

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**  
**дисциплины «Основы кроссплатформенного программирования»**

Выполнил:

Кононов Дмитрий Александрович

1 курс, группа ИТС-б-о-22-1,

11.03.02 «Инфокоммуникационные  
технологии и системы связи»,

направленность (профиль)

«Инфокоммуникационные системы и  
сети», очная форма обучения

---

(подпись)

Руководитель практики:

Воронкин Р. А., доцент кафедры  
инфокоммуникаций

---

(подпись)

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

Ставрополь, 2023 г.

## Тема: Основы ветвления Git

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

### Ход работы:

```
Администратор: Windows Pc
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 1.79 KiB | 1.79 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/KononovDmitry/1.3.git
   d3b7386..ba9513c  main -> main
PS C:\Git\1.3> 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
       2.txt
       3.txt

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Git\1.3> git add 1.txt
PS C:\Git\1.3> git commit -m "add 1.txt file"
[main eefab3b] add 1.txt file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 1.txt
PS C:\Git\1.3> git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 343 bytes | 343.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/KononovDmitry/1.3.git
   ba9513c..eefab3b  main -> main
PS C:\Git\1.3> git add 2.txt
PS C:\Git\1.3> git add 3.txt
PS C:\Git\1.3> git push
Everything up-to-date
PS C:\Git\1.3> |
```

Проиндексировал первый, второй и третий файл

```
Администратор: Windows Pc x + v
ba9513c..eefab3b main -> main
PS C:\Git\1.3> git add 2.txt
PS C:\Git\1.3> git add 3.txt
PS C:\Git\1.3> git push
Everything up-to-date
PS C:\Git\1.3> git log
commit eefab3b1c04cb3c5c0002dc29c4d7f955eal402a (HEAD -> main, origin/main, origin/HEAD)
Author: MononovDmitry <dmononov2903@gmail.com>
Date: Wed Mar 29 13:58:19 2023 +0300

    add 1.txt file

commit ba9513ce37bc8a128eb0ee1843ad94e28e0e4d3d
Author: MononovDmitry <dmononov2903@gmail.com>
Date: Wed Mar 29 13:57:19 2023 +0300

    add files

commit d3b7306397bd041765d2f492a7580c11a614d09d
Author: MononovDmitry <126630318+MononovDmitry@users.noreply.github.com>
Date: Wed Mar 29 13:56:17 2023 +0300

    Initial commit
PS C:\Git\1.3> git rebase -i HEAD~N
fatal: invalid upstream 'HEAD~N'
PS C:\Git\1.3> git add 2.txt 3.txt
PS C:\Git\1.3> git commit -m "add 2.txt and 3.txt"
[main 47cd68c] add 2.txt and 3.txt
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 2.txt
create mode 100644 3.txt
PS C:\Git\1.3> git rebase -i HEAD~1
hint: Waiting for your editor to close the file... unix2dos: converting file C:/Git/1.3/.git/rebase-merge/git-rebase-todo to DOS format...
dos2unix: converting file C:/Git/1.3/.git/rebase-merge/git-rebase-todo to Unix format...
Stopped at 47cd68c... add 2.txt and 3.txt
You can amend the commit now, with

    git commit --amend

Once you are satisfied with your changes, run

    git rebase --continue
PS C:\Git\1.3> git commit --amend -m "add 2.txt and 3.txt"
[detached HEAD 1alc1fc] add 2.txt and 3.txt
Date: Wed Mar 29 14:06:50 2023 +0300
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 2.txt
create mode 100644 3.txt
PS C:\Git\1.3> git add 2.txt 3.txt
PS C:\Git\1.3> git rebase --continue
Successfully rebased and updated refs/heads/main.
PS C:\Git\1.3> git push --force
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 244 bytes | 244.00 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/MononovDmitry/1.3.git
eefab3b..1alc1fc main -> main
PS C:\Git\1.3>
```

