

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 №1.2**  
**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:** Исследование возможностей Git для работы с локальными репозиториями

**Цель:** исследовать базовые возможности системы контроля версий Git для работы с локальными репозиториями

### Ход работы:

```
Администратор: Windows Pc X + v
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Установите последнюю версию PowerShell для новых функций и улучшения! https://aka.ms/PSWindows

PS C:\Git> git clone https://github.com/KononovDmitry/-1.1.git
Cloning into '-1.1'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (23/23), done.
Receiving objects: 100% (32/32), 775.80 KiB | 1.99 MiB/s, done.

Resolving deltas: 100% (10/10), done.
PS C:\Git> cd -1.1
PS C:\Git\-1.1> git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
PS C:\Git\-1.1> git log
commit afc22807d8688763716d214a35fd75e2cf4caec2 (HEAD -> main, origin/main, origin/HEAD)
Merge: c0014f3 5394845
Author: Dmitry <dkononov2903@gmail.com>
Date: Wed Mar 1 07:21:34 2023 +0300

Merge branch 'main' of https://github.com/KononovDmitry/-1.1

commit c0014f30f2b923960cc77c41506253ba53169690
Author: Dmitry <dkononov2903@gmail.com>
Date: Wed Mar 1 07:18:35 2023 +0300
```

### Комманда «git clone»

```
Администратор: Windows Pc X + v
PS C:\Git\-1.1> git log
commit afc22807d8688763716d214a35fd75e2cf4caec2 (HEAD -> main, origin/main, origin/HEAD)
Merge: c0014f3 5394845
Author: Dmitry <dkononov2903@gmail.com>
Date: Wed Mar 1 07:21:34 2023 +0300

Merge branch 'main' of https://github.com/KononovDmitry/-1.1

commit c0014f30f2b923960cc77c41506253ba53169690
Author: Dmitry <dkononov2903@gmail.com>
Date: Wed Mar 1 07:18:35 2023 +0300

Laba

commit 539484545f02f965ed085b380f0f75d8cc0e6cb6
Author: KononovDmitry <126630318+KononovDmitry@users.noreply.github.com>
Date: Wed Mar 1 07:13:28 2023 +0300

Delete g.gitignore

commit 9471a238d22eba489a778f35118de48c9f84b7be
Author: KononovDmitry <126630318+KononovDmitry@users.noreply.github.com>
Date: Wed Mar 1 07:13:17 2023 +0300

Delete .gitignore.txt

commit 7c709217ea232d859e9e5c36f29d4b1236217f79
Author: Dmitry <dkononov2903@gmail.com>
Date: Wed Mar 1 07:11:38 2023 +0300
```

### Комманда «git log»



```
Администратор: Windows Pc
Laba.pdf | Bin 0 -> 881514 bytes
1 file changed, 0 insertions(+), 0 deletions(-)

commit 539484545f02f965ed085b380f0f75d8cc0e6cb6
Author: KononovDmitry <126630318+KononovDmitry@users.noreply.github.com>
Date:   Wed Mar 1 07:13:28 2023 +0300

    Delete g.gitignore

g.gitignore | 185 -----
1 file changed, 185 deletions(-)

commit 9471a238d22eba489a778f35118de48c9f84b7be
Author: KononovDmitry <126630318+KononovDmitry@users.noreply.github.com>
Date:   Wed Mar 1 07:13:17 2023 +0300

PS C:\Git\~1.1> git log --pretty=format:"%h - %an, %ar : %s"
afc2280 - Dmitry, 2 weeks ago : Merge branch 'main' of https://github.com/KononovDmitry/~1.1
c0014f3 - Dmitry, 2 weeks ago : Laba
5394845 - KononovDmitry, 2 weeks ago : Delete g.gitignore
9471a23 - KononovDmitry, 2 weeks ago : Delete .gitignore.txt
7c70021 - Dmitry, 2 weeks ago : replace .gitignore
402dd37 - Dmitry, 2 weeks ago : add else
d726a6e - Dmitry, 2 weeks ago : add if
58334c3 - Dmitry, 2 weeks ago : add input
290f2f1 - Dmitry, 2 weeks ago : import library
a645c62 - Dmitry, 2 weeks ago : add gitignore
18fdbd8 - Dmitry, 2 weeks ago : about me 1
4b5e16a - Dmitry, 2 weeks ago : about me
PS C:\Git\~1.1>
```

