

Северо-Кавказский федеральный университет
Институт математики и информационных технологий

ОТЧЕТ
о выполнении лабораторной работы №3
по дисциплине
«Основы Программной Инженерии»

Выполнил:

Ботвинкин Никита Сергеевич

студент 2 курса, ПИЖ-б-о-20-1 группы
бакалавриата «Программная инженерия»
очной формы обучения

Ставрополь, 2021

СКРИНШОТЫ РЕПОЗИТОРИЯ

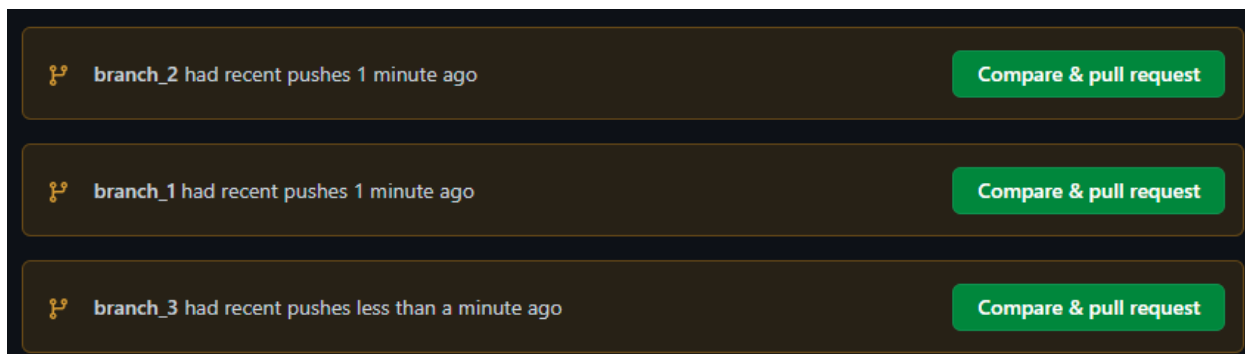


Рисунок 3.1 – обновленные ветки

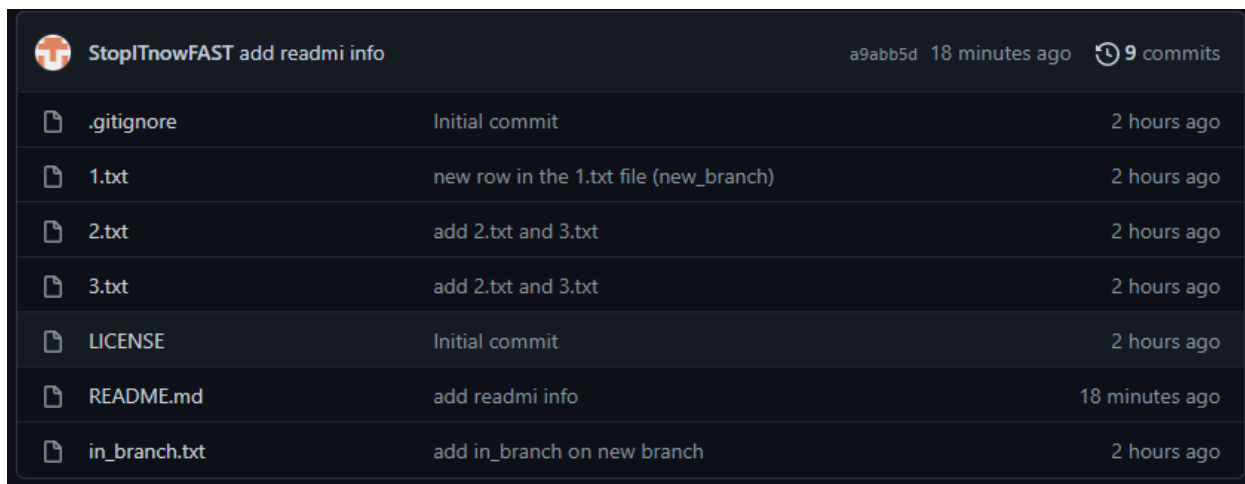


Рисунок 3.2 – файлы в главной ветке

СКРИНШОТЫ КОМАНДНОЙ СТРОКИ

```
C:\GitStuff\3Lab>git checkout branch_2
Switched to branch 'branch_2'
```

Рисунок 3.3 – переключение веток

```
C:\GitStuff\3Lab>git push origin branch_1
Enumerating objects: 50, done.
Counting objects: 100% (48/48), done.
Delta compression using up to 8 threads
Compressing objects: 100% (32/32), done.
Writing objects: 100% (44/44), 3.83 KiB | 489.00 KiB/s, done.
Total 44 (delta 16), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (16/16), completed with 1 local object.
remote:
remote: Create a pull request for 'branch_1' on GitHub by visiting:
remote:   https://github.com/StopITnowFAST/3Lab/pull/new/branch_1
remote:
To https://github.com/StopITnowFAST/3Lab.git
 * [new branch]      branch_1 -> branch_1
```

