

Федеральное государственное автономное образовательное учреждение  
высшего образования «Северо-Кавказский федеральный университет»  
Институт цифрового развития

ОТЧЕТ ПО ДИСЦИПЛИНЕ  
ОСНОВЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ

Выполнила:

Мирзаева Камилла Мирзаевна

2 курс, группа ПИЖ-б-о-20-1

Принял:

Воронкин Роман Александрович

Ставрополь, 2021 г.

```
C:\Users\Камилла>cd work3
Системе не удается найти указанный путь.

C:\Users\Камилла>git clone https://github.com/mirzaevakamilla/work3.git
Cloning into 'work3'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.

C:\Users\Камилла>cd work3

C:\Users\Камилла\work3>explorer .

C:\Users\Камилла\work3>git status
On branch main
Your branch is up to date with 'origin/main'.

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

nothing added to commit but untracked files present (use "git add" to track)

C:\Users\Камилла\work3>git add .

C:\Users\Камилла\work3>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.txt
    new file:   2.txt.txt
    new file:   3.txt.txt

C:\Users\Камилла\work3>git commit -m "Добавление трех файлов "
[main a71eb3c] Добавление трех файлов
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 1.txt.txt
 create mode 100644 2.txt.txt
 create mode 100644 3.txt.txt

C:\Users\Камилла\work3>git checkout -- 1.txt
error: pathspec '1.txt' did not match any file(s) known to git

C:\Users\Камилла\work3>git checkout -- 1.txt.txt

C:\Users\Камилла\work3>git commit -m "add 2.txt and 3.txt"
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\Users\Камилла\work3>_
```

Рисунок 1 - скриншот терминала командами Git

```
C:\Users\Камилла\work3>git add .

C:\Users\Камилла\work3>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.txt
        new file:   2.txt.txt
        new file:   3.txt.txt

C:\Users\Камилла\work3>git commit -m "Добавление трех файлов "
[main a71eb3c] Добавление трех файлов
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt.txt
create mode 100644 2.txt.txt
create mode 100644 3.txt.txt

C:\Users\Камилла\work3>git checkout -- 1.txt
error: pathspec '1.txt' did not match any file(s) known to git

C:\Users\Камилла\work3>git checkout -- 1.txt.txt

C:\Users\Камилла\work3>git commit -m "add 2.txt and 3.txt"
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\Users\Камилла\work3>git branch my_first_branch

C:\Users\Камилла\work3>git checkout my_first_branch
Switched to branch 'my_first_branch'

C:\Users\Камилла\work3>git add in_branch.txt
fatal: pathspec 'in_branch.txt' did not match any files

C:\Users\Камилла\work3>git add in_branch.txt

C:\Users\Камилла\work3>git commit -m "change"
[my_first_branch f768ef3] change
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 in_branch.txt

C:\Users\Камилла\work3>
```

Рисунок 2 - скриншот терминала командами Git

```
Git CMD
C:\Users\Камилла\work3>git checkout -- 1.txt
error: pathspec '1.txt' did not match any file(s) known to git

C:\Users\Камилла\work3>git checkout -- 1.txt.txt

C:\Users\Камилла\work3>git commit -m "add 2.txt and 3.txt"
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\Users\Камилла\work3>git branch my_first_branch

C:\Users\Камилла\work3>git checkout my_first_branch
Switched to branch 'my_first_branch'

C:\Users\Камилла\work3>git add in_branch.txt
fatal: pathspec 'in_branch.txt' did not match any files

C:\Users\Камилла\work3>git add in_branch.txt

C:\Users\Камилла\work3>git commit -m "change"
[my_first_branch f768ef3] change
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 in_branch.txt

C:\Users\Камилла\work3>git branch master

C:\Users\Камилла\work3>git log
commit f768ef32fd9d91fce6da1b1e2722daedcf4d9637 (HEAD -> my_first_branch, master)
Author: mirzaevakamilla <mirzaeva.ncfu>
Date: Tue Sep 21 22:09:30 2021 +0300

    change

commit a71eb3c816a7a689088ae4e6a23cabd15663b63c (main)
Author: mirzaevakamilla <mirzaeva.ncfu>
Date: Tue Sep 21 21:33:03 2021 +0300

    Добавление трех файлов

commit 4aa236ef352f354bbc0520d00cbd15167cfd683d (origin/main, origin/HEAD)
Author: mirzaevakamilla <90276669+mirzaevakamilla@users.noreply.github.com>
Date: Tue Sep 21 21:23:49 2021 +0300

    Initial commit

C:\Users\Камилла\work3>git checkout master
Switched to branch 'master'

C:\Users\Камилла\work3>git branch new_branch

C:\Users\Камилла\work3>git checkout new_branch
Switched to branch 'new_branch'

C:\Users\Камилла\work3>git add .

C:\Users\Камилла\work3>git status
On branch new_branch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   1.txt.txt

C:\Users\Камилла\work3>git commit -m "Изменения в файле 1"
[new_branch 7a2de1e] Изменения в файле 1
 1 file changed, 1 insertion(+)

C:\Users\Камилла\work3>
```