Комманда «git log --pretty=format:"%h - %an, %ar : %s"»

```
Администратор: Windows Pc
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'
PS C:\Git\~1.1> git log --graph
fatal: ambiguous argument '--graph': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'
PS C:\Git\~1.1> git log --pretty=format:"%h %s" --graph
fatal: ambiguous argument '--graph': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'
PS C:\Git\~1.1> git log --pretty=format:"%h - %s" --graph
fatal: ambiguous argument '--graph': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'
PS C:\Git\~1.1> git log --pretty=format:"%h %s" --graph
*   afc2280 Merge branch 'main' of https://github.com/KononovDmitry/~1.1
|/
| * 5394845 Delete g.gitignore
| * 9471a23 Delete .gitignore.txt
| * c0014f3 Laba
|/
* 7c70021 replace .gitignore
* 402dd37 add else
* d726a6e add if
* 58334c3 add input
* 290f2f1 import library
* a645c62 add gitignore
* 18fdbd8 about me 1
* 4b5e16a about me
PS C:\Git\~1.1>
```

Комманда «git log --pretty=format:"%h %s" --graph»

```
Администратор: Windows Pc
fatal: ambiguous argument '--graph': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'
PS C:\Git\1.1> git log --pretty=format:"%h %s" --graph
fatal: ambiguous argument '--graph': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'
PS C:\Git\1.1> git log --pretty=format:"%h - %s" --graph
fatal: ambiguous argument '--graph': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'
PS C:\Git\1.1> git log --pretty=format:"%h %s" --graph
*   afc2280 Merge branch 'main' of https://github.com/KononovDmitry/-1.1
|
| * 5394845 Delete g.gitignore
| * 9471a23 Delete .gitignore.txt
| * c0014f3 Laba
|/
* 7c70021 replace .gitignore
* 402dd37 add else
* d726a6e add if
* 58334c3 add input
* 290f2f1 import library
* a645c62 add gitignore
* 18fdbd8 about me 1
* 4b5e16a about me
PS C:\Git\1.1> git log --since=30.minutes
PS C:\Git\1.1> git log --since=30.minutes
PS C:\Git\1.1> git log --since=2.weeks
PS C:\Git\1.1>
```

Комманда «git log --since=30.minutes»

```
Администратор: Windows Pc
PS C:\Git\1.1> git log --since=2.weeks
PS C:\Git\1.1> git log -s #include
commit afc22807d8688763716d214a35fd75e2cf4caec2 (HEAD -> main, origin/main, origin/HEAD)
Merge: c0014f3 5394845
Author: Dmitry <dkononov2903@gmail.com>
Date:   Wed Mar 1 07:21:34 2023 +0300

    Merge branch 'main' of https://github.com/KononovDmitry/-1.1

commit c0014f30f2b923960cc77c41506253ba53169690
Author: Dmitry <dkononov2903@gmail.com>
Date:   Wed Mar 1 07:18:35 2023 +0300

    Laba

commit 539484545f02f965ed085b380f0f75d8cc0e6cb6
Author: KononovDmitry <126630318+KononovDmitry@users.noreply.github.com>
Date:   Wed Mar 1 07:13:28 2023 +0300

    Delete g.gitignore

commit 9471a238d22eba489a778f35118de48c9f84b7be
Author: KononovDmitry <126630318+KononovDmitry@users.noreply.github.com>
Date:   Wed Mar 1 07:13:17 2023 +0300

    Delete .gitignore.txt

commit 7c700217ea232d859e9e5c36f29d4b1236217f79
Author: Dmitry <dkononov2903@gmail.com>
Date:   Wed Mar 1 07:11:38 2023 +0300
```

Комманда «git log -s #include»

```
Администратор: Windows Pc
PS C:\Git\1.1> git add README.md
PS C:\Git\1.1> git commit -m "read"
[main 5d0b210] read
1 file changed, 1 insertion(+)
PS C:\Git\1.1> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 363 bytes | 363.00 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/KononovDmitry/-1.1.git
   afc2280..5d0b210  main -> main
PS C:\Git\1.1> git add LICENSE
PS C:\Git\1.1> git commit -m "add license"
[main c2c50d5] add license
1 file changed, 21 insertions(+)
 create mode 100644 LICENSE
PS C:\Git\1.1> git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 895 bytes | 895.00 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/KononovDmitry/-1.1.git
   5d0b210..c2c50d5  main -> main
PS C:\Git\1.1> |
```

