

ДИСЦИПЛИНА	ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНЫХ ПРИЛОЖЕНИЙ (полное наименование дисциплины без сокращений)
ИНСТИТУТ	ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
КАФЕДРА	МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ И СТАНДАРТИЗАЦИИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ (полное наименование кафедры)
ВИД УЧЕБНОГО МАТЕРИАЛА	МАТЕРИАЛЫ ДЛЯ ПРАКТИЧЕСКИХ/СЕМИНАРСКИХ ЗАНЯТИЙ (в соответствии с пп.1-11)
ПРЕПОДАВАТЕЛЬ	ЖМАТОВ ДМИТРИЙ ВЛАДИМИРОВИЧ, МЕЛЬНИКОВ ДЕНИС АЛЕКСАНДРОВИЧ, МИРОНОВ АЛЕКСЕЙ ИГОРЕВИЧ, ОВЧИННИКОВ МИХАИЛ АНДРЕЕВИЧ, ПЕТРЕНКО АЛЕКСАНДР АНАТОЛЬЕВИЧ, ПЕТРОВА АННА АЛЕКСАНДРОВНА, СНЕГИРЕВ ДМИТРИЙ АЛЕКСЕЕВИЧ, ТУМАНОВА МАРИНА БОРИСОВНА, ХЛЕБНИКОВА ВАЛЕРИЯ ЛЕОНИДОВНА, ЧЕРНОВ ЕВГЕНИЙ АЛЕКСАНДРОВИЧ (фамилия, имя, отчество)
СЕМЕСТР	4, 2023-2024 (указать семестр обучения, учебный год)

Практическая работа №1

Арефьев Александр Михайлович

ИКБО-04-22

ОГЛАВЛЕНИЕ

ЗАДАНИЕ.....	3
Цель работы:.....	3
Часть 1. Основные команды Git 1.....	3
Часть 2. Системы управления репозиториями.....	7
Часть 3. Работа с ветвлением и оформление кода.....	11
Ответы на контрольные вопросы.....	18
Вывод.....	20

ЗАДАНИЕ

Цель работы:

Получить навыки по работе с командной строкой и git'ом. Ответить на контрольные вопросы.

Часть 1. Основные команды Git 1.

1. Установите и настройте клиент git на своей рабочей станции.
2. Создайте локальный репозиторий и добавьте в него несколько файлов. Внесите изменения в один из файлов.
3. Проиндексируйте изменения и проверьте состояние.
4. Сделайте коммит того, что было проиндексировано в репозиторий. Добавьте к коммиту комментарий.
5. Измените еще один файл. Добавьте это изменение в индекс git. Измените файл еще раз. Проверьте состояние и произведите коммит проиндексированного изменения. Теперь добавьте второе изменение в индекс, а затем проверьте состояние с помощью команды `git status`. Сделайте коммит второго изменения.
6. Просмотрите историю коммитов с помощью команды `git log`. Ознакомьтесь с параметрами команды и используйте некоторые из них для различного формата отображения истории коммитов.

7. Верните рабочий каталог к одному из предыдущих состояний.
8. Изучите, как создавать теги для коммитов для использования в будущем.
9. Отмените некоторые изменения в рабочем каталоге (до и после индексирования).
10. Отмените один из коммитов в локальной репозитории.

```
❏ ~ cd repos
❏ ~/repos mkdir new_rep
❏ ~/repos cd new_rep
❏ ~/repos/new_rep git init
подсказка: Using 'master' as the name for the initial branch. This default branch name
подсказка: is subject to change. To configure the initial branch name to use in all
подсказка: of your new repositories, which will suppress this warning, call:
подсказка:
подсказка:     git config --global init.defaultBranch <name>
подсказка:
подсказка: Names commonly chosen instead of 'master' are 'main', 'trunk' and
подсказка: 'development'. The just-created branch can be renamed via this command:
подсказка:
подсказка:     git branch -m <name>
Инициализирован пустой репозиторий Git в /home/merc/repos/new_rep/.git/
❏ ~/repos/new_rep git P master touch new_file.txt
❏ ~/repos/new_rep git P master 71 git add .
❏ ~/repos/new_rep git P master +1 git add new_file.txt
❏ ~/repos/new_rep git P master +1 git status
Текущая ветка: master

Еще нет коммитов

Изменения, которые будут включены в коммит:
(используйте «git rm --cached <файл>...», чтобы убрать из индекса)
    новый файл:    new_file.txt

❏ ~/repos/new_rep git P master +1 git commit -m "hello new_file"
[master (корневой коммит) e4313e7] hello new_file
1 file changed, 1 insertion(+)
create mode 100644 new_file.txt
❏ ~/repos/new_rep git P master
```

