

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 №3**  
**дисциплины «Программирование на Python»**

Выполнил:

Лейс Алексей Вячеславович  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Программное обеспечение средств  
вычислительной техники и  
автоматизированных систем»

---

(подпись)

Руководитель практики: кандидат тех.  
наук доцент кафедры  
инфокоммуникаций: Воронкин Р.А

---

(подпись)

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

Ставрополь, 2022 г.

**Тема:** Основы ветвления Git

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

Порядок выполнения работы:

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

Проиндексировал первый файл и сделать коммит с комментарием "add 1.txt file".

Проиндексировал второй и третий файлы.

Перезаписал уже сделанный коммит с новым комментарием "add 2.txt and 3.txt."

Результат:

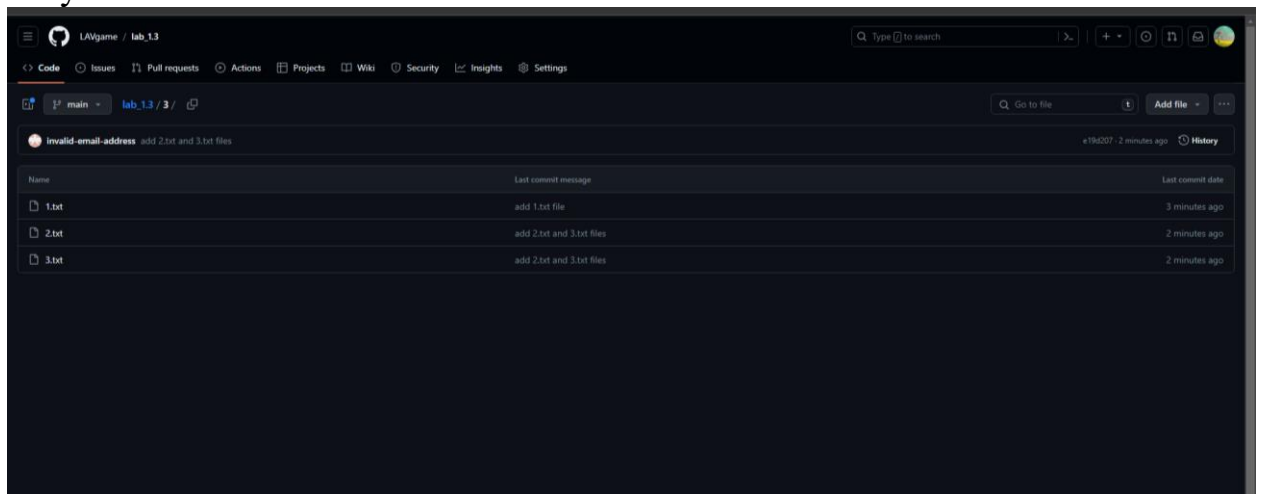


Рисунок 1 Созданные файлы

Создать новую ветку my\_first\_branch.

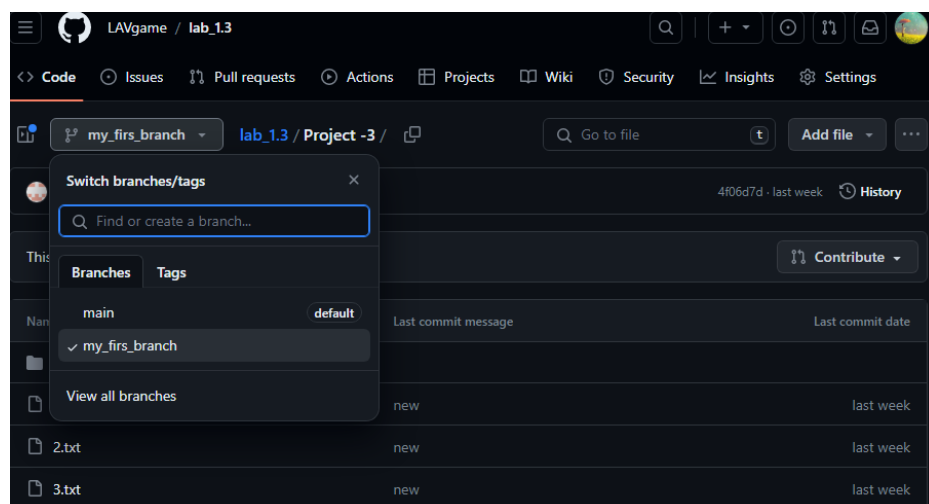


Рисунок 2 Созданная ветка

Создание на новой ветке файла.