Перезаписал созданный коммит

```
Администратор: Windows Pc x + v

git commit --amend

Once you are satisfied with your changes, run

git rebase --continue
PS C:\Git\1.3> git commit --amend -m "add 2.txt and 3.txt"
[detached HEAD 1a1c1fc] add 2.txt and 3.txt
Date: Wed Mar 29 14:06:50 2023 +0300
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 2.txt
create mode 100644 3.txt
PS C:\Git\1.3> git add 2.txt 3.txt
PS C:\Git\1.3> git rebase --continue
Successfully rebased and updated refs/heads/main.
PS C:\Git\1.3> git push --force
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 244 bytes | 244.00 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/KononovDmitry/1.3.git
eefab3b..1a1c1fc main -> main
PS C:\Git\1.3> git branch
* main
PS C:\Git\1.3> git branch my_first_branch
PS C:\Git\1.3> git checkout my_first_branch
Switched to branch 'my_first_branch'
PS C:\Git\1.3> git checkout -b my_first_branch
fatal: a branch named 'my_first_branch' already exists
PS C:\Git\1.3> git push
fatal: The current branch my_first_branch has no upstream branch.
To push the current branch and set the remote as upstream, use

git push --set-upstream origin my_first_branch

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

PS C:\Git\1.3> |
```

Создал новую ветку

```
Администратор: Windows Pc
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 244 bytes | 244.00 MiB/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/MononovDmitry/1.3.git
   eefab3b..1a1c1fc  main -> main
PS C:\Git\1.3> git branch
* main
PS C:\Git\1.3> git branch my_first_branch
PS C:\Git\1.3> git checkout my_first_branch
Switched to branch 'my_first_branch'
PS C:\Git\1.3> git checkout -b my_first_branch
fatal: a branch named 'my_first_branch' already exists
PS C:\Git\1.3> git push
fatal: The current branch my_first_branch has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin my_first_branch

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.
PS C:\Git\1.3> git branch
  main
* my_first_branch
PS C:\Git\1.3> git checkout my_first_branch
Already on 'my_first_branch'
PS C:\Git\1.3> touch in_branch.txt
touch : Имя "touch" не распознано как имя командлета, функции, файла сценария или выполняемой программы. Проверьте прав
ильность написания имени, а также наличие и правильность пути, после чего повторите попытку.
строка:1 знак:1
+ touch in_branch.txt
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (touch:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Git\1.3> touch in_branch.txt
touch : Имя "touch" не распознано как имя командлета, функции, файла сценария или выполняемой программы. Проверьте прав
ильность написания имени, а также наличие и правильность пути, после чего повторите попытку.
строка:1 знак:1
+ touch in_branch.txt
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (touch:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Git\1.3> git add in_branch.txt
PS C:\Git\1.3> git commit -m "Add in_branch.txt file"
[my_first_branch bd2860e] Add in_branch.txt file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 in_branch.txt
PS C:\Git\1.3> git push
fatal: The current branch my_first_branch has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin my_first_branch

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.
PS C:\Git\1.3> |
```

Создал новый файл in\_branch.txt

```
Администратор: Windows Pc X + v

PS C:\Git\1.3> git branch
  main
* my_first_branch
PS C:\Git\1.3> git checkout my_first_branch
Already on 'my_first_branch'
PS C:\Git\1.3> touch in_branch.txt
touch : Имя "touch" не распознано как имя командлета, функции, файла сценария или выполняемой программы. Проверьте правильность написания имени, а также наличие и правильность пути, после чего повторите попытку.
строка:1 знак:1
+ touch in_branch.txt
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (touch:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Git\1.3> touch in_branch.txt
touch : Имя "touch" не распознано как имя командлета, функции, файла сценария или выполняемой программы. Проверьте правильность написания имени, а также наличие и правильность пути, после чего повторите попытку.
строка:1 знак:1
+ touch in_branch.txt
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (touch:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Git\1.3> git add in_branch.txt
PS C:\Git\1.3> git commit -m "Add in_branch.txt file"
[my_first_branch bd2860e] Add in_branch.txt file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 in_branch.txt
PS C:\Git\1.3> git push
fatal: The current branch my_first_branch has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin my_first_branch

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

PS C:\Git\1.3> git branch
  main
* my_first_branch
PS C:\Git\1.3> git checkout master
error: pathspec 'master' did not match any file(s) known to git
PS C:\Git\1.3> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Git\1.3> |
```

