

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**

Выполнила:
Кузнецова Алена Валерьевна
1 курс, группа ИВТ-б-о-21-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

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

1. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT.

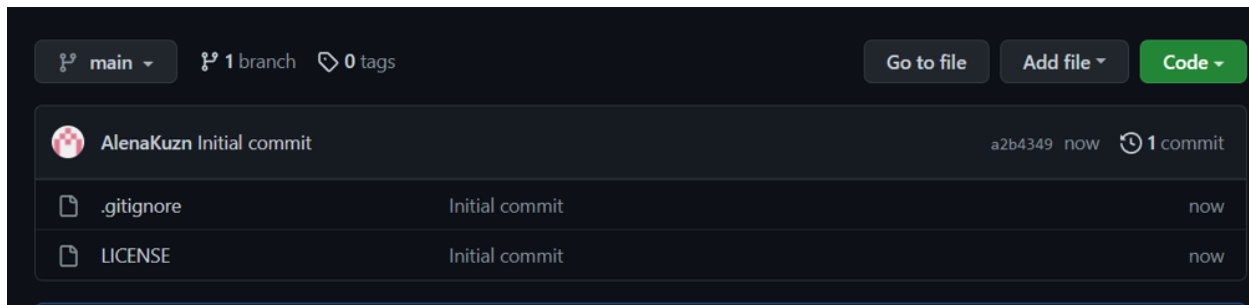


Рисунок 1 – Создание репозитория

2. Создать три файла: 1.txt, 2.txt, 3.txt.

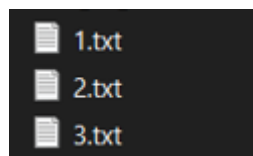


Рисунок 2 – Созданные файлы

3. Проиндексировать файлы и сделать коммит

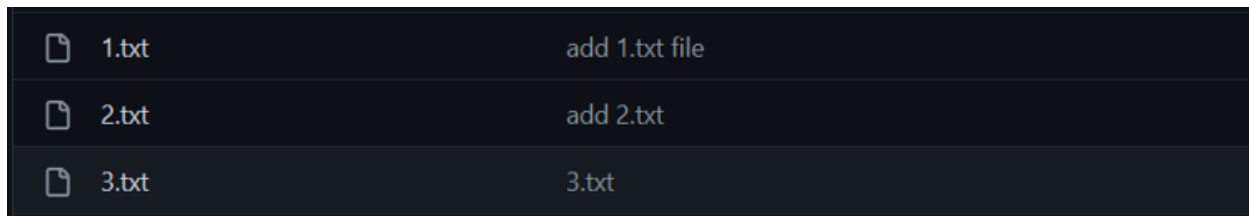


Рисунок 3 – Измененные файлы

4. Создать новую ветку my_first_branch.

```
C:\Users\akuzn\1.3>git branch my_first_branch  
  
C:\Users\akuzn\1.3>git checkout my_first_branch  
Switched to branch 'my_first_branch'
```

Рисунок 4 – Создание ветки

5. Перейти на ветку и создать новый файл in_branch.txt, закоммитить изменения.

```
C:\Users\akuzn\1.3>git add in_branch.txt

C:\Users\akuzn\1.3>git commit -m"in_br"
[my_first_branch dd97443] in_br
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 in_branch.txt

C:\Users\akuzn\1.3>git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 252 bytes | 63.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/AlenaKuzn/1.3.git
 60f8b12..dd97443 my_first_branch -> my_first_branch
```

Рисунок 5 – Отправка нового файла

6. Сделать изменения в файле 1.txt, добавить строчку “new row in the 1.txt file”, закоммитить изменения.

```
C:\Users\akuzn\1.3>git add 1.txt

C:\Users\akuzn\1.3>git commit -m"1"
[my_first_branch d94ea9c] 1
 1 file changed, 1 insertion(+)

C:\Users\akuzn\1.3>git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 299 bytes | 74.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/AlenaKuzn/1.3.git
 dd97443..d94ea9c my_first_branch -> my_first_branch
```

Рисунок 6 – Отправка изменений файла 1.txt

7. Удалить ветки my_first_branch и new_branch.

```
C:\Users\akuzn\1.3>git branch -d my_first_branch
Deleted branch my_first_branch (was d94ea9c).

C:\Users\akuzn\1.3>git branch -d new_branch
Deleted branch new_branch (was dd97443).
```

Рисунок 7 – Удаление веток

8. Создать ветки branch_1 и branch_2.

```
C:\Users\akuzn\1.3>git branch branch_1

C:\Users\akuzn\1.3>git branch branch_2
```

Рисунок 8 – Создание веток

9. Слить изменения ветки branch_2 в ветку branch_1.

```
C:\Users\akuzn\1.3>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.
```

Рисунок 9 – Слияние веток

10. Решить конфликт файла 1.txt в ручном режиме, а конфликт 3.txt используя команду git mergetool с помощью одной из доступных утилит, например Meld.

```
C:\Users\akuzn\1.3>git status
On branch branch_1
Your branch is up to date with 'origin/branch_1'.

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Changes to be committed:
  modified:   1.txt

Unmerged paths:
  (use "git add <file>..." to mark resolution)
  both modified:   3.txt
```

Рисунок 10 – Решение конфликтов

11. Перейти на ветку branch_3 и добавить файл файл 2.txt строку "the final fantasy in the 4.txt file".

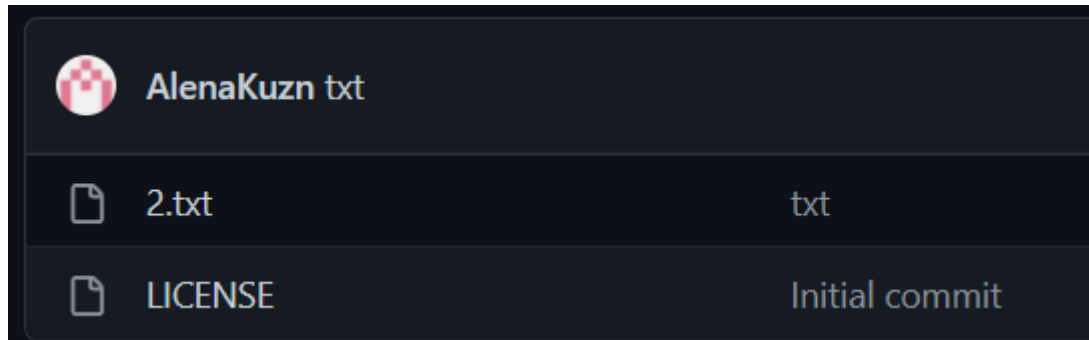


Рисунок 11 – Добавленный файл

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>`.