподаватель

Воронкин Р.А.

(подпись)

Ставрополь 2022

Ход работы:

```
C:\Users\User\Desktop\lab5\lab5>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:   1.txt
        new file:   2.txt
        new file:   3.txt
```

Рисунок 1 – Создала файлы

```
C:\Users\User\Desktop\lab5\lab5>git commit -m "add 1.txt file"
On branch main
Your branch is ahead of 'origin/main' by 4 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

Рисунок 2 – Закоммитила 1 файл

```
C:\Users\User\Desktop\lab5\lab5>git commit --amend -m "add 2.txt and 3.txt"
[main 6a3eddc] add 2.txt and 3.txt
Date: Tue May 17 22:38:29 2022 +0300
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 3.txt
```

Рисунок 3 – Перезаписала последний коммит

```
C:\Users\User\Desktop\lab5\lab5>git branch my_first_branch

C:\Users\User\Desktop\lab5\lab5>git checkout my_first_branch
Switched to branch 'my_first_branch'

C:\Users\User\Desktop\lab5\lab5>
```

Рисунок 4 – Создала новую ветку, затем перешла на неё

```

C:\Users\User\Desktop\lab5\lab5>git status
On branch my_first_branch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    branch.txt

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

C:\Users\User\Desktop\lab5\lab5>git add branch.txt

C:\Users\User\Desktop\lab5\lab5>git commit -m "branch.txt committed"
On branch my_first_branch
nothing to commit, working tree clean

```

Рисунок 5 – Создала файл и закоммитила изменения на ветке my_first_branch

```

C:\Users\User\Desktop\lab5\lab5>git checkout -b new_branch
Switched to a new branch 'new_branch'

C:\Users\User\Desktop\lab5\lab5>git status
On branch new_branch
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:   1.txt

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

C:\Users\User\Desktop\lab5\lab5>git add 1.txt.txt
fatal: pathspec '1.txt.txt' did not match any files

C:\Users\User\Desktop\lab5\lab5>git add 1.txt

C:\Users\User\Desktop\lab5\lab5>git commit -m "change 1.txt"
[new_branch 467b294] change 1.txt
1 file changed, 1 insertion(+)

```

Рисунок 6 – Создала ветку new_branch, сразу перешла на нее, изменила файл 1.txt и закоммитила изменения

```

C:\Users\User\Desktop\lab5\lab5>git merge my_first_branch
Updating 6a3eddc..0967346
Fast-forward
  branch.txt | 0
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 branch.txt

C:\Users\User\Desktop\lab5\lab5>git merge new_branch
Updating 0967346..467b294
Fast-forward
  1.txt | 1 +
  1 file changed, 1 insertion(+)

```

Рисунок 7 – Слила все ветки в одну

```
C:\Users\User\Desktop\lab5\lab5>git branch -d my_first_branch
Deleted branch my_first_branch (was 0967346).

C:\Users\User\Desktop\lab5\lab5>git branch -d new_branch
Deleted branch new_branch (was 467b294).
```

Рисунок 8 – Удалила ветки

```
C:\Users\User\Desktop\lab5\lab5>git branch branch_1
C:\Users\User\Desktop\lab5\lab5>git branch branch_2
C:\Users\User\Desktop\lab5\lab5>git branch
  branch_1
  branch_2
* main
```

Рисунок 9 – Создала новые ветки

```
C:\Users\User\Desktop\lab5\lab5>git checkout branch_1
Switched to branch 'branch_1'

C:\Users\User\Desktop\lab5\lab5>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:   1.txt
        modified:   3.txt

no changes added to commit (use "git add" and/or "git commit -a")
C:\Users\User\Desktop\lab5\lab5>git add 1.txt 3.txt
C:\Users\User\Desktop\lab5\lab5>git commit -m"change files"
[branch_1 2d169e5] change files
 2 files changed, 2 insertions(+), 1 deletion(-)
```

Рисунок 10 – Перешла на ветку branch_1, изменила файлы и закоммитила изменения

```

C:\Users\User\Desktop\lab5\lab5>git checkout branch_2
Switched to branch 'branch_2'