Рисунок 1 — Задача 1-4

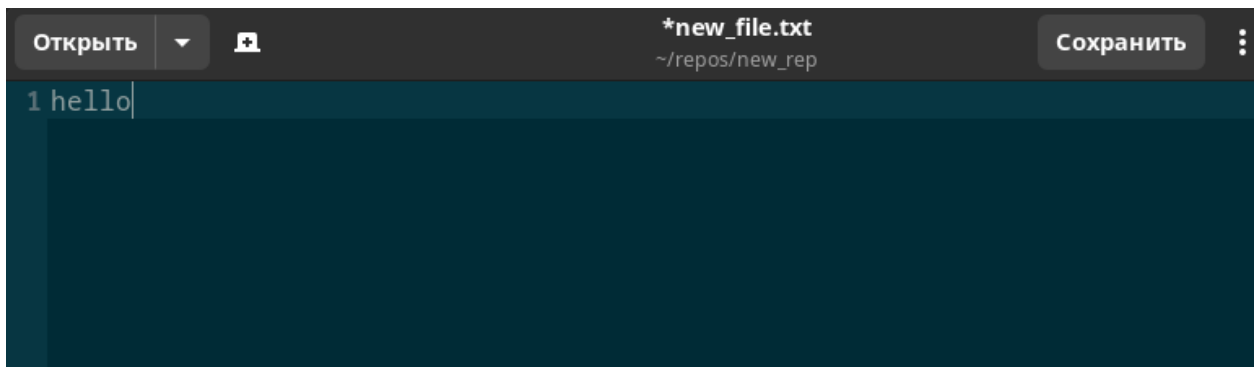


Рисунок 2 — Файл который был изменен между git add

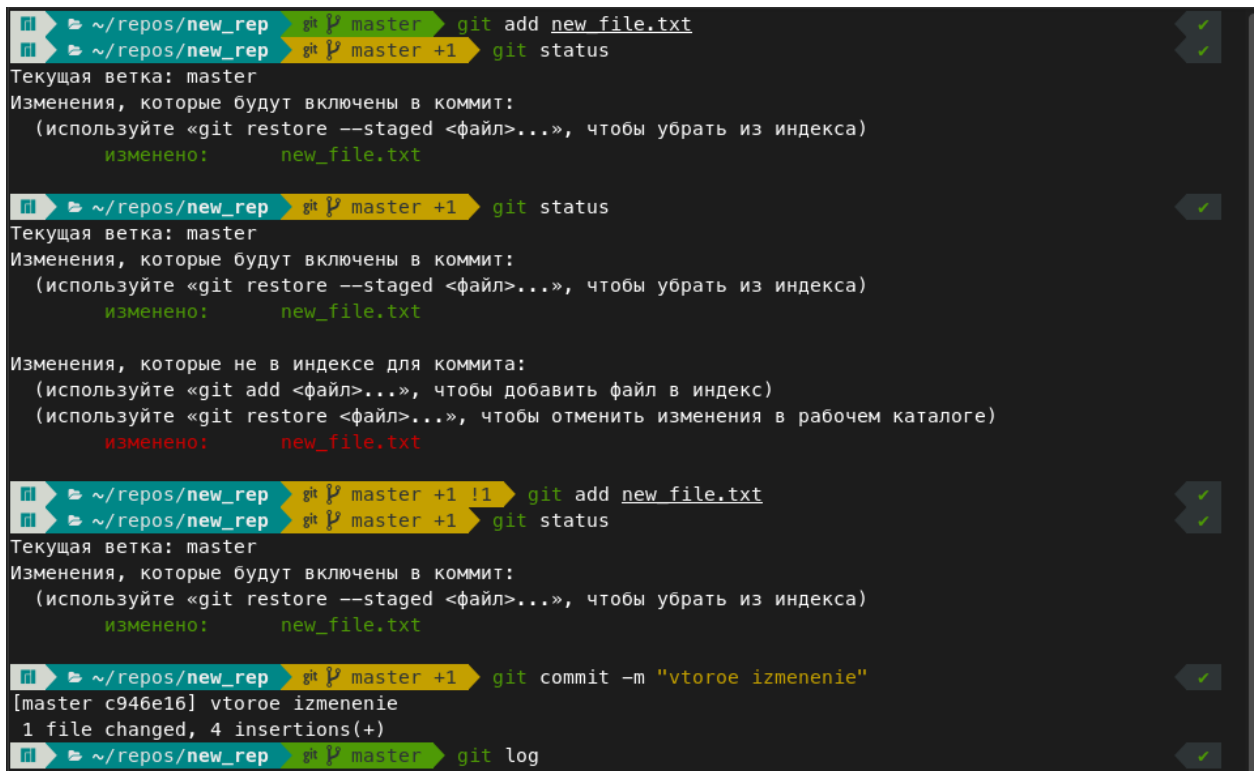


Рисунок 3 — Задачи 5-6

```
git log
commit c946e167a8ce53e3f39d96bc01d0777e8dc5ead4 (HEAD -> master)
Author: Arefiev Alexandr <alexanderarefievrus@gmail.com>
Date: Sat Mar 2 20:51:48 2024 +0300

    vtoroe izmenenie

commit e4313e7d3e3abf8b66949b4ed1881fdd08213541
Author: Arefiev Alexandr <alexanderarefievrus@gmail.com>
Date: Sat Mar 2 20:43:21 2024 +0300


    hello new_file

(END)
```

Рисунок 4 — Отображение git log

Открыть

▼



new_file.txt
~/repos/new_rep

```
1 hello
2
3 new change
4
5 new new change
```

Рисунок 5 — Файл над которым проводились изменения

```
~/repos/new_rep master git checkout c946e167a8ce53e3f39d96bc01d0777e8dc5ead4
Примечание: переключение на «c946e167a8ce53e3f39d96bc01d0777e8dc5ead4».

Вы сейчас в состоянии «отсоединённого указателя HEAD». Можете осмотреться,
внести экспериментальные изменения и зафиксировать их, также можете
отменить любые коммиты, созданные в этом состоянии, не затрагивая другие
ветки, переключившись обратно на любую ветку.

Если хотите создать новую ветку для сохранения созданных коммитов, можете
сделать это (сейчас или позже), используя команду switch с параметром -с.
Например:

git switch -с <новая-ветка>