Рисунок 3 - скриншот терминала командами Git

```
Git CMD
Author: mirzaevakamilla <mirzaeva.ncfu>
Date: Tue Sep 21 22:09:30 2021 +0300

change

commit a71eb3c816a7a689088ae4e6a23cabd15663b63c (main)
Author: mirzaevakamilla <mirzaeva.ncfu>
Date: Tue Sep 21 21:33:03 2021 +0300

Добавление трех файлов

commit 4aa236ef352f354bbc0520d00cbd15167cfd683d (origin/main, origin/HEAD)
Author: mirzaevakamilla <90276669+mirzaevakamilla@users.noreply.github.com>
Date: Tue Sep 21 21:23:49 2021 +0300

Initial commit

C:\Users\Камилла\work3>git checkout master
Switched to branch 'master'

C:\Users\Камилла\work3>git branch new_branch

C:\Users\Камилла\work3>git checkout new_branch
Switched to branch 'new_branch'

C:\Users\Камилла\work3>git add .

C:\Users\Камилла\work3>git status
On branch new_branch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   1.txt.txt

C:\Users\Камилла\work3>git commit -m "Изменения в файле 1"
[new_branch 7a2dele] Изменения в файле 1
1 file changed, 1 insertion(+)


C:\Users\Камилла\work3>git checkout master
Switched to branch 'master'

C:\Users\Камилла\work3>git merge my_first_branch
Already up to date.

C:\Users\Камилла\work3>git merge new_branch
Updating f768ef3..7a2dele
Fast-forward
 1.txt.txt | 1 +
 1 file changed, 1 insertion(+)

C:\Users\Камилла\work3>
```

Рисунок 4 - скриншот терминала командами Git

 Git CMD  

```
C:\Users\Камилла\work3>git commit -m "Изменения в файле 1"
[new_branch 7a2de1e] Изменения в файле 1
1 file changed, 1 insertion(+)

C:\Users\Камилла\work3>git checkout master
Switched to branch 'master'

C:\Users\Камилла\work3>git merge my_first_branch
Already up to date.

C:\Users\Камилла\work3>git merge new_branch
Updating f768ef3..7a2de1e
Fast-forward
 1.txt.txt | 1 +
1 file changed, 1 insertion(+)

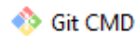
C:\Users\Камилла\work3>git branch --no-merged

C:\Users\Камилла\work3>git branch -d my_first_branch
Deleted branch my_first_branch (was f768ef3).

C:\Users\Камилла\work3>git branch -d new_branch
Deleted branch new_branch (was 7a2de1e).

C:\Users\Камилла\work3>_
```