Вернулся на ветку main

```
Администратор: Windows Pc X + v
илиность написания имени, а также наличие и правильность пути, после чего повторите попытку.
строка:1 знак:1
+ touch in_branch.txt
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (touch:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Git\1.3> git add in_branch.txt
PS C:\Git\1.3> git commit -m "Add in_branch.txt file"
[my_first_branch bd2860e] Add in_branch.txt file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 in_branch.txt
PS C:\Git\1.3> git push
fatal: The current branch my_first_branch has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin my_first_branch

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

PS C:\Git\1.3> git branch
main
* my_first_branch
PS C:\Git\1.3> git checkout master
error: pathspec 'master' did not match any file(s) known to git
PS C:\Git\1.3> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Git\1.3> git checkout -b new_branch
Switched to a new branch 'new_branch'
PS C:\Git\1.3> git branch
main
my_first_branch
* new_branch
PS C:\Git\1.3> git status
On branch new_branch
nothing to commit, working tree clean
PS C:\Git\1.3> git add 1.txt
PS C:\Git\1.3> 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")
PS C:\Git\1.3> git add 1.txt
PS C:\Git\1.3> git commit -m "Add new row to 1.txt file"
[new_branch e3a049a] Add new row to 1.txt file
1 file changed, 1 insertion(+)
PS C:\Git\1.3> git push
fatal: The current branch new_branch has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin new_branch

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

PS C:\Git\1.3> |
```

Сделал изменения в 1.txt

```
Администратор: Windows Pc
PS C:\Git\1.3> git add 1.txt
PS C:\Git\1.3> 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")
PS C:\Git\1.3> git add 1.txt
PS C:\Git\1.3> git commit -m "Add new row to 1.txt file"
[new_branch e3a049a] Add new row to 1.txt file
1 file changed, 1 insertion(+)
PS C:\Git\1.3> git push
fatal: The current branch new_branch has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin new_branch

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.
PS C:\Git\1.3> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Git\1.3> git merge my_first_branch
Updating 1a1c1fc..bd2860e
Fast-forward
 in_branch.txt | 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 in_branch.txt
PS C:\Git\1.3> git status
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
PS C:\Git\1.3> git merge new_branch
Merge made by the 'ort' strategy.
1.txt | 1 +
1 file changed, 1 insertion(+)
PS C:\Git\1.3> git status
On branch main
Your branch is ahead of 'origin/main' by 3 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
PS C:\Git\1.3> git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 700 bytes | 350.00 KiB/s, done.
Total 7 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To https://github.com/MononovDmitry/1.3.git
1a1c1fc..720ab16  main -> main
PS C:\Git\1.3>
```

## Слияние двух веток



```
Администратор: Windows Pc
Fast-forward
in_branch.txt | 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 in_branch.txt
PS C:\Git\1.3> git status
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
PS C:\Git\1.3> git merge new_branch
Merge made by the 'ort' strategy.
1.txt | 1 +
1 file changed, 1 insertion(+)
PS C:\Git\1.3> git status
On branch main
Your branch is ahead of 'origin/main' by 3 commits.
(use "git push" to publish your local commits)

nothing to commit, working tree clean
PS C:\Git\1.3> git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 700 bytes | 350.00 KiB/s, done.
Total 7 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To https://github.com/KononovDmitry/1.3.git
1alc1fc..720ab16 main -> main
PS C:\Git\1.3> git branch
* main
  my_first_branch
  new_branch
PS C:\Git\1.3> git checkout main
Already on 'main'
Your branch is up to date with 'origin/main'.
PS C:\Git\1.3> git branch -d my_first_branch
Deleted branch my_first_branch (was bd2860e).
PS C:\Git\1.3> git branch -d new_branch
Deleted branch new_branch (was e3a049a).
PS C:\Git\1.3> |
```

Удаление прошлых веток

```
Администратор: Windows Pc
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 in_branch.txt
PS C:\Git\1.3> git status
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
PS C:\Git\1.3> git merge new_branch
Merge made by the 'ort' strategy.
 1.txt | 1 +
 1 file changed, 1 insertion(+)
PS C:\Git\1.3> git status
On branch main
Your branch is ahead of 'origin/main' by 3 commits.
(use "git push" to publish your local commits)

nothing to commit, working tree clean
PS C:\Git\1.3> git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 700 bytes | 350.00 KiB/s, done.
Total 7 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To https://github.com/KononovDmitry/1.3.git
 1a1c1fc..720ab16 main -> main
PS C:\Git\1.3> git branch
* main
  my_first_branch
  new_branch
PS C:\Git\1.3> git checkout main
Already on 'main'
Your branch is up to date with 'origin/main'.
PS C:\Git\1.3> git branch -d my_first_branch
Deleted branch my_first_branch (was bd2860e).
PS C:\Git\1.3> git branch -d new_branch
Deleted branch new_branch (was e3a049a).
PS C:\Git\1.3> git branch branch_1
PS C:\Git\1.3> git branch branch_2
PS C:\Git\1.3>
```