C:\Users\User\Desktop\lab5\lab5>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:   1.txt
        modified:   3.txt

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

C:\Users\User\Desktop\lab5\lab5>git add 1.txt 3.txt

C:\Users\User\Desktop\lab5\lab5>git commit -m "CHANGES"
[branch_2 571aa8e] CHANGES
 2 files changed, 2 insertions(+), 1 deletion(-)

```

Рисунок 11 – Перешла на ветку branch_2, изменила файлы и закоммитила изменения

```

D:\Git\Lab_5>git status
On branch branch_1
All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:
        modified:   1.txt.txt
        modified:   3.txt.txt

D:\Git\Lab_5>git commit
[branch_1 a733499] Merge branch 'branch_2' into branch_1

```

Рисунок 12 – Исправила конфликты и слила ветки

```

C:\Users\User\Desktop\lab5\lab5>git push origin branch_1
Enumerating objects: 43, done.
Counting objects: 100% (43/43), done.
Delta compression using up to 8 threads
Compressing objects: 100% (26/26), done.
Writing objects: 100% (42/42), 3.21 KiB | 274.00 KiB/s, done.
Total 42 (delta 14), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (14/14), done.
remote:
remote: Create a pull request for 'branch_1' on GitHub by visiting:
remote:   https://github.com/aikanyshkaukanbekova/lab5/pull/new/branch_1
remote:
To https://github.com/aikanyshkaukanbekova/lab5.git
 * [new branch]      branch_1 -> branch_1

```

Рисунок 13 – Отправила ветку branch_1 на GitHub

```
C:\Users\User\Desktop\lab5\lab5>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'.

C:\Users\User\Desktop\lab5\lab5>git branch
  branch_1
  branch_2
* branch_3
  main
```

Рисунок 14 – Создал ветку отслеживания в локальном репозитории

```
C:\Users\User\Desktop\lab5\lab5>git checkout branch_2
Switched to branch 'branch_2'

C:\Users\User\Desktop\lab5\lab5>git rebase main
Current branch branch_2 is up to date.
```

Рисунок 15 – Переместил ветку main на ветку branch_2

```
C:\Users\User\Desktop\lab5\lab5>git push --set-upstream origin branch_2
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'branch_2' on GitHub by visiting:
remote:   https://github.com/aikanyshkaukanbekova/lab5/pull/new/branch_2
remote:
To https://github.com/aikanyshkaukanbekova/lab5.git
 * [new branch]      branch_2 -> branch_2
branch 'branch_2' set up to track 'origin/branch_2'.
```

Рисунок 16 – Отправил изменения веток на GitHub

Ответы на контрольные вопросы:

1) Что такое ветка?

Это последовательность коммитов, отклоняющаяся от основной линии разработки.

2) Что такое HEAD?

Это указатель, задача которого ссылаться на определённый коммит в репозитории, если точнее: на коммит, который станет родителем для следующего коммита.

3) Способы создания веток?

Новую ветку можно создать командой `git branch <название_ветки>` или на удаленном репозитории `git hub`.

4) Как узнать текущую ветку?

При помощи команды `git branch`.

5) Как переключаться между ветками?

При помощи команды `git checkout <название_ветки>`.

6) Что такое удаленная ветка?

Это ветка, находящаяся на удаленном репозитории. Или ссылка на состояние ветки на удаленном репозитории.

7) Что такое ветка отслеживания?

Это ветка в локальном репозитории, которая напрямую связана с удаленной веткой на удаленном репозитории.

8) Как создать ветку отслеживания?

Командой `git checkout --track origin/<название_ветки>`.

9) Как отправить изменения из локальной ветки в удаленную ветку?

Командой `git push origin <название_ветки>`.

10) В чем отличие команд `git fetch` и `git pull`?

`Git pull` – это сочетание команд `git fetch` (получение изменений с удаленного репозитория) и `git merge` (объединение веток).

11) Как удалить локальную и удаленную ветки?

Используя команду `git branch -d <название_ветки>`. Для удаление удаленной ветки существует команда `git push origin -d <название_ветки>`.