Или отмените эту операцию с помощью:

git switch -

Отключите этот совет, установив переменную конфигурации
advice.detachedHead в значение false

HEAD сейчас на c946e16 vtoroe izmenenie
~/repos/new_rep @c946e167 git tag -a New_tag c946e167a8ce53e3f39d96bc01d0777e8dc5ead4 -m "Н
вучился."
~/repos/new_rep #New_tag
~/repos/new_rep #New_tag git checkout -- new_file.txt
~/repos/new_rep #New_tag git add new_file.txt
~/repos/new_rep #New_tag +1 git reset HEAD new_file.txt
Непроеиндексированные изменения после сброса:
M   new_file.txt
~/repos/new_rep #New_tag !1 git checkout -- new_file.txt
~/repos/new_rep #New_tag git reset --hard HEAD~1
Указатель HEAD сейчас на коммите e4313e7 hello new_file
~/repos/new_rep @e4313e7
```

Рисунок 6 — Задачи 7-10

Часть 2. Системы управления репозиториями

1. Создайте аккаунт на GitHub (у кого нет),
2. Создайте репозиторий на GitHub и на локальной машине, согласно выбранной теме проекта,
3. Создайте несколько файлов на локальной машине при помощи консоли,
4. Создайте SSH-ключ для авторизации,
5. Свяжите репозиторий локальной машины с репозиторием на GitHub при помощи консоли,
6. Создайте новую ветку в репозитории с помощью команды,

произведите в ней какие-нибудь изменения, а после слейте с веткой master,

7. Выполните цепочку действий в репозитории, согласно вариантам.