Создал новые ветки

```
Администратор: Windows Pc X + v
Deleted branch my_first_branch (was bd2860e).
PS C:\Git\1.3> git branch -d new_branch
Deleted branch new_branch (was e3a049a).
PS C:\Git\1.3> git branch branch_1
PS C:\Git\1.3> git branch branch_2
PS C:\Git\1.3> git branch
  branch_1
  branch_2
* main
PS C:\Git\1.3> git checkout branch_1
Switched to branch 'branch_1'
PS C:\Git\1.3> 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
        modified:   3.txt

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Git\1.3> git add 1.txt 3.txt
PS C:\Git\1.3> git commit -m "Fix 1.txt and 3.txt files"
[branch_1 4bf0f56] Fix 1.txt and 3.txt files
 2 files changed, 2 insertions(+), 1 deletion(-)
PS C:\Git\1.3> 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

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

PS C:\Git\1.3> git status
On branch branch_1
nothing to commit, working tree clean
PS C:\Git\1.3> 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

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

PS C:\Git\1.3> git push --set-upstream origin branch_1
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 350 bytes | 350.00 KiB/s, done.
Total 4 (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_1' on GitHub by visiting:
remote:   https://github.com/KononovDmitry/1.3/pull/new/branch_1
remote:
To https://github.com/KononovDmitry/1.3.git
 * [new branch]   branch_1 -> branch_1
branch 'branch_1' set up to track 'origin/branch_1'.
PS C:\Git\1.3> |
```

Изменение файлов в branch\_1

```
Администратор: Windows Pc
[branch 1 af8f56] Fix 1.txt and 3.txt files
2 files changed, 2 insertions(+), 1 deletion(-)
PS C:\Git\1.3> 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

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

PS C:\Git\1.3> git status
On branch branch_1
nothing to commit, working tree clean
PS C:\Git\1.3> 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

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

PS C:\Git\1.3> git push --set-upstream origin branch_1
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 350 bytes | 350.00 KiB/s, done.
Total 4 (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_1' on GitHub by visiting:
remote:   https://github.com/KononovDmitry/1.3/pull/new/branch_1
remote:
To https://github.com/KononovDmitry/1.3.git
 * [new branch]      branch_1 -> branch_1
branch 'branch_1' set up to track 'origin/branch_1'.
PS C:\Git\1.3> git branch
* branch_1
  main
PS C:\Git\1.3> git checkout branch_2
Switched to branch 'branch_2'
PS C:\Git\1.3> 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
        modified:   3.txt

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Git\1.3> git add 1.txt 3.txt
PS C:\Git\1.3> git commit -m "My fixes in 1.txt and 3.txt files"
[branch_2 af81a9] My fixes in 1.txt and 3.txt files
2 files changed, 2 insertions(+), 1 deletion(-)
PS C:\Git\1.3> 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'.

PS C:\Git\1.3> git push --set-upstream origin branch_2
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 364 bytes | 182.00 KiB/s, done.
Total 4 (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/KononovDmitry/1.3/pull/new/branch_2
remote:
To https://github.com/KononovDmitry/1.3.git
 * [new branch]      branch_2 -> branch_2
branch 'branch_2' set up to track 'origin/branch_2'.
```