Рисунок 5 - скриншот терминала командами Git



```
On branch new_branch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   1.txt.txt

C:\Users\Камилла\work3>git commit -m "Изменения в файле 1"
[new_branch 7a2dele] Изменения в файле 1
1 file changed, 1 insertion(+)

C:\Users\Камилла\work3>git checkout master
Switched to branch 'master'

C:\Users\Камилла\work3>git merge my_first_branch
Already up to date.

C:\Users\Камилла\work3>git merge new_branch
Updating f768ef3..7a2dele
Fast-forward
 1.txt.txt | 1 +
1 file changed, 1 insertion(+)

C:\Users\Камилла\work3>git branch --no-merged

C:\Users\Камилла\work3>git branch -d my_first_branch
Deleted branch my_first_branch (was f768ef3).

C:\Users\Камилла\work3>git branch -d new_branch
Deleted branch new_branch (was 7a2dele).

C:\Users\Камилла\work3>git branch branch_1

C:\Users\Камилла\work3>git branch branch_2

C:\Users\Камилла\work3>git checkout branch_1
Switched to branch 'branch_1'

C:\Users\Камилла\work3>git add .

C:\Users\Камилла\work3>git status
On branch branch_1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   1.txt.txt
        modified:   3.txt.txt

C:\Users\Камилла\work3>git commit -m "Change 1,3"
[branch_1 0ffb2a9] Change 1,3
2 files changed, 2 insertions(+), 1 deletion(-)

C:\Users\Камилла\work3>git checkout branch_2
Switched to branch 'branch_2'

C:\Users\Камилла\work3>git add .

C:\Users\Камилла\work3>git status
On branch branch_2
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   1.txt.txt
        modified:   3.txt.txt

C:\Users\Камилла\work3>git commit -m "change"
[branch_2 2e8ac75] change
2 files changed, 2 insertions(+), 1 deletion(-)

C:\Users\Камилла\work3>
```

Рисунок 6 -скриншот терминала командами Git

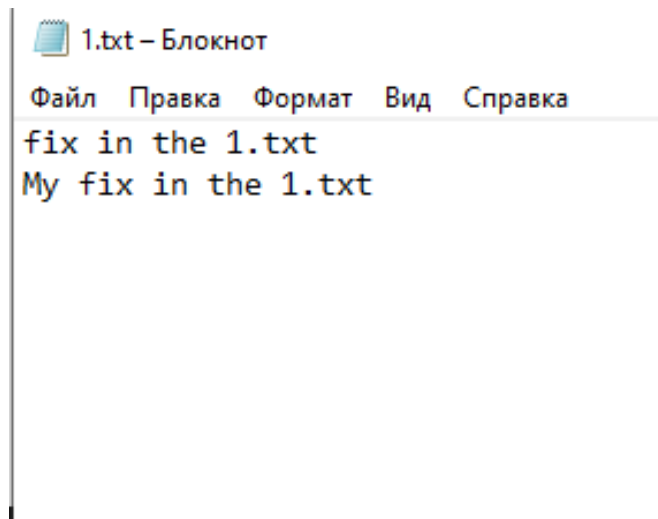


Рисунок 7 - решение конфликта ручным способом

```
Git CMD

C:\Users\Камилла\work3>git commit -m "Change 1,3"
[branch_1 0ffb2a9] Change 1,3
 2 files changed, 2 insertions(+), 1 deletion(-)

C:\Users\Камилла\work3>git checkout branch_2
Switched to branch 'branch_2'

C:\Users\Камилла\work3>git add .

C:\Users\Камилла\work3>git status
On branch branch_2
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   1.txt.txt
        modified:   3.txt.txt

C:\Users\Камилла\work3>git commit -m "change"
[branch_2 2e8ac75] change
 2 files changed, 2 insertions(+), 1 deletion(-)

C:\Users\Камилла\work3>git checkout branch_1
Switched to branch 'branch_1'

C:\Users\Камилла\work3>git merge branch_2
Auto-merging 3.txt.txt
CONFLICT (content): Merge conflict in 3.txt.txt
Auto-merging 1.txt.txt
CONFLICT (content): Merge conflict in 1.txt.txt
Automatic merge failed; fix conflicts and then commit the result.

C:\Users\Камилла\work3>git add .

C:\Users\Камилла\work3>git mergetool

This message is displayed because 'merge.tool' is not configured.
See 'git mergetool --tool-help' or 'git help config' for more details.
'git mergetool' will now attempt to use one of the following tools:
tortoisemerge emerge vimdiff nvimdiff
No files need merging

C:\Users\Камилла\work3>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

C:\Users\Камилла\work3>
```