Таблица 1 — Индивидуальный вариант

№ варианта	Задание
1	<ol style="list-style-type: none">1) Клонировать непустой удаленный репозиторий на локальную машину2) Создайте новую ветку и выведите список всех веток3) Произведите 3 коммита в новой ветке4) Выгрузите изменения в удаленный репозиторий5) Откатите ветку обратно на 2 коммита (в том числе в удаленном репозитории)6) Выведите в консоли различия между веткой master и новой веткой7) Слейте новую ветку с master при помощи merge


```
~/repos git init "proekt"
подсказка: Using 'master' as the name for the initial branch. This default branch name
подсказка: is subject to change. To configure the initial branch name to use in all
подсказка: of your new repositories, which will suppress this warning, call:
подсказка:
подсказка:     git config --global init.defaultBranch <name>
подсказка:
подсказка: Names commonly chosen instead of 'master' are 'main', 'trunk' and
подсказка: 'development'. The just-created branch can be renamed via this command:
подсказка:
подсказка:     git branch -m <name>
Инициализирован пустой репозиторий Git в /home/merc/repos/proekt/.git/
~/repos cd proekt
~/repos/proekt git P master echo "new_file" > new_file.txt
~/repos/proekt git P master ?1 echo "new_another_file" > new_file2.txt
~/repos/proekt git P master ?2 ls
new_file2.txt new_file.txt
~/repos/proekt git P master ?2 ssh-keygen -t rsa -b 2048 -C "hello_project_ssh_key"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/merc/.ssh/id_rsa):
Created directory '/home/merc/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/merc/.ssh/id_rsa
Your public key has been saved in /home/merc/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Gk1+DEG23ZfyIY7PxGqEWno8xr3NveJ7i5Y/0mAUnAw hello_project_ssh_key
The key's randomart image is:
+---[RSA 2048]-----+
|      .+ E+ .      |
|      . + . = .    |
|      + . +. +     |
|      + + +. = .    |
|      . S =. + .    |
|      0 + =0        |
|      + * +.0+      |
|      0 0 +=0+      |
|      .0**=+        |
+---[SHA256]-----+
~/repos/proekt git P master ?2 git remote add origin git@github.com:MercurIc/proekt.git
```

Рисунок 8 — Задачи 1 — 5

```
~/repos ssh -T git@github.com
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvC0qU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Enter passphrase for key '/home/merc/.ssh/id_rsa':
Enter passphrase for key '/home/merc/.ssh/id_rsa':
Enter passphrase for key '/home/merc/.ssh/id_rsa':
Hi MercurIc! You've successfully authenticated, but GitHub does not provide shell access.
```

Рисунок 8.1 — SSH ключ

```
~/repos/proekt git P master ?2 git add .
~/repos/proekt git P master +2 git commit -m "first commit"
[master (корневой коммит) ef72588] first commit
2 files changed, 2 insertions(+)
create mode 100644 new_file.txt
create mode 100644 new_file2.txt
~/repos/proekt git P master git branch new-branch
~/repos/proekt git P master git checkout new-branch
Переключились на ветку «new-branch»
~/repos/proekt git P new-branch echo "hello_git" > new_file.txt
~/r/proekt git P new-branch !1 git add .
~/r/proekt git P new-branch +1 git commit -m "commit in new branch"
[new-branch c1c70fd] commit in new branch
1 file changed, 1 insertion(+), 1 deletion(-)
~/repos/proekt git P new-branch git checkout master
Переключились на ветку «master»
~/repos/proekt git P master git merge new-branch
Обновление ef72588..c1c70fd
Fast-forward
 new_file.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
~/repos/proekt git P master
```