## Изменение файлов в branch\_2

```
Администратор: Windows Pc
Switched to branch 'branch_2'
PS C:\Git\1.3> 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
    modified:   3.txt

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Git\1.3> git add 1.txt 3.txt
PS C:\Git\1.3> git commit -m "My fixes in 1.txt and 3.txt files"
[branch_2 af81a9] My fixes in 1.txt and 3.txt files
2 files changed, 2 insertions(+), 1 deletion(-)
PS C:\Git\1.3> 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'.
PS C:\Git\1.3> git push --set-upstream origin branch_2
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 364 bytes | 182.00 KiB/s, done.
Total 4 (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/KononovDmitry/1.3/pull/new/branch_2
remote:
To https://github.com/KononovDmitry/1.3.git
 * [new branch]      branch_2 -> branch_2
branch 'branch_2' set up to track 'origin/branch_2'.
PS C:\Git\1.3> git branch
  branch_1
 * branch_2
   main
PS C:\Git\1.3> git pull
Already up to date.
PS C:\Git\1.3> git checkout branch_1
Switched to branch 'branch_1'
Your branch is up to date with 'origin/branch_1'.
PS C:\Git\1.3> git merge branch_2
It не удалось обработать преобразование аргументов для параметра "test". Не удалось преобразовать значение "branch_2" типа "System.String" в тип "System.Management.Automation.ScriptBlock".
cpovai1 anav:18
+ git merge branch_2
+
+ CategoryInfo          : InvalidData: (:) [It], ParameterBindingArgumentTransformationException
+ FullyQualifiedErrorId : ParameterArgumentTransformationError,It

PS C:\Git\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.
PS C:\Git\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)

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

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Git\1.3> git push
Everything up-to-date
PS C:\Git\1.3> |
```

## Слияние изменений

```
Администратор: Windows Pc X + v
Automatic merge failed; fix conflicts and then commit the result.
PS C:\Git\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)

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

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Git\1.3> git push
Everything up-to-date
PS C:\Git\1.3> git checkout branch_1
error: you need to resolve your current index first
1.txt: needs merge
3.txt: needs merge
PS C:\Git\1.3> git checkout branch_1
error: you need to resolve your current index first
1.txt: needs merge
3.txt: needs merge
PS C:\Git\1.3> git pull
error: Pulling is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
PS C:\Git\1.3> git checkout branch_2
error: you need to resolve your current index first
1.txt: needs merge
3.txt: needs merge
PS C:\Git\1.3> git add 1.txt
PS C:\Git\1.3> 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
Merging:
3.txt

Normal merge conflict for '3.txt':
  (local): modified file
  (remote): modified file
Hit return to start merge resolution tool (vimdiff):
PS C:\Git\1.3> git add 3.txt
PS C:\Git\1.3> git commit -m "fix 1 and 3"
[branch_1 76a730c] fix 1 and 3
PS C:\Git\1.3> git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 412 bytes | 412.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/NononovDmitry/1.3.git
  4bf0f56..76a730c branch_1 -> branch_1
PS C:\Git\1.3>
```

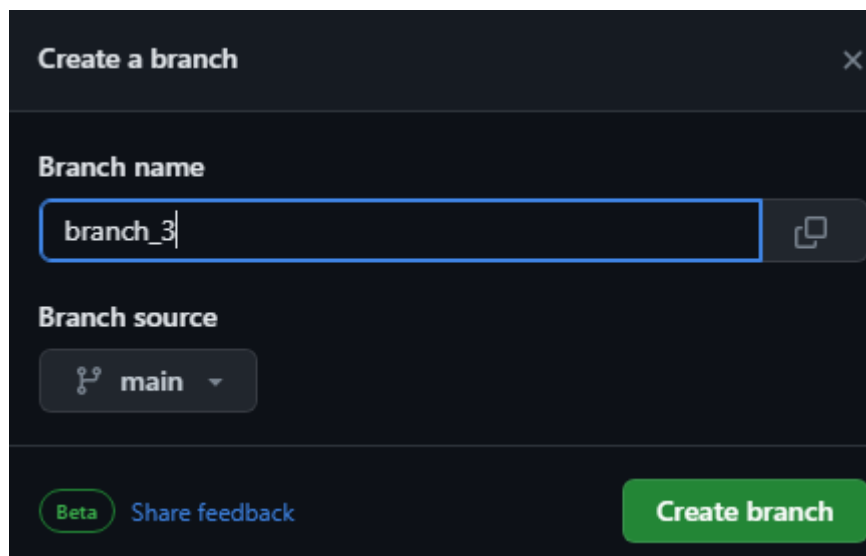
## Решение конфликтов

```
Администратор: Windows Pc
PS C:\Git\1.3> git pull
error: Pulling is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
PS C:\Git\1.3> git checkout branch_2
error: you need to resolve your current index first
1.txt: needs merge
3.txt: needs merge
PS C:\Git\1.3> git add 1.txt
PS C:\Git\1.3> 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
Merging:
3.txt

Normal merge conflict for '3.txt':
{local}: modified file
{remote}: modified file
Hit return to start merge resolution tool (vimdiff):
PS C:\Git\1.3> git add 3.txt
PS C:\Git\1.3> git commit -m "fix 1 and 3"
[branch_1 76a730c] fix 1 and 3
PS C:\Git\1.3> git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 412 bytes | 412.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/KononovDmitry/1.3.git
  4bf0f56..76a730c  branch_1 -> branch_1
PS C:\Git\1.3> git branch
* branch_1
  branch_2
  main
PS C:\Git\1.3> git pull
Already up to date.
PS C:\Git\1.3> git push -u origin branch_1
Everything up-to-date
branch 'branch_1' set up to track 'origin/branch_1'.
PS C:\Git\1.3>
```

