

**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 ПРАКТИЧЕСКОЙ РАБОТЕ №1.3
дисциплины
«Основы кроссплатформенного программирования»**

Выполнила студентка группы
ИВТ-б-о-21-1
Яковлева Е.А. « » _____ 20__ г.
Подпись студента _____
Работа защищена
« » _____ 20__ г.

Проверил доцент
Кафедры инфокоммуникаций,
старший преподаватель
Воронкин Р.А.

(подпись)

Ставрополь, 2022 г.

Тема: Основы ветвления Git

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

Порядок выполнения работы:

Создала общедоступный репозиторий на GitHub, в котором использованы лицензия MIT. Клонировала репозиторий.

```
C:\Users\Elizaveta>cd/d C:\Users\Elizaveta\Desktop\git
C:\Users\Elizaveta\Desktop\git>git clone https://github.com/liz4simpson/lr1.3.git
Cloning into 'lr1.3'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
```

Рисунок 1. Клонирование репозитория

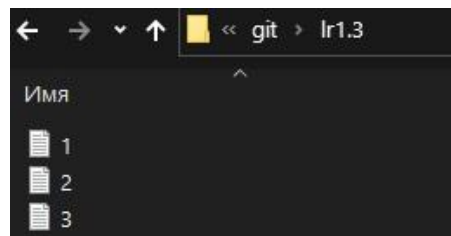


Рисунок 2. Создала файлы

```
C:\Users\Elizaveta\Desktop\git\lr1.3>git add 1.txt
C:\Users\Elizaveta\Desktop\git\lr1.3>git commit -m "add 1.txt file"
[main 38214cb] add 1.txt file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt
C:\Users\Elizaveta\Desktop\git\lr1.3>
```

Рисунок 3. Закоммитила 1 файл

```
C:\Users\Elizaveta\Desktop\git\lr1.3>git add .
C:\Users\Elizaveta\Desktop\git\lr1.3>git commit --amend -m "add 2.txt and 3.txt"
[main 93a46e6] add 2.txt and 3.txt
Date: Tue Sep 20 02:43:52 2022 +0300
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt
create mode 100644 2.txt
create mode 100644 3.txt
```

Рисунок 4. Перезаписала последний коммит

```
C:\Users\Elizaveta\Desktop\git\lr1.3>git branch my_first_branch
C:\Users\Elizaveta\Desktop\git\lr1.3>git checkout my_first_branch
Switched to branch 'my_first_branch'
```

Рисунок 5. Создала новую ветку, затем перешла на неё

```

C:\Users\Elizaveta\Desktop\git\lr1.3>git status
On branch my_first_branch
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        in_branch.txt

nothing added to commit but untracked files present (use "git add" to tra
C:\Users\Elizaveta\Desktop\git\lr1.3>git add in_branch.txt
C:\Users\Elizaveta\Desktop\git\lr1.3>git commit -m "in_branch.txt commite
[my_first_branch c56b61c] in_branch.txt committed
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 in_branch.txt

```

Рисунок 6. Создала файл и закоммитила изменения на ветке
my_first_branch

```

C:\Users\Elizaveta\Desktop\git\lr1.3>git add in_branch.txt
C:\Users\Elizaveta\Desktop\git\lr1.3>git commit -m "in_branch.txt committed"
[my_first_branch c56b61c] in_branch.txt committed
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 in_branch.txt
C:\Users\Elizaveta\Desktop\git\lr1.3>git checkout -b new_branch
Switched to a new branch 'new_branch'
C:\Users\Elizaveta\Desktop\git\lr1.3>git branch
  main
  my_first_branch
* new_branch

```

```

C:\Users\Elizaveta\Desktop\git\lr1.3>git branch
  main
  my_first_branch
* new_branch
C:\Users\Elizaveta\Desktop\git\lr1.3>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\Elizaveta\Desktop\git\lr1.3>git add 1.txt
C:\Users\Elizaveta\Desktop\git\lr1.3>git commit -m "change 1.txt"
[new_branch 5538461] change 1.txt
1 file changed, 1 insertion(+)

```

Рисунок 7. Создала ветку new_branch, сразу перешла на нее, изменила
файл 1.txt и закоммитила изменения

```

D:\Git\Lab_5>git merge my_first_branch
Updating 3f91ad4..daf722f
Fast-forward
  in_branch.txt | 0
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 in_branch.txt
D:\Git\Lab_5>git merge new_branch
Merge made by the 'ort' strategy.
  1.txt.txt | 1 +
  1 file changed, 1 insertion(+)