Рисунок 9 — Задача 6

```
~/repos/Coffeemachine main git checkout -b branch2
Переключились на новую ветку «branch2»
~/repos/Coffeemachine branch2 git branch

zsh: suspended git branch
~/repos/Coffeemachine branch2 echo "c2" > file1.txt
~/repos/Coffeemachine branch2 git add .
~/repos/Coffeemachine branch2 git commit -m "c1"
[branch2 312e939] c1
1 file changed, 1 insertion(+)
create mode 100644 file1.txt
~/repos/Coffeemachine branch2 echo "c2" > file2.txt
~/repos/Coffeemachine branch2 git add .
~/repos/Coffeemachine branch2 git commit -m "c2"
[branch2 005ce0e] c2
1 file changed, 1 insertion(+)
create mode 100644 file2.txt
~/repos/Coffeemachine branch2 echo "c3" > file3.txt
~/repos/Coffeemachine branch2 git add .
~/repos/Coffeemachine branch2 git commit -m "c3"
[branch2 782b420] c3
1 file changed, 1 insertion(+)
create mode 100644 file3.txt
~/repos/Coffeemachine branch2 git push origin branch2
Username for 'https://github.com': Mercurlc
Password for 'https://Mercurlc@github.com':
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 12 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (8/8), 656 байтов | 656.00 КиБ/с, готово.
Всего 8 (изменений 2), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (2/2), done.
remote:
remote: Create a pull request for 'branch2' on GitHub by visiting:
remote:   https://github.com/Mercurlc/Coffeemachine/pull/new/branch2
remote:
To https://github.com/Mercurlc/Coffeemachine.git
 * [new branch] branch2 -> branch2
~/repos/Coffeemachine branch2 git reset --hard HEAD~2
Указатель HEAD сейчас на коммита 312e939 c1
~/repos/Coffeemachine branch2 git push origin branch2 --force
Username for 'https://github.com': Mercurlc
Password for 'https://Mercurlc@github.com':
Всего 0 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To https://github.com/Mercurlc/Coffeemachine.git
+ 782b420...312e939 branch2 -> branch2 (forced update)
~/repos/Coffeemachine branch2 git diff main -- branch2

zsh: suspended git diff main -- branch2
~/repos/Coffeemachine branch2 git checkout main
Переключились на ветку «main»
Эта ветка соответствует «origin/main».
~/repos/Coffeemachine main git merge branch2
Обновление a1622a4..312e939
Fast-forward
 file1.txt | 1 +
1 file changed, 1 insertion(+)
create mode 100644 file1.txt
~/repos/Coffeemachine main git push origin main
Username for 'https://github.com': Mercurlc
Password for 'https://Mercurlc@github.com':
Всего 0 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To https://github.com/Mercurlc/Coffeemachine.git
a1622a4..312e939 main -> main
```

Рисунок 10 — Персональные задания

Часть 3. Работа с ветвлением и оформление кода

1. Сделайте форк репозитория в соответствии с вашим вариантом
2. Склонируйте его на локальную машину
3. Создайте две ветки branch1 и branch2 от последнего коммита в master'e
4. Проведите по 3 коммита в каждую из веток, которые меняют один и тот