Добавил лицензию

```
Администратор: Windows Pc
5d0b210..c2c50d5  main -> main
PS C:\Git\1.1> git add prog.py
PS C:\Git\1.1> git commit -m "add print"
[main a3c7b92] add print
1 file changed, 1 insertion(+)
create mode 100644 prog.py
PS C:\Git\1.1> 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), 282 bytes | 282.00 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/KononovDmitry/-1.1.git
c2c50d5..a3c7b92  main -> main
PS C:\Git\1.1> git add prog.py
PS C:\Git\1.1> git commit -m "for X"
[main a5be906] for X
1 file changed, 2 insertions(+)
PS C:\Git\1.1> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 298 bytes | 298.00 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/KononovDmitry/-1.1.git
a3c7b92..a5be906  main -> main
PS C:\Git\1.1> git add prog.py
PS C:\Git\1.1> git commit -m "for Y"
[main 6c21fee] for Y
1 file changed, 2 insertions(+), 1 deletion(-)
PS C:\Git\1.1> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 307 bytes | 307.00 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/KononovDmitry/-1.1.git
a5be906..6c21fee  main -> main
PS C:\Git\1.1> git add prog.py
PS C:\Git\1.1> git commit -m "for Z"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
PS C:\Git\1.1> git add prog.py
```

```
Администратор: Windows Pc
PS C:\Git\1.1> git add prog.py
PS C:\Git\1.1> git commit -m "for W"
On branch main
Your branch is up to date with 'origin/main'.

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:   prog.py

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Git\1.1> git add prog.py
PS C:\Git\1.1> git commit -m "for W"
[main fb22e3e] for W
 1 file changed, 2 insertions(+), 1 deletion(-)
PS C:\Git\1.1> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 295 bytes | 295.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/KononovDmitry/-1.1.git
   d0d7607..fb22e3e  main -> main
PS C:\Git\1.1> git add prog.py
PS C:\Git\1.1> git commit -m "add if"
[main fd9d18b] add if
 1 file changed, 2 insertions(+), 1 deletion(-)
PS C:\Git\1.1> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 332 bytes | 332.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/KononovDmitry/-1.1.git
   fb22e3e..fd9d18b  main -> main
PS C:\Git\1.1> git add prog.py
PS C:\Git\1.1> git commit -m "add print and this end this program"
[main cceff7b] add print and this end this program
 1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Git\1.1> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 319 bytes | 319.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
```

Добавил программу



```
Администратор: Windows Pc
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 332 bytes | 332.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/KononovDmitry/-1.1.git
   fb22e3e..fd9d18b  main -> main
PS C:\Git\1.1> git add prog.py
PS C:\Git\1.1> git commit -m "add print and this end this program"
[main cceff7b] add print and this end this program
   1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Git\1.1> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 319 bytes | 319.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/KononovDmitry/-1.1.git
   fd9d18b..cceff7b  main -> main
PS C:\Git\1.1> git commit -m "-" -amend
error: pathspec '-amend' did not match any file(s) known to git
PS C:\Git\1.1> git add forgotten_file
fatal: pathspec 'forgotten_file' did not match any files
PS C:\Git\1.1> git commit --amend
hint: Waiting for your editor to close the file... unix2dos: converting file C:/Git/-1.1/.git/COMMIT_EDITMSG to DOS form
at...

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Wed Mar 15 13:25:09 2023 +0300
#
# On branch main
# Your branch is up to date with 'origin/main'.
#
# Changes to be committed:
#   modified:   prog.py
#
```

```
Администратор: Windows Pc
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 332 bytes | 332.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/KononovDmitry/-1.1.git
   fb22e3e..fd9d18b  main -> main
PS C:\Git\1.1> git add prog.py
PS C:\Git\1.1> git commit -m "add print and this end this program"
[main cceff7b] add print and this end this program
   1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Git\1.1> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 319 bytes | 319.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/KononovDmitry/-1.1.git
   fd9d18b..cceff7b  main -> main
PS C:\Git\1.1> git commit -m "-" -amend
error: pathspec '-amend' did not match any file(s) known to git
PS C:\Git\1.1> git add forgotten_file
fatal: pathspec 'forgotten_file' did not match any files
PS C:\Git\1.1> git commit --amend
hint: Waiting for your editor to close the file... unix2dos: converting file C:/Git/-1.1/.git/COMMIT_EDITMSG to DOS form
at...
dos2unix: converting file C:/Git/-1.1/.git/COMMIT_EDITMSG to Unix format...
[main 121317a] add print and this end this program
   Date: Wed Mar 15 13:25:09 2023 +0300
   1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Git\1.1>
```