Отправил ветку branch\_1



Создал branch\_3 на GitHub

```
Администратор: Windows Pc X + v
error: you need to resolve your current index first
1.txt: needs merge
3.txt: needs merge
PS C:\Git\1.3> git add 1.txt
PS C:\Git\1.3> 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
Merging:
3.txt

Normal merge conflict for '3.txt':
  (local): modified file
  (remote): modified file
Hit return to start merge resolution tool (vimdiff):
PS C:\Git\1.3> git add 3.txt
PS C:\Git\1.3> git commit -m "fix 1 and 3"
[branch_1 76a730c] fix 1 and 3
PS C:\Git\1.3> git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 412 bytes | 412.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/MononovDmitry/1.3.git
  4bf0f56..76a730c  branch_1 -> branch_1
PS C:\Git\1.3> git branch
* branch_1
  branch_2
  main
PS C:\Git\1.3> git pull
Already up to date.
PS C:\Git\1.3> git push -u origin branch_1
Everything up-to-date
branch 'branch_1' set up to track 'origin/branch_1'.
PS C:\Git\1.3> git bracnh
git: 'bracnh' is not a git command. See 'git --help'.

The most similar command is
  branch
PS C:\Git\1.3> git branch
* branch_1
  branch_2
  main
PS C:\Git\1.3> git fetch
From https://github.com/MononovDmitry/1.3
  * [new branch]      branch_3 -> origin/branch_3
PS C:\Git\1.3> git branch --track branch_3 origin/branch_3
branch 'branch_3' set up to track 'origin/branch_3'.
PS C:\Git\1.3> git checkout branch_3
Switched to branch 'branch_3'
Your branch is up to date with 'origin/branch_3'.
PS C:\Git\1.3> git push
Everything up-to-date
PS C:\Git\1.3> |
```

Создал ветку отслеживания в локальном репозитории



```
Администратор: Windows Pc
+ nano 2.txt
+ CategoryInfo          : ObjectNotFound: (nano:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Git\1.3> git status
On branch branch_3
Your branch is up to date with 'origin/branch_3'.

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:   2.txt

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Git\1.3> git add 2.txt
PS C:\Git\1.3> git commit -m "Add final fantasy line to 2.txt"
[branch_3 d055ae6] Add final fantasy line to 2.txt
 1 file changed, 1 insertion(+)
PS C:\Git\1.3> git commit -m "Add final fantasy line to 4.txt"
On branch branch_3
Your branch is ahead of 'origin/branch_3' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
PS C:\Git\1.3> git branch
  branch_1
  branch_2
* branch_3
  main
PS C:\Git\1.3> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Git\1.3> git merge branch_2
Updating 720ab16..af181a9
Fast-forward
 1.txt | 2 +-
 3.txt | 1 +
 2 files changed, 2 insertions(+), 1 deletion(-)
PS C:\Git\1.3> git branch -d branch_2
Deleted branch branch_2 (was af181a9).
PS C:\Git\1.3> |
```