же кусочек файла

5. Выполните слияние ветки branch1 в ветку branch2, разрешив конфликты при этом

6. Выгрузите все изменения во всех ветках в удаленный репозиторий

7. Проведите еще 3 коммита в ветку branch1

8. Склонируйте репозиторий еще раз в другую директорию

9. В новом клоне репозитории сделайте 3 коммита в ветку branch1

10. Выгрузите все изменения из нового репозитория в удаленный репозиторий

11. Вернитесь в старый клон с репозиторием, выгрузите изменения с опцией —force

12. Получите все изменения в новом репозитории

Таблица 2 — Индивидуальный вариант

№ Варианта	Репозиторий
1	https://github.com/liamg/traitor

```
~/repos git clone git@github.com:Mercurlc/traitor.git
Клонирование в «traitor»...
Enter passphrase for key '/home/merc/.ssh/id_rsa':
remote: Enumerating objects: 1354, done.
remote: Counting objects: 100% (185/185), done.
remote: Compressing objects: 100% (61/61), done.
remote: Total 1354 (delta 133), reused 124 (delta 124), pack-reused 1169
Получение объектов: 100% (1354/1354), 2.94 МиБ | 2.05 МиБ/с, готово.
Определение изменений: 100% (704/704), готово.

~/repos cd traitor
~/repos/traitor main git checkout master
error: pathspec 'master' did not match any file(s) known to git
~/repos/traitor main git checkout main
Уже на «main»
Эта ветка соответствует «origin/main».
~/repos/traitor main git checkout -b branch1
Переключились на новую ветку «branch1»
~/repos/traitor branch1 git checkout main
Переключились на ветку «main»
Эта ветка соответствует «origin/main».
~/repos/traitor main git checkout -b branch2
Переключились на новую ветку «branch2»
~/repos/traitor branch2 echo 'text' > 'vendor/modules.txt'
~/repos/traitor branch2 !1 git add .
~/repos/traitor branch2 +1 git commit -m "first"
[branch2 df79847] first
1 file changed, 1 insertion(+), 33 deletions(-)
~/repos/traitor branch2 echo 'text2' >> 'vendor/modules.txt'
~/repos/traitor branch2 !1 git add .
~/repos/traitor branch2 +1 git commit -m "second"
[branch2 2e806bc] second
1 file changed, 1 insertion(+)
~/repos/traitor branch2 echo 'text3' >> 'vendor/modules.txt'
~/repos/traitor branch2 !1 git add .
~/repos/traitor branch2 +1 git commit -m "three"
[branch2 5e6de91] three
1 file changed, 1 insertion(+)
~/repos/traitor branch2 git checkout branch1
Переключились на ветку «branch1»
~/repos/traitor branch1 echo 'text1_1' > 'vendor/modules.txt'
~/repos/traitor branch1 !1 git add .
~/repos/traitor branch1 +1 git commit -m "first"
[branch1 0a72bae] first
1 file changed, 1 insertion(+), 33 deletions(-)
~/repos/traitor branch1 и так далее
```

Рисунок 11 — Задачи 1-4

```
❏ ~/repos/traitor P branch1 git checkout branch2 ✓ < ⚙
Переключились на ветку «branch2»
❏ ~/repos/traitor P branch2 git merge branch1 ✓ < ⚙
Автослияние vendor/modules.txt
КОНФЛИКТ (содержимое): Конфликт слияния в vendor/modules.txt
Сбой автоматического слияния; исправьте конфликты, затем зафиксируйте результат.
❏ ~/r/traitor P branch2 merge ~1 git checkout --theirs vendor/modules.txt 1 ✖ ⚙
Updated 1 path from the index
❏ ~/r/traitor P branch2 merge ~1 git add . ✓ < ⚙
❏ ~/r/traitor P branch2 merge +1 git commit -m "fix" ✓ < ⚙
[branch2 f5a7f64] fix
❏ ~/repos/traitor P branch2 git merge branch1 ✓ < ⚙
уже актуально.
❏ ~/repos/traitor P branch2 git push origin branch1 ✓ < ⚙
Enter passphrase for key '/home/merc/.ssh/id_rsa':
Перечисление объектов: 15, готово.
Подсчет объектов: 100% (15/15), готово.
При сжатии изменений используется до 12 потоков
Сжатие объектов: 100% (9/9), готово.
Запись объектов: 100% (12/12), 889 байтов | 889.00 КиБ/с, готово.
Всего 12 (изменений 6), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (6/6), completed with 2 local objects.
remote:
remote: Create a pull request for 'branch1' on GitHub by visiting:
remote:   https://github.com/Mercurlc/traitor/pull/new/branch1
remote:
To github.com:Mercurlc/traitor.git
* [new branch]      branch1 -> branch1
❏ ~/r/traitor P branch2 git push origin branch2 5s ⏸ ⚙
Enter passphrase for key '/home/merc/.ssh/id_rsa':
Перечисление объектов: 19, готово.
Подсчет объектов: 100% (19/19), готово.
При сжатии изменений используется до 12 потоков
Сжатие объектов: 100% (10/10), готово.
Запись объектов: 100% (13/13), 1.05 КиБ | 1.05 МиБ/с, готово.
Всего 13 (изменений 6), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (6/6), completed with 2 local objects.
remote:
remote: Create a pull request for 'branch2' on GitHub by visiting:
remote:   https://github.com/Mercurlc/traitor/pull/new/branch2
remote:
To github.com:Mercurlc/traitor.git
* [new branch]      branch2 -> branch2
❏ ~/r/traitor P branch2 4s ⏸ ⚙
```