Комманда «git commit –amend»

```
Администратор: Windows Pc
Writing objects: 100% (3/3), 315 bytes | 315.00 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/KononovDmitry/-1.1.git
   5af2490..b9f1648  main -> main
PS C:\Git\~1.1> git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
PS C:\Git\~1.1> git checkout -- prog.py
PS C:\Git\~1.1> git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
PS C:\Git\~1.1> git add prog.py
PS C:\Git\~1.1> git commit -m "code delete"
[main 5c0a9d2] code delete
   1 file changed, 2 insertions(+), 1 deletion(-)
PS C:\Git\~1.1> 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\~1.1> git reset HEAD prog.py
PS C:\Git\~1.1> 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\~1.1> |
```

## Комманда «git reset HEAD»

```
Администратор: Windows Pc
origin https://github.com/KononovDmitry/-1.1.git (fetch)
origin https://github.com/KononovDmitry/-1.1.git (push)
PS C:\Git\~1.1> git remote -v
origin https://github.com/KononovDmitry/-1.1.git (fetch)
origin https://github.com/KononovDmitry/-1.1.git (push)
PS C:\Git\~1.1> git add pb https://github.com/KononovDmitry/rabota.git
fatal: pathspec 'pb' did not match any files
PS C:\Git\~1.1> git add pb https://github.com/KononovDmitry/rabota
fatal: pathspec 'pb' did not match any files
PS C:\Git\~1.1> git clone https://github.com/KononovDmitry/rabota.git
Cloning into 'rabota'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
PS C:\Git\~1.1> git remote -v
origin https://github.com/KononovDmitry/-1.1.git (fetch)
origin https://github.com/KononovDmitry/-1.1.git (push)
PS C:\Git\~1.1> git remote add pb https://github.com/KononovDmitry/rabota
PS C:\Git\~1.1> git remote -v
origin https://github.com/KononovDmitry/-1.1.git (fetch)
origin https://github.com/KononovDmitry/-1.1.git (push)
pb      https://github.com/KononovDmitry/rabota (fetch)
pb      https://github.com/KononovDmitry/rabota (push)
PS C:\Git\~1.1> git fetch pb
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 1.19 KiB | 21.00 KiB/s, done.
From https://github.com/KononovDmitry/rabota
 * [new branch]      main      -> pb/main
PS C:\Git\~1.1> |
```

## Комманда «git remote -v»

```
Администратор: Windows Pc
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
PS C:\Git\1.1> git remote -v
origin https://github.com/KononovDmitry/-1.1.git (fetch)
origin https://github.com/KononovDmitry/-1.1.git (push)
PS C:\Git\1.1> git remote add pb https://github.com/KononovDmitry/rabota
PS C:\Git\1.1> git remote -v
origin https://github.com/KononovDmitry/-1.1.git (fetch)
origin https://github.com/KononovDmitry/-1.1.git (push)
pb https://github.com/KononovDmitry/rabota (fetch)
pb https://github.com/KononovDmitry/rabota (push)
PS C:\Git\1.1> git fetch pb
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 1.19 KiB | 21.00 KiB/s, done.
From https://github.com/KononovDmitry/rabota
 * [new branch]      main      -> pb/main
PS C:\Git\1.1> git remote show origin
* remote origin
  Fetch URL: https://github.com/KononovDmitry/-1.1.git
  Push URL:  https://github.com/KononovDmitry/-1.1.git
  HEAD branch: main
  Remote branch:
    main tracked
  Local branch configured for 'git pull':
    main merges with remote main
  Local ref configured for 'git push':
    main pushes to main (up to date)
PS C:\Git\1.1> |
```