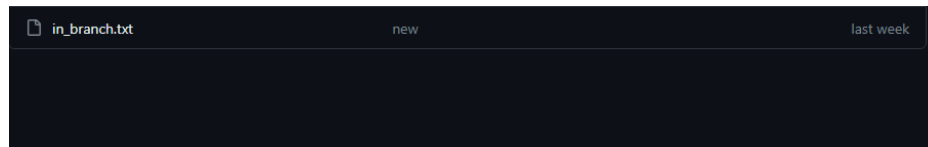


Рисунок 3 Файл на ветке

Удалить ветки my\_first\_branch и new\_branch.

```
Student@PC-01-9_521 MINGW64 /е/ИВТ-6-о-22-1/Программирование на Python/3/lab_1.
3 (main)
$ git branch new_branch -d
warning: deleting branch 'new_branch' that has been merged to
       'refs/remotes/origin/new_branch', but not yet merged to HEAD.
Deleted branch new_branch (was 3f8d8fc).

Student@PC-01-9_521 MINGW64 /е/ИВТ-6-о-22-1/Программирование на Python/3/lab_1.
3 (main)
$ git branch my_first_branch -d
error: The branch 'my_first_branch' is not fully merged.
If you are sure you want to delete it, run 'git branch -D my_first_branch'.

Student@PC-01-9_521 MINGW64 /е/ИВТ-6-о-22-1/Программирование на Python/3/lab_1.
3 (main)
$ git branch my_first_branch -D
Deleted branch my_first_branch (was 5ca7280).

Student@PC-01-9_521 MINGW64 /е/ИВТ-6-о-22-1/Программирование на Python/3/lab_1.
3 (main)
$ |
```

Рисунок 4 Удаление ветки

Создать ветки branch\_1 и branch\_2.

```
Student@PC-01-9_521 MINGW64 /е/ИВТ-6-о-22-1/Программирование на Python/3/lab_1.
3 (main)
$ git branch branch_1

Student@PC-01-9_521 MINGW64 /е/ИВТ-6-о-22-1/Программирование на Python/3/lab_1.
3 (main)
$ git branch branch_2
```

Рисунок 5 Создание веток

```

vipal@LAV MINGW64 /i/ИВТ-6-о-22-1/Программирование на Python/3/lab_1.3 (branch_
1)
$ git checkout branch_2
Switched to branch 'branch_2'

vipal@LAV MINGW64 /i/ИВТ-6-о-22-1/Программирование на Python/3/lab_1.3 (branch_
2)
$ git add .

vipal@LAV MINGW64 /i/ИВТ-6-о-22-1/Программирование на Python/3/lab_1.3 (branch_2)
$ git commit -m "new"
[branch_2 5dc16c2] new
 2 files changed, 2 insertions(+)

vipal@LAV MINGW64 /i/ИВТ-6-о-22-1/Программирование на Python/3/lab_1.3 (branch_2)
$ git push
fatal: The current branch branch_2 has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin branch_2

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.

vipal@LAV MINGW64 /i/ИВТ-6-о-22-1/Программирование на Python/3/lab_1.3 (branch_2)
$ ^C

vipal@LAV MINGW64 /i/ИВТ-6-о-22-1/Программирование на Python/3/lab_1.3 (branch_2)
$ git push --set-upstream origin branch_2
Enumerating objects: 9, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 397 bytes | 198.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'branch_2' on GitHub by visiting:
remote:   https://github.com/LAVgame/lab_1.3/pull/new/branch_2
remote:
To https://github.com/LAVgame/lab_1.3
 * [new branch]      branch_2 -> branch_2
branch 'branch_2' set up to track 'origin/branch_2'.

```

Рисунок 5 Переход из веток

```

vipal@LAV MINGW64 /i/ИВТ-6-о-22-1/Программирование на Python/3/lab_1.3 (branch_2)
$ git checkout branch_1
Switched to branch 'branch_1'
Your branch is up to date with 'origin/branch_1'.

vipal@LAV MINGW64 /i/ИВТ-6-о-22-1/Программирование на Python/3/lab_1.3 (branch_1)
$ git merge branch_2
Auto-merging Project -3/1.txt
CONFLICT (content): Merge conflict in Project -3/1.txt
Auto-merging Project -3/3.txt
CONFLICT (content): Merge conflict in Project -3/3.txt
Automatic merge failed; fix conflicts and then commit the result.

vipal@LAV MINGW64 /i/ИВТ-6-о-22-1/Программирование на Python/3/lab_1.3 (branch_1|MERGING)
$ 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)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
   both modified:   Project -3/1.txt
   both modified:   Project -3/3.txt

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

```

Рисунок 6 Слияние веток

Отправить ветку branch\_1 на GitHub.

```
Changes to be committed:
  modified:   Project -3/1.txt
  modified:   Project -3/3.txt

vipal@LAV MINGW64 /i/ИБТ-6-о-22-1/Программирование на Python/3/lab_1.3 (branch_1|MERGING)
$ git commit -m "new1,2"
[branch_1 3099bf1] new1,2

vipal@LAV MINGW64 /i/ИБТ-6-о-22-1/Программирование на Python/3/lab_1.3 (branch_1)
$ git push
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 484 bytes | 48.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/LAVgame/lab_1.3
   c25375e..3099bf1  branch_1 -> branch_1

vipal@LAV MINGW64 /i/ИБТ-6-о-22-1/Программирование на Python/3/lab_1.3 (branch_1)
$ git branch branch_3

vipal@LAV MINGW64 /i/ИБТ-6-о-22-1/Программирование на Python/3/lab_1.3 (branch_1)
$ git checkout branch_3
Switched to branch 'branch_3'

vipal@LAV MINGW64 /i/ИБТ-6-о-22-1/Программирование на Python/3/lab_1.3 (branch_3)
$
```

