

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования «СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Институт цифрового развития

ОТЧЁТ

по лабораторной работе №1.2

Дисциплина: «Основы кроссплатформенного программирования»

Тема: «Исследование возможностей Git для работы с локальными
репозиториями»

Выполнила: студентка 1 курса,
группы ИВТ-б-о-21-1
Диченко Дина Алексеевна

Ставрополь 2022

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

Выполнение работы:

1. Изучила теоретический материал работы.
2. Создала общедоступный репозиторий GitHub.

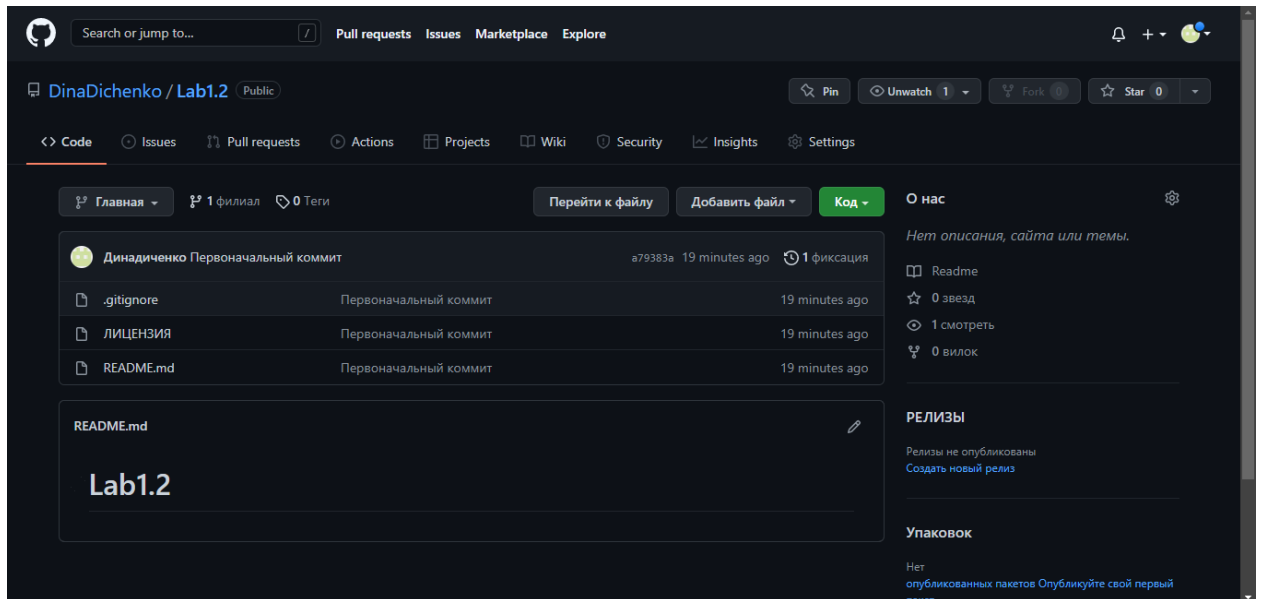


Рисунок 1. Создание репозитория на GitHub

3. Проработала примеры лабораторной работы.

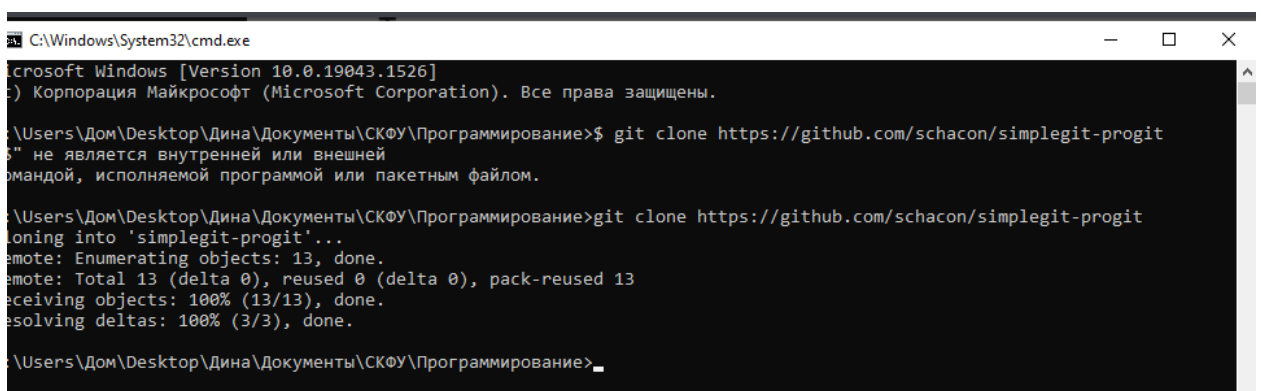


Рисунок 2. Клонирование репозитория

```

git-progit (master)
git log
commit ca82a6dff817ec66f44342007202690a93763949 (HEAD -> master, origin/master,
origin/HEAD)
author: Scott Chacon <schacon@gmail.com>
date: Mon Mar 17 21:52:11 2008 -0700

    changed the verison number

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
author: Scott Chacon <schacon@gmail.com>
date: Sat Mar 15 16:40:33 2008 -0700

    removed unnecessary test code

commit a11bef06a3f659402fe7563abf99ad00de2209e6
author: Scott Chacon <schacon@gmail.com>
date: Sat Mar 15 10:31:28 2008 -0700

    first commit

om@DESKTOP-EH2DCJR MINGW64 ~/Desktop/Дина/Документы/СКФУ/Программирование/simp1
git-progit (master)