Комманда «git remote show origin»

```
Администратор: Windows Pc
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
PS C:\Git\1.1> git remote -v
origin https://github.com/KononovDmitry/-1.1.git (fetch)
origin https://github.com/KononovDmitry/-1.1.git (push)
PS C:\Git\1.1> git remote add pb https://github.com/KononovDmitry/rabota
PS C:\Git\1.1> git remote -v
origin https://github.com/KononovDmitry/-1.1.git (fetch)
origin https://github.com/KononovDmitry/-1.1.git (push)
pb https://github.com/KononovDmitry/rabota (fetch)
pb https://github.com/KononovDmitry/rabota (push)
PS C:\Git\1.1> git fetch pb
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 1.19 KiB | 21.00 KiB/s, done.
From https://github.com/KononovDmitry/rabota
 * [new branch]      main      -> pb/main
PS C:\Git\1.1> git remote show origin
* remote origin
  Fetch URL: https://github.com/KononovDmitry/-1.1.git
  Push URL:  https://github.com/KononovDmitry/-1.1.git
  HEAD branch: main
  Remote branch:
    main tracked
  Local branch configured for 'git pull':
    main merges with remote main
  Local ref configured for 'git push':
    main pushes to main (up to date)
PS C:\Git\1.1> git remote rename pb rab
Renaming remote references: 100% (1/1), done.
PS C:\Git\1.1> |
```

Комманда «git remote rename»

```
Администратор: Windows Pc
PS C:\Git\1.1> git remote -v
origin https://github.com/KononovDmitry/-1.1.git (fetch)
origin https://github.com/KononovDmitry/-1.1.git (push)
PS C:\Git\1.1> git remote add pb https://github.com/KononovDmitry/rabota
PS C:\Git\1.1> git remote -v
origin https://github.com/KononovDmitry/-1.1.git (fetch)
origin https://github.com/KononovDmitry/-1.1.git (push)
pb https://github.com/KononovDmitry/rabota (fetch)
pb https://github.com/KononovDmitry/rabota (push)
PS C:\Git\1.1> git fetch pb
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 1.19 KiB | 21.00 KiB/s, done.
From https://github.com/KononovDmitry/rabota
* [new branch]      main      -> pb/main
PS C:\Git\1.1> git remote show origin
* remote origin
Fetch URL: https://github.com/KononovDmitry/-1.1.git
Push URL: https://github.com/KononovDmitry/-1.1.git
HEAD branch: main
Remote branch:
main tracked
Local branch configured for 'git pull':
main merges with remote main
Local ref configured for 'git push':
main pushes to main (up to date)
PS C:\Git\1.1> git remote rename pb rab
Renaming remote references: 100% (1/1), done.
PS C:\Git\1.1> git remote remove rab
PS C:\Git\1.1> git remote
origin
PS C:\Git\1.1> |
```

## Комманда «git remote remove»

```
Администратор: Windows Pc
origin
PS C:\Git\1.1> git tag
PS C:\Git\1.1> v1.0
v1.0 : Имя "v1.0" 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
+ v1.0
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (v1.0:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Git\1.1> git tag -l "v1.0"
PS C:\Git\1.1> git tag
PS C:\Git\1.1> git tag list
PS C:\Git\1.1> git push
Everything up-to-date
PS C:\Git\1.1> git tag
list
PS C:\Git\1.1> git tag -d list
Deleted tag 'list' (was 08282e3)
PS C:\Git\1.1> git tag v1.0
PS C:\Git\1.1> git tag
v1.0
PS C:\Git\1.1> git tag v1.1
PS C:\Git\1.1> git tag v1.2
PS C:\Git\1.1> git tag v1.3
PS C:\Git\1.1> git tag v1.4
PS C:\Git\1.1> git tag
v1.0
v1.1
v1.2
v1.3
v1.4
PS C:\Git\1.1> |
```