```

Рисунок 8. Слила все ветки в одну

```
D:\Git\Lab_5>git branch -d my_first_branch
Deleted branch my_first_branch (was daf722f).

D:\Git\Lab_5>git branch -d new_branch
Deleted branch new_branch (was 9483e1b).
```

Рисунок 9. Удалила ветки

```
D:\Git\Lab_5>git branch branch_1

D:\Git\Lab_5>git branch branch_2

D:\Git\Lab_5>git branch
  branch_1
  branch_2
* main
```

Рисунок 10. Создала новые ветки

```
D:\Git\Lab_5>git checkout branch_1
Switched to branch 'branch_1'

D:\Git\Lab_5>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.txt
        modified:   3.txt.txt

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

D:\Git\Lab_5>git add 1.txt.txt 3.txt.txt

D:\Git\Lab_5>git commit -m "Change files"
[branch_1 8903f91] Change files
 2 files changed, 2 insertions(+), 1 deletion(-)
```

Рисунок 11. Перешла на ветку branch_1, изменила файлы и
закоммитила изменения

```

D:\Git\Lab_5>git checkout branch_2
Switched to branch 'branch_2'

D:\Git\Lab_5>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.txt
        modified:   3.txt.txt

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

D:\Git\Lab_5>git add 1.txt.txt 3.txt.txt

D:\Git\Lab_5>git commit -m "CHAnges"
[branch_2 335c059] CHAnges
 2 files changed, 2 insertions(+), 1 deletion(-)

```

Рисунок 12. 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 13. Исправила конфликты вручную и слила ветки

```

D:\Git\Lab_5>git push origin branch_1
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 12 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (22/22), 1.76 KiB | 1.76 MiB/s, done.
Total 22 (delta 8), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (8/8), done.
remote:
remote: Create a pull request for 'branch_1' on GitHub by visiting:
remote:   https://github.com/Konstellation/Lab_5/pull/new/branch_1
remote:
To https://github.com/Konstellation/Lab_5.git
 * [new branch]      branch_1 -> branch_1

```

Рисунок 14. Отправила ветку branch_1 на GitHub


```
C:\Users\Elizaveta\Desktop\git\13>git checkout --track origin/branch_3
Switched to a new branch 'branch_3'
branch 'branch_3' set up to track 'origin/branch_3'.
```

Рисунок 15. Создала ветку отслеживания в локальной репозитории

```
C:\Users\Elizaveta\Desktop\git\13>git branch
branch_1
branch_2
* branch_3
main

C:\Users\Elizaveta\Desktop\git\13>git rebase branch_2
Successfully rebased and updated refs/heads/branch_3.
```

Рисунок 16. Переместила ветку main на ветку branch_2

```
C:\Users\Elizaveta\Desktop\git\13>git push --set-upstream origin branch_2
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 381 bytes | 15.00 KiB/s, done.
Total 4 (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/liz4simpson/13/pull/new/branch_2
remote:
To https://github.com/liz4simpson/13.git
 * [new branch]      branch_2 -> branch_2
branch 'branch_2' set up to track 'origin/branch_2'.

C:\Users\Elizaveta\Desktop\git\13>
C:\Users\Elizaveta\Desktop\git\13>git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 4 commits.
(use "git push" to publish your local commits)

C:\Users\Elizaveta\Desktop\git\13>git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/liz4simpson/13.git
 8712333..8752285  main -> main
```

Рисунок 17. Отправила изменения веток на GitHub

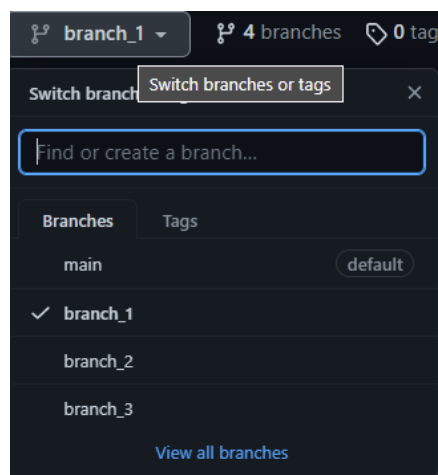


Рисунок 18. Зафиксировала изменения на удаленном репозитории

Вывод: в ходе лабораторной работы были изучены базовых возможности по работе с локальными и удаленными ветками Git.

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` 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`.

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/`.

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

Командой `git push origin`.

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`. Для удаление удаленной ветки существует команда `git push origin -d`.