Рисунок 7 Готовые изменения и отправка

```
aac9ac7 (log) HEAD@{18}: checkout: moving from new_branch to main
5e40f42 HEAD@{19}: commit: new 1.txt
aac9ac7 (log) HEAD@{20}: checkout: moving from main to new_branch
aac9ac7 (log) HEAD@{21}: checkout: moving from new_branch to main
aac9ac7 (log) HEAD@{22}: checkout: moving from main to new_branch
aac9ac7 (log) HEAD@{23}: checkout: moving from my_firs_branch to main
4f06d7d (origin/my_firs_branch, my_firs_branch) HEAD@{24}: commit: new
aac9ac7 (log) HEAD@{25}: checkout: moving from main to my_firs_branch
aac9ac7 (log) HEAD@{26}: commit: new
05a8aa6 HEAD@{27}: pull: Merge made by the 'recursive' strategy.
2e50094 HEAD@{28}: checkout: moving from my_first_branch to main
5ca7280 HEAD@{29}: commit: new
e19d207 HEAD@{30}: checkout: moving from main to my_first_branch
2e50094 HEAD@{31}: commit: new
b6d8fc9 HEAD@{32}: commit: new
e19d207 HEAD@{33}: commit: add 2.txt and 3.txt files
309f7d0 HEAD@{34}: commit: add 1.txt file
49626cf HEAD@{35}: clone: from https://github.com/LAVgame/lab_1.3
```

Рисунок 8 Создание 3 ветки и слияние Main с 2й веткой

### Ответы на вопросы:

1. Что такое ветка?

Ветка в Git - это отдельная линия разработки, которая позволяет работать над изменениями независимо от других веток.

## 2. Что такое HEAD?

HEAD - это указатель на текущий коммит в вашей репозитории. Он указывает на последний коммит в текущей активной ветке.

## 3. Способы создания веток:

- ``git branch <branch_name>`` - создать локальную ветку.
- ``git checkout -b <branch_name>`` - создать и переключиться на новую локальную ветку.
- ``git push origin <branch_name>`` - создать удаленную ветку.

## 4. Как узнать текущую ветку?

Используйте команду ``git branch`` или ``git status`` для определения текущей ветки.

## 5. Как переключаться между ветками?

Используйте команду ``git checkout <branch_name>`` для переключения на другую ветку.

## 6. Что такое удаленная ветка?

Удаленная ветка - это ветка, которая существует на удаленном сервере Git, а не локально.

## 7. Что такое ветка отслеживания?

Ветка отслеживания (tracking branch) - это локальная ветка, связанная с удаленной веткой, что позволяет отслеживать изменения на удаленном сервере.

## 8. Как создать ветку отслеживания?

Используйте команду ``git checkout --track origin/<branch_name>`` для создания локальной ветки, отслеживающей удаленную ветку.

## 9. Как отправить изменения из локальной ветки в удаленную ветку?

Используйте команду ``git push origin <local_branch>:<remote_branch>``.

## 10. В чем отличие команд git fetch и git pull?

``git fetch`` загружает изменения с удаленного репозитория, но не объединяет их с текущей веткой, в то время как ``git pull`` загружает и объединяет изменения.

## 11. Как удалить локальную и удаленную ветки?

- Для удаления локальной ветки: ``git branch -d <branch_name>`` или ``git branch -D <branch_name>`` (если ветка не слита).

- Для удаления удаленной ветки: ``git push origin --delete <branch_name>``.

## 12. Основные типы веток в модели git-flow:

Основные типы веток в модели git-flow включают "feature" (для новых функций), "release" (для подготовки релизов), "hotfix" (для быстрых исправлений), "develop" (для разработки) и "master" (для стабильных релизов).

Организация работы с ветками в git-flow предполагает создание и слияние конкретных типов веток в соответствии с целями разработки.

Недостатки git-flow включают сложность и избыточность структуры ветвления, что может быть излишним для небольших проектов.

## 13. Инструменты для работы с ветками Git в средствах с GUI:

- GitHub Desktop: Обеспечивает удобный интерфейс для создания, переключения и удаления веток.

- Sourcetree: Позволяет визуально управлять ветками, включая создание, слияние и удаление.

- GitKraken: Обладает функциональностью для работы с ветками и отслеживания изменений.

- GitExtensions: Предоставляет удобные инструменты для управления ветками и слияниями.

**Вывод:** В ходе исследования базовых возможностей по работе с локальными и удаленными ветками Git были рассмотрены ключевые аспекты управления ветками в системе контроля версий Git.