## Комманда «git tag»

```
Администратор: Windows Pc
PS C:\Git\~1.1> git tag -l "v1.0"
PS C:\Git\~1.1> git tag
PS C:\Git\~1.1> git tag list
PS C:\Git\~1.1> git push
Everything up-to-date
PS C:\Git\~1.1> git tag
list
PS C:\Git\~1.1> git tag -d list
Deleted tag 'list' (was 08282e3)
PS C:\Git\~1.1> git tag v1.0
PS C:\Git\~1.1> git tag
v1.0
PS C:\Git\~1.1> git tag v1.1
PS C:\Git\~1.1> git tag v1.2
PS C:\Git\~1.1> git tag v1.3
PS C:\Git\~1.1> git tag v1.4
PS C:\Git\~1.1> git tag
v1.0
v1.1
v1.2
v1.3
v1.4
PS C:\Git\~1.1> git tag -a v1.4 -m "my version 1.4"
fatal: tag 'v1.4' already exists
PS C:\Git\~1.1> git tag -a v1.5 -m "my version 1.5"
PS C:\Git\~1.1> git tag
v1.0
v1.1
v1.2
v1.3
v1.4
v1.5
PS C:\Git\~1.1> |
```

Комманда «git tag -a v1.5 -m "my version 1.5"»

```
Администратор: Windows Pc
error: failed to push some refs to 'https://github.com/KononovDmitry/~1.1.git'
PS C:\Git\~1.1> git push
Everything up-to-date
PS C:\Git\~1.1> git push origin -tags
error: src refspec -tags does not match any
error: failed to push some refs to 'https://github.com/KononovDmitry/~1.1.git'
PS C:\Git\~1.1> git push origin --tags
error: src refspec --tags does not match any
error: failed to push some refs to 'https://github.com/KononovDmitry/~1.1.git'
PS C:\Git\~1.1> git tag
v1.0
v1.1
v1.2
v1.3
v1.4
v1.5
PS C:\Git\~1.1> git pust --tags
git: 'pust' is not a git command. See 'git --help'.

The most similar command is
push
PS C:\Git\~1.1> git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 168 bytes | 84.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/KononovDmitry/~1.1.git
* [new tag]          v1.0 -> v1.0
* [new tag]          v1.1 -> v1.1
* [new tag]          v1.2 -> v1.2
* [new tag]          v1.3 -> v1.3
* [new tag]          v1.4 -> v1.4
* [new tag]          v1.5 -> v1.5
PS C:\Git\~1.1> |
```

Комманда «git push -tags»

```
Администратор: Windows Pc
The most similar command is
push
PS C:\Git\~1.1> git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 168 bytes | 84.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/KononovDmitry/~1.1.git
* [new tag]          v1.0 -> v1.0
* [new tag]          v1.1 -> v1.1
* [new tag]          v1.2 -> v1.2
* [new tag]          v1.3 -> v1.3
* [new tag]          v1.4 -> v1.4
* [new tag]          v1.5 -> v1.5
PS C:\Git\~1.1> git checkout v1.1
Note: switching to 'v1.1'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 08282e3 change
PS C:\Git\~1.1>
```

## Комманда «git checkout v1.1»

### Ответы на контрольные вопросы:

1. За просмотр истории коммитов отвечает команда `git log`. В сочетании с различными параметрами эта команда выводит историю по-разному.

`git log -p`, расширенный вывод истории,  
`git log --oneline`, короткая запись,  
`git log --stat --graph`, история в виде дерева

2. Кроме опций для форматирования вывода, `git log` имеет ряд полезных ограничительных параметров, то есть параметров, которые дают возможность отобразить часть коммитов. Параметр `-2`, который отображает только два последних коммита. На самом деле, вы можете задать `-<n>`, где `n` это количество отображаемых коммитов.

А вот параметры, ограничивающие по времени, такие как `--since` и `--until`, весьма полезны. Например, следующая команда выдаёт список коммитов, сделанных за последние две недели:

```
$ git log --since=2.weeks
```

3. Если вы хотите изменить только сообщение вашего последнего коммита, это очень просто:

```
$ git commit --amend
```

Эта команда откроет в вашем текстовом редакторе сообщение вашего последнего коммита для того, чтобы вы могли его исправить. Когда вы сохраните его и закроете редактор, будет создан новый коммит, содержащий это сообщение, который теперь и будет вашим последним коммитом.