Переместил ветку

```
Администратор: Windows Pc
PS C:\Git\1.3> git status
On branch branch_3
Your branch is up to date with 'origin/branch_3'.

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:   2.txt

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Git\1.3> git add 2.txt
PS C:\Git\1.3> git commit -m "Add final fantasy line to 2.txt"
[branch_3 d855ae6] Add final fantasy line to 2.txt
 1 file changed, 1 insertion(+)
PS C:\Git\1.3> git commit -m "Add final fantasy line to 4.txt"
On branch branch_3
Your branch is ahead of 'origin/branch_3' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
PS C:\Git\1.3> git branch
  branch_1
  branch_2
* branch_3
  main
PS C:\Git\1.3> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Git\1.3> git merge branch_2
Updating 728ab16..af181a9
Fast-forward
 1.txt | 2 +-
 3.txt | 1 +
 2 files changed, 2 insertions(+), 1 deletion(-)
PS C:\Git\1.3> git branch -d branch_2
Deleted branch branch_2 (was af181a9).
PS C:\Git\1.3> git branch
  branch_1
  branch_3
* main
PS C:\Git\1.3> git pull
Already up to date.
PS C:\Git\1.3> git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/KononovDmitry/1.3.git
 728ab16..af181a9  main -> main
PS C:\Git\1.3> git checkout branch_2
Switched to a new branch 'branch_2'
branch 'branch_2' set up to track 'origin/branch_2'.
PS C:\Git\1.3> git checkout branch_2
Already on 'branch_2'
Your branch is up to date with 'origin/branch_2'.
PS C:\Git\1.3> git pull
Already up to date.
PS C:\Git\1.3> git push
Everything up-to-date
PS C:\Git\1.3> |
```

Отправил изменения веток

## 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.

### 2. Что такое HEAD?

HEAD в Git-это указатель на текущую ссылку ветви, которая, в свою очередь, является указателем на последний сделанный вами коммит или последний коммит, который был извлечен из вашего рабочего каталога.

### 3. Способы создания веток.

Создать ветку можно с помощью двух команд. Команда, которая просто создает ветку: `git branch "name_branch"`.

Команда, которая создает ветку и сразу же к ней переходит: `git checkout -b "name_branch"`.

4. Как узнать текущую ветку?

Текущую ветку можно узнать с помощью команды: `git branch`.

5. Как переключаться между ветками?

Между ветками можно переключаться с помощью команды: `git checkout "name_branch"`.

6. Что такое удаленная ветка?

Удаленные ветки - это ссылки на определённое состояние удалённых веток. Это локальные ветки, которые нельзя перемещать.

7. Что такое ветка отслеживания?

Отслеживаемые ветки — это локальные ветки, которые напрямую связаны с удалённой веткой. Если, находясь на отслеживаемой ветке, вы наберёте `git push`, Git уже будет знать, на какой сервер и в какую ветку отправлять изменения.

8. Как создать ветку отслеживания?

Ветку отслеживания можно создать с помощью команды: `git checkout -- track origin/<name_branch>`.

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

Отправить изменения из локальной ветки в удаленную можно с помощью команды: `git push origin <name_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`.

11. Как удалить локальную и удаленную ветки?

Для удаление локальной ветки используется команда: `git branch -d <name_branch>`

Для удаления удаленной ветки используется команда: `git push --delete origin/<name_branch>`

12. Какие основные типы веток присутствуют в модели git-flow? Как организована работа с ветками в модели git-flow? В чем недостатки git-flow?

Существуют следующие типы ветвей:

- 1) ветви функциональностей;
- 2) ветви релизов;
- 3) ветви исправлений.

Ветви функциональностей (feature branches), также называемые иногда тематическими ветвями (topic branches), используются для разработки новых функций, которые должны появиться в текущем или будущем релизах.

Ветви релизов (release branches) используются для подготовки к выпуску новых версий продукта. Они позволяют расставить финальные точки над *i* перед выпуском новой версии.

Ветви для исправлений (hotfix branches) весьма похожи на ветви релизов (release branches), так как они тоже используются для подготовки новых выпусков продукта, разве лишь незапланированных.

Недостатки git flow: авторам приходится использовать ветку develop вместо master, поскольку master зарезервирован для кода.

Вторая проблема процесса git flow – сложности, возникающие из-за веток для патчей и для релиза. Подобная структура может подойти некоторым организациям, но для абсолютного большинства она просто убийственно излишняя.

**Вывод:** исследовал базовые возможности по работе с локальными и удаленными ветками Git.