Рисунок 12 — Задача 5-6

```
~/repos/traitor git P branch2 git checkout branch1
Переключились на ветку «branch1»
~/repos/traitor git P branch1 echo 'text1_3' > 'new_file1.txt'
~/repos/traitor git P branch1 ?1 git add .
~/repos/traitor git P branch1 +1 git commit -m "new1"
[branch1 3f1cb7e] new1
1 file changed, 1 insertion(+)
create mode 100644 new_file1.txt
~/repos/traitor git P branch1 echo 'text1_3' > 'new_file2.txt'
~/repos/traitor git P branch1 ?1 git add .
~/repos/traitor git P branch1 +1 git commit -m "new2"
[branch1 6647132] new2
1 file changed, 1 insertion(+)
create mode 100644 new_file2.txt
~/repos/traitor git P branch1 echo 'text1_3' >> 'new_file2.txt'
~/repos/traitor git P branch1 !1 git add .
~/repos/traitor git P branch1 +1 git commit -m "new3"
[branch1 d066b1f] new3
1 file changed, 1 insertion(+)
~/repos/traitor git P branch1 git status
Текущая ветка: branch1
нечего коммитить, нет изменений в рабочем каталоге
~/repos/traitor git P branch1 cd ../
~/repos mkdir new_repos
~/repos cd new_repos
cd: Нет такого файла или каталога: new_repos
~/repos cd new_repos
~/repos/new_repos git clone git@github.com:Mercurlc/traitor.git
Клонирование в «traitor»...
remote: Enumerating objects: 1379, done.
remote: Counting objects: 100% (210/210), done.
remote: Compressing objects: 100% (72/72), done.
remote: Total 1379 (delta 146), reused 144 (delta 132), pack-reused 1169
Получение объектов: 100% (1379/1379), 2.94 МиБ | 3.24 МиБ/с, готово.
Определение изменений: 100% (717/717), готово.
~/repos/new_repos cd traitor
~/repos/new_repos/traitor git P main
```

Рисунок 13 — Задачи 7-8

```
~/repos/new_repos/traitor P main git checkout branch1
branch 'branch1' set up to track 'origin/branch1'.
Переключились на новую ветку «branch1»
~/r/new_repo/traitor P branch1 echo 'text1_3' > 'another_commits.txt'
~/r/new_repo/traitor P branch1 ?1 git add .
~/r/new_repo/traitor P branch1 +1 git commit -m "a_1"
[branch1 5421c4c] a_1
1 file changed, 1 insertion(+)
create mode 100644 another_commits.txt
~/r/new_repo/traitor P branch1 +1 echo 'text1_3' > 'another_commits2.txt'
~/r/new_repo/traitor P branch1 +1 ?1 git add .
~/r/new_repo/traitor P branch1 +1 +1 git commit -m "a_2"
[branch1 cef0bd4] a_2
1 file changed, 1 insertion(+)
create mode 100644 another_commits2.txt
~/r/new_repo/traitor P branch1 +2 echo 'new_txt' >> 'another_commits2.txt'
~/r/new_repo/traitor P branch1 +2 !1 git commit -m "a_2"
Текущая ветка: branch1
Ваша ветка опережает «origin/branch1» на 2 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)

Изменения, которые не в индексе для коммита:
(используйте «git add <файл>...», чтобы добавить файл в индекс)
(используйте «git restore <файл>...», чтобы отменить изменения в рабочем каталоге)
изменено: another_commits2.txt

индекс пуст (используйте «git add» и/или «git commit -a»)
~/r/new_repo/traitor P branch1 +2 !1 echo 'new_txt' >> 'another_commits2.txt' 1 x
~/r/new_repo/traitor P branch1 +2 !1
~/r/new_repo/traitor P branch1 +2 !1 git add .
~/r/new_repo/traitor P branch1 +2 +1 git commit -m "a_3"
[branch1 13f55ac] a_3
1 file changed, 2 insertions(+)
~/r/new_repo/traitor P branch1 +3 git status
Текущая ветка: branch1
Ваша ветка опережает «origin/branch1» на 3 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)

ничего коммитить, нет изменений в рабочем каталоге
~/r/new_repo/traitor P branch1 +3 git push origin branch1 --force
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 12 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (8/8), 722 байта | 722.00 КиБ/с, готово.
Всего 8 (изменений 3), повторно использовано 1 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To github.com:Mercurlc/traitor.git
77b8a6a..13f55ac branch1 -> branch1
~/r/new_repos/traitor P branch1
```