Если вы создали коммит и затем хотите изменить зафиксированный снимок, добавив или изменив файлы (возможно, вы забыли добавить вновь созданный файл, когда совершали изначальный коммит), то процесс выглядит в основном так же. Вы добавляете в индекс необходимые изменения, редактируя файл и выполняя для него `git add` или `git rm` для отслеживаемого файла, а последующая команда `git commit --amend` берет вашу текущую область подготовленных изменений и делает её снимок для нового коммита.

4. Отмена может потребоваться, если вы сделали коммит слишком рано, например, забыв добавить какие-то файлы или комментарий к коммиту. Если вы хотите переделать коммит — внесите необходимые изменения, добавьте их в индекс и сделайте коммит ещё раз, указав параметр `--amend`:

```
$ git commit --amend
```

5. Прямо под текстом «Changes to be committed» говорится: используйте `git reset HEAD <file>...` для исключения из индекса.

```
$ git reset HEAD CONTRIBUTING.md
```

Unstaged changes after reset:

```
M    CONTRIBUTING.md
```

```
$ git status
```

On branch master Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

```
renamed:    README.md -> README
```

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

modified: CONTRIBUTING.md

6. Распределённый репозиторий — это, фактически, удалённая копия вашей программы, которую вы делаете с помощью git (или другой системы контроля версий) на удалённом сервере, доступном через интернет. После этого все, кому вы дадите доступ (в т.ч. и вы сами), смогут скачивать вашу программу, вносить в неё изменения и загружать их на удалённый сервер с помощью всё того же git.

7. Для того, чтобы просмотреть список настроенных удалённых репозиториях, вы можете запустить команду git remote. Она выведет названия доступных удалённых репозиториях. Если вы клонировали репозиторий, то увидите как минимум origin — имя по умолчанию, которое Git даёт серверу, с которого производилось клонирование

8. Чтобы добавить новый удалённый репозиторий, выполните команду git remote add в терминале в каталоге, в котором хранится репозиторий.

Команда git remote add принимает два аргумента:

имя удалённого репозитория, например, origin;

URL-адрес удалённого репозитория,

9. \$ git fetch [remote-name]

Данная команда связывается с указанным удалённым проектом и забирает все те данные проекта, которых у вас ещё нет. После того как вы выполнили команду, у вас должны появиться ссылки на все ветки из этого



удалённого проекта, которые вы можете просмотреть или слить в любой момент.

Когда вы хотите поделиться своими наработками, вам необходимо отправить их в удалённый репозиторий. Команда для этого действия простая: `git push <remote-name> <branch-name>`. Чтобы отправить вашу ветку `master` на сервер `origin` (повторимся, что клонирование обычно настраивает оба этих имени автоматически), вы можете выполнить следующую команду для отправки ваших коммитов:

```
$ git push origin master
```

10. Для того, чтобы просмотреть список настроенных удалённых репозиториев, вы можете запустить команду `git remote`. Она выведет названия доступных удалённых репозиториев. Если вы клонировали репозиторий, то увидите как минимум `origin` — имя по умолчанию, которое Git даёт серверу, с которого производилось клонирование:

11. «Git» теги — это метки, которыми можно пометить коммиты в истории коммитов, хранящихся в Git-репозитории. Часто тегами помечают коммиты, после которых выходит очередной релиз компьютерной программы, если речь идет о проекте написания некой компьютерной программы.

12. Git использует два основных типа тегов: легковесные и аннотированные.

Создание аннотированного тега в Git выполняется легко. Самый простой способ — это указать `-a` при выполнении команды `tag`:

Легковесный тег — это ещё один способ пометить коммит. По сути, это контрольная сумма коммита, сохранённая в файл — больше никакой информации не хранится. Для создания легковесного тега не передавайте опций `-a`, `-s` и `-m`, укажите только название:

13. Это запускает `git fsck --unreachable`, используя все ссылки, доступные `vrefs/`, необязательно с дополнительным набором объектов, указанных в командной строке, и удаляет все распакованные объекты, недоступные для любого из этих головных объектов, из базы данных объектов.

Кроме того, он удаляет распакованные объекты, которые также находятся в пакетах, запустив `git prune-packed` . Он также удаляет записи из `.git/shallow`, которые недоступны ни по одной ссылке.

**Вывод:** исследовал базовые возможности системы контроля версий Git для работы с локальными репозиториями.