Рисунок 3.4 – отправка изменений ветки

```
C:\GitStuff\3Lab>git fetch origin
From https://github.com/StopITnowFAST/3Lab
 * [new branch]      branch_3 -> origin/branch_3
```

Рисунок 3.5 – получение ветки branch_3

```
C:\GitStuff\3Lab>git merge main
Already up to date.
```

Рисунок 3.6 – слияние веток

```
C:\GitStuff\3Lab>git branch
branch_1
* branch_2
branch_3
main
```

Рисунок 3.7 – все ветки

ОТВЕТЫ НА КОТРОЛЬНЫЕ ВОПРОСЫ

1. Ветка – это простой перемещаемый указатель на один из коммитов
2. HEAD – это указатель, задача которого ссылаться на определенный коммит в репозитории. Определенный коммит тут, это или родитель следующего коммита, или коммит, относительно которого будет создана рабочая копия во время операции checkout.
3. Ветку можно создать командой `git branch` (создание ветки, но не переключение), `git checkout -b` или через github.
4. Узнать текущую ветку можно переключившись на неё командой `git checkout`.
5. Переключиться на ветку можно командой `git checkout`. Если добавить параметр `-b`, то ветка не только переключится, но и создастся.
6. Удаленная ветка – это ссылка (указатель) в удаленном репозитории, которая включает ветки, теги и т.д.
7. Ветки отслеживания – это локальные ветки, которые связаны и удаленной веткой
8. Ветку отслеживания можно создать через команду `-track`
9. Изменения из локальной ветки в удаленную можно отправить по команде `git push <remote> <branch>`
10. `Git fetch` получает все изменения, которых у вас еще нет, но не будет изменять состояние рабочей директории. Команда `git fetch` – это грубо говоря порядок команд `git fetch` и `git merge`
11. Удалить удаленную ветку можно через команду `git push origin -delete <remote>`. Удалить локальную ветку можно через команду `git branch -d <branch>`
12. Типы веток `git flow`: функциональная, магистральная, ветка разработки, главная ветка, ветки выпуска, ветки исправления.
В `Git-flow` используется больше веток, каждая из которых существует дольше, а коммиты обычно крупнее. Разработчики создают функциональную ветку и откладывают ее слияние с главной магистральной веткой до завершения работы над функцией.
`Git-flow` не слишком удобно применять в процессах CI/CD.
13. В `SM` ветки представлены в виде линий разного цвета. По ним удобно смотреть, когда ветки были созданы, слиты, изменены и т.д. Для переключения веток есть отдельное окно.

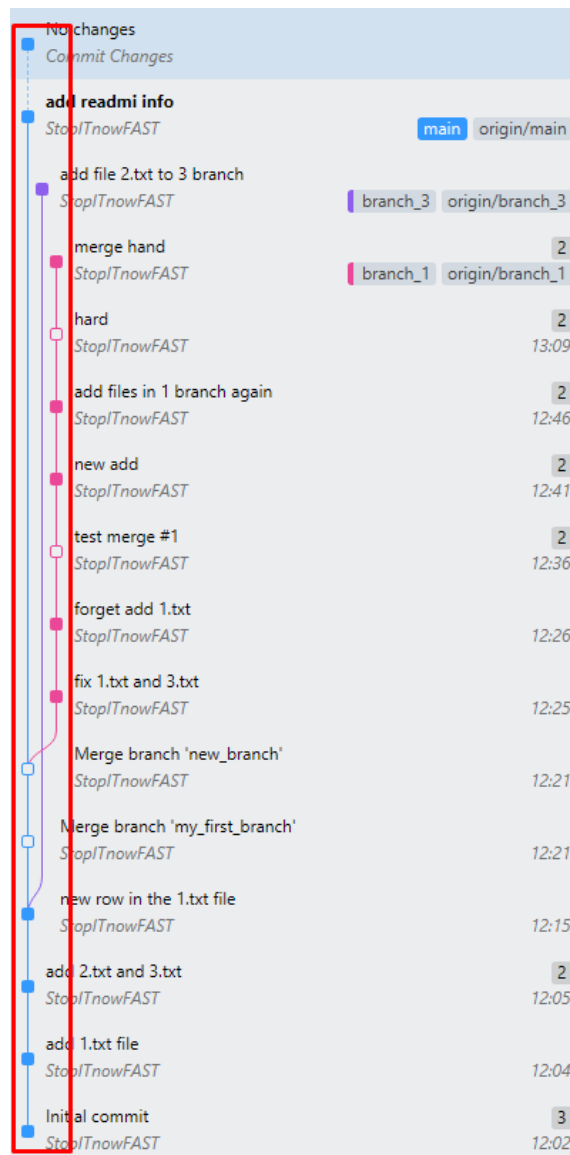


Рисунок 3.9 – ветки в sublime merge

Checkout Branch	Branch Name
branch_1	
branch_2	
branch_3	
main	

Рисунок 3.10 – переключение веток в sublime merge