Рисунок 8 - скриншот терминала командами Git



```
Git CMD

C:\Users\Камилла\work3>git push
fatal: The current branch branch_1 has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin branch_1

C:\Users\Камилла\work3>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

C:\Users\Камилла\work3>git commit -m "change1"
[branch_1 2189454] change1

C:\Users\Камилла\work3>git push
fatal: The current branch branch_1 has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin branch_1

C:\Users\Камилла\work3>git push --set-upstream origin branch_1
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 12 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (20/20), 1.59 KiB | 1.59 MiB/s, done.
Total 20 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), done.
remote:
remote: Create a pull request for 'branch_1' on GitHub by visiting:
remote:   https://github.com/mirzaevakamilla/work3/pull/new/branch_1
remote:
To https://github.com/mirzaevakamilla/work3.git
 * [new branch]      branch_1 -> branch_1
Branch 'branch_1' set up to track remote branch 'branch_1' from 'origin'.

C:\Users\Камилла\work3>
```

Рисунок 9 - скриншот терминала командами Git

```
Git CMD

git push --set-upstream origin branch_1

C:\Users\Камилла\work3>git push --set-upstream origin branch_1
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 12 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (20/20), 1.59 KiB | 1.59 MiB/s, done.
Total 20 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), done.
remote:
remote: Create a pull request for 'branch_1' on GitHub by visiting:
remote:   https://github.com/mirzaevakamilla/work3/pull/new/branch_1
remote:
To https://github.com/mirzaevakamilla/work3.git
 * [new branch]      branch_1 -> branch_1
Branch 'branch_1' set up to track remote branch 'branch_1' from 'origin'.

C:\Users\Камилла\work3>git branch branch_3

C:\Users\Камилла\work3>git checkout branch_3
Switched to branch 'branch_3'

C:\Users\Камилла\work3>git push origin branch_3
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'branch_3' on GitHub by visiting:
remote:   https://github.com/mirzaevakamilla/work3/pull/new/branch_3
remote:
To https://github.com/mirzaevakamilla/work3.git
 * [new branch]      branch_3 -> branch_3

C:\Users\Камилла\work3>
```

Рисунок 10 - скриншот терминала командами Git

```
Git CMD
warning: refname 'origin/branch_3' is ambiguous.
Switched to branch 'origin/branch_3'

C:\Users\Камилла\work3>git checkout branch_3
Switched to branch 'branch_3'

C:\Users\Камилла\work3>git add .

C:\Users\Камилла\work3>git commit -m "change2"
[branch_3 3a4c327] change2
1 file changed, 1 insertion(+)

C:\Users\Камилла\work3>
```

Рисунок 11 - скриншот терминала командами Git

```
Git CMD
branch_1
branch_2
* branch_3
main
master
origin/branch_3

C:\Users\Камилла\work3>git checkout origin/branch_3
warning: refname 'origin/branch_3' is ambiguous.
Switched to branch 'origin/branch_3'

C:\Users\Камилла\work3>git checkout branch_3
Switched to branch 'branch_3'

C:\Users\Камилла\work3>git add .

C:\Users\Камилла\work3>git commit -m "change2"
[branch_3 3a4c327] change2
1 file changed, 1 insertion(+)

C:\Users\Камилла\work3>git checkout --track origin/branch_3
warning: refname 'origin/branch_3' is ambiguous.
fatal: A branch named 'branch_3' already exists.

C:\Users\Камилла\work3>git checkout branch_2
Switched to branch 'branch_2'

C:\Users\Камилла\work3>git rebase master
Current branch branch_2 is up to date.

C:\Users\Камилла\work3>git push 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/mirzaevakamilla/work3/pull/new/branch_2
remote:
To https://github.com/mirzaevakamilla/work3.git
 * [new branch]      branch_2 -> branch_2

C:\Users\Камилла\work3>git push origin master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/mirzaevakamilla/work3/pull/new/master
remote:
To https://github.com/mirzaevakamilla/work3.git
 * [new branch]      master -> master

C:\Users\Камилла\work3>
```

Рисунок 12 - скриншот терминала командами Git

## Ответы на контрольные вопросы:

### 1. Что такое ветка?

Почти каждая система контроля версий (СКВ) в какой-то форме поддерживает ветвление. Используя ветвление, Вы отклоняетесь от основной линии разработки и продолжаете работу независимо от неё, не вмешиваясь в основную линию. Ветка в Git — это простой перемещаемый указатель на один из таких коммитов. По умолчанию, имя основной ветки в Git — master. Как только вы начнёте создавать коммиты, ветка master будет всегда указывать на последний коммит. Каждый раз при создании коммита указатель ветки master будет передвигаться на следующий коммит автоматически.