Рисунок 14 — Задачи 9-10


```
~/r/new_repos/traitor P branch1 cd ../
~/repos/new_repos cd ../traitor
~/repos/traitor P branch1 git push origin branch1 --force
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 12 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (8/8), 691 байт | 691.00 КиБ/с, готово.
Всего 8 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To github.com:Mercurlc/traitor.git
+ 13f55ac...d066b1f branch1 -> branch1 (forced update)
~/repos/traitor P branch1 cd ../new_repos/traitor
~/r/new_repos/traitor P branch1 git pull origin branch1
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 8 (delta 3), reused 8 (delta 3), pack-reused 0
Распаковка объектов: 100% (8/8), 671 байт | 671.00 КиБ/с, готово.
Из github.com:Mercurlc/traitor
* branch          branch1      -> FETCH_HEAD
+ 13f55ac...d066b1f branch1      -> origin/branch1 (принудительное обновление)
подсказка: You have divergent branches and need to specify how to reconcile them.
подсказка: You can do so by running one of the following commands sometime before
подсказка: your next pull:
подсказка:
подсказка:   git config pull.rebase false  # merge
подсказка:   git config pull.rebase true    # rebase
подсказка:   git config pull.ff only         # fast-forward only
подсказка:
подсказка: You can replace "git config" with "git config --global" to set a default
подсказка: preference for all repositories. You can also pass --rebase, --no-rebase,
подсказка: or --ff-only on the command line to override the configured default per
подсказка: invocation.
fatal: Need to specify how to reconcile divergent branches.
~/r/new_repo/traitor P branch1 13f3 git merge origin/branch1
Merge made by the 'ort' strategy.
 new_file1.txt | 1 +
 new_file2.txt | 2 ++
2 files changed, 3 insertions(+)
create mode 100644 new_file1.txt
create mode 100644 new_file2.txt
~/r/new_repo/traitor P branch1 14
```

Рисунок 15 — Задачи 11-12

ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. **К какому типу систем контроля версий относится Git?** Git относится к типу распределённых систем контроля версий, где каждый разработчик имеет локальную копию всей истории репозитория, что позволяет работать с ним даже без доступа к сети.
2. **Что такое репозиторий Git?** Репозиторий Git — это каталог файлов вашего проекта. Он содержит все файлы, историю их изменений, а также управляющую информацию, которая нужна Git для работы.
3. **Что такое коммит?** Коммит в Git — это снимок всех файлов в вашем репозитории на момент времени. Каждый коммит сохраняется в истории изменений и имеет уникальный идентификатор (хеш).
4. **Что такое ветка в репозитории Git?** Ветка в Git — это легковесный "указатель" на один из коммитов. Новые ветки позволяют разработчикам работать над различными функциями или исправлениями независимо друг от друга.
5. **Что такое тег в репозитории Git?** Тег в Git — это ссылка на определённый момент в истории репозитория. Обычно теги используются для отметки релизов или важных точек в истории проекта.
6. **Что такое слияние двух веток?** Слияние двух веток в Git — это процесс объединения изменений из одной ветки в другую. Это часто делается для включения новой функциональности или исправлений в основную ветку проекта.
7. **Что такое конфликт в Git? Как его решить и почему они бывают?** Конфликт в Git возникает, когда один и тот же участок кода был изменён в двух разных коммитах, которые Git пытается объединить. Разрешить конфликт можно путём ручного редактирования файлов, чтобы выбрать, какие изменения следует сохранить, и затем сделать коммит с этими

изменениями.

8. **Для чего нужен .gitignore?** Файл `.gitignore` используется для указания файлов или паттернов файлов, которые Git должен игнорировать (не отслеживать и не включать в коммиты). Это могут быть временные файлы, логи, файлы среды разработки и прочие файлы, которые не должны попадать в репозиторий.

ВЫВОД

Получил навыки по работе с командной строкой и git'ом. Ответил на контрольные вопросы.