### 2. Что такое HEAD?

HEAD — это указатель, задача которого ссылаться на определенный коммит в репозитории.

### 3. Способы создания веток.

Вы можете это сделать командой `git branch`. Команда `git branch` только создаёт новую ветку, но не переключает на неё.

### 4. Как узнать текущую ветку?

Вы можете легко это увидеть при помощи простой команды `git log`, которая покажет вам куда указывают указатели веток. Эта опция называется `--decorate`.

### 5. Как переключаться между ветками?

Для переключения на существующую ветку выполните команду `git checkout`.

### 6. Что такое удаленная ветка?

Удалённые ссылки — это ссылки (указатели) в ваших удалённых репозиториях, включая ветки, теги и так далее.

### 7. Что такое ветка отслеживания?

Ветки слежения — это ссылки на определённое состояние удалённых веток. Это локальные ветки, которые нельзя перемещать; Git перемещает их автоматически при любой коммуникации с удалённым репозиторием, чтобы гарантировать точное соответствие с ним. Представляйте их как закладки для напоминания о том, где ветки в удалённых репозиториях находились во время последнего подключения к ним.

### 8. Как создать ветку отслеживания?

Имена веток слежения имеют вид `<remote>/<branch>` . Например, если вы хотите посмотреть, как выглядела ветка `master` на сервере `origin` во время последнего соединения с ним, используйте ветку `origin/master` . Если вы с коллегой работали над одной задачей и он отправил на сервер ветку `iss53` , при том что у вас может быть своя локальная ветка `iss53` , удалённая ветка будет представлена веткой слежения с именем `origin/iss53` .

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

Если у вас есть ветка `serverfix` , над которой вы хотите работать с кем-то ещё, вы можете отправить её точно так же, как вы отправляли вашу первую ветку. Выполните команду `git push <remote> <branch>`.

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

Команда `git fetch` получает с сервера все изменения, которых у вас ещё нет, но не будет изменять состояние вашей рабочей директории. Эта команда просто получает данные и позволяет вам самостоятельно сделать слияние. Тем не менее, существует команда `git pull` , которая в большинстве случаев является командой `git fetch` , за которой непосредственно следует команда `git merge` . Если у вас настроена ветка слежения как показано в предыдущем разделе, или она явно установлена, или она была создана автоматически командами `clone` или `checkout` , `git pull` определит сервер и ветку, за которыми следит ваша текущая ветка, получит данные с этого сервера и затем попытается слить удалённую ветку.

#### 11. Как удалить локальную и удалённую ветки?

Вы можете удалить ветку на удалённом сервере используя параметр `--delete` для команды `git push` . Для удаления ветки `serverfix` на сервере, выполните следующую команду: `git push origin --delete serverfix`

#### 12. Изучить модель ветвления `git-flow` (использовать материалы статей <https://www.atlassian.com/ru/git/tutorials/comparing-workflows/gitflow-workflow>, <https://habr.com/ru/post/106912/>). Какие основные типы веток присутствуют в модели `git-flow`? Как организована работа с ветками в модели `git-flow`? В чем недостатки `git-flow`?

`Git-flow` — альтернативная модель ветвления `Git`, в которой используются функциональные ветки и несколько основных веток. В этом рабочем процессе для регистрации истории проекта вместо одной ветки `main` используются две ветки. В главной ветке `main` хранится официальная история релиза, а ветка разработки `develop` предназначена для объединения всех функций. Кроме того, для удобства рекомендуется присваивать всем коммитам в ветке `main` номер версии.

13. На прошлой лабораторной работе было задание выбрать одно из программных средств с GUI для работы с Git. Необходимо в рамках этого вопроса привести описание инструментов для работы с ветками Git, предоставляемых этим средством.