```

Рисунок 3. Результат работы команды git log

```

commit ca82a6dff817ec66f44342007202690a93763949 (HEAD -> master, origin/mas
origin/HEAD)
Author: Scott Chacon <schacon@gmail.com>
Date: Mon Mar 17 21:52:11 2008 -0700

    changed the verison number

diff --git a/Rakefile b/Rakefile
index a874b73..8f94139 100644
--- a/Rakefile
+++ b/Rakefile
@@ -5,7 +5,7 @@ require 'rake/gempackagetask'
spec = Gem::Specification.new do |s|
  s.platform = Gem::Platform::RUBY
  s.name = "simplegit"
- s.version = "0.1.0"
+ s.version = "0.1.1"
  s.author = "Scott Chacon"
  s.email = "schacon@gmail.com"
  s.summary = "A simple gem for using Git in Ruby code."

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gmail.com>
:|

```

Рисунок 4. Результат работы опции -p -2

```

origin/HEAD)
author: Scott Chacon <schacon@gmail.com>
date: Mon Mar 17 21:52:11 2008 -0700

    changed the verison number

Rakefile | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
author: Scott Chacon <schacon@gmail.com>
date: Sat Mar 15 16:40:33 2008 -0700

    removed unnecessary test code

lib/simplegit.rb | 5 -----
1 file changed, 5 deletions(-)

commit a11bef06a3f659402fe7563abf99ad00de2209e6
author: Scott Chacon <schacon@gmail.com>
date: Sat Mar 15 10:31:28 2008 -0700

    first commit

README | 6 ++++++
Rakefile | 23 ++++++
lib/simplegit.rb | 25 ++++++
3 files changed, 54 insertions(+)

```

Рисунок 5. Результат работы опции --stat

```

egit-progit (master)
$ git log --pretty=oneline
ca82a6dff817ec66f44342007202690a93763949 (HEAD -> master, origin/master, origin/HEAD) changed the verison number
085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7 removed unnecessary test code
a11bef06a3f659402fe7563abf99ad00de2209e6 first commit

Дом@DESKTOP-EH2DCJR MINGW64 ~/Desktop/Дина/Документы/СКФУ/Программирование/simplegit
egit-progit (master)
$ git log --pretty=format:"%h - %an, %ar : %s"
ca82a6d - Scott Chacon, 14 years ago : changed the verison number
085bb3b - Scott Chacon, 14 years ago : removed unnecessary test code
a11bef0 - Scott Chacon, 14 years ago : first commit

Дом@DESKTOP-EH2DCJR MINGW64 ~/Desktop/Дина/Документы/СКФУ/Программирование/simplegit
egit-progit (master)
$ git log --pretty=format:"%h %s" --graph
* ca82a6d changed the verison number
* 085bb3b removed unnecessary test code
* a11bef0 first commit

Дом@DESKTOP-EH2DCJR MINGW64 ~/Desktop/Дина/Документы/СКФУ/Программирование/simplegit
egit-progit (master)

```

Рисунок 6. Результаты работы опции --pretty

```

Дом@DESKTOP-EH2DCJR MINGW64 ~/Desktop/Дина/Документы/СКФУ/Программирование/simplegit
$ git remote -v
origin https://github.com/schacon/ticgit (fetch)
origin https://github.com/schacon/ticgit (push)

Дом@DESKTOP-EH2DCJR MINGW64 ~/Desktop/Дина/Документы/СКФУ/Программирование/simplegit

```

Рисунок 7. Просмотр удаленных репозитория

```
$ git fetch pb
remote: Enumerating objects: 43, done.
remote: Counting objects: 100% (22/22), done.
remote: Total 43 (delta 22), reused 22 (delta 22), pack-reused 21
Unpacking objects: 100% (43/43), 5.99 KiB | 2.00 KiB/s, done.
From https://github.com/paulboone/ticgit
* [new branch]      master    -> pb/master
* [new branch]      ticgit    -> pb/ticgit
```

Рисунок 8. Добавление удаленных репозитория

```
$ git push origin master
remote: Permission to schacon/ticgit.git denied to DinaDichenko.
fatal: unable to access 'https://github.com/schacon/ticgit/': The requested URL
returned error: 403

Дом@DESKTOP-EH2DCJR MINGW64 ~/Desktop/Дина/Документы/СКФУ/Программирование/ticgit
$ git remote show origin
remote origin
Fetch URL: https://github.com/schacon/ticgit
Push URL: https://github.com/schacon/ticgit
HEAD branch: master
Remote branches:
  master tracked
  ticgit tracked
Local branch configured for 'git pull':
  master merges with remote master
Local ref configured for 'git push':
  master pushes to master (up to date)
```

Рисунок 9. Отправка изменений в удаленный репозиторий и просмотр удаленного репозитория

```
Дом@DESKTOP-EH2DCJR MINGW64 ~/Desktop/Дина/Документы/СКФУ/Программирование/ticgit
$ git remote rename pb paul

Дом@DESKTOP-EH2DCJR MINGW64 ~/Desktop/Дина/Документы/СКФУ/Программирование/ticgit
$ remote
bash: remote: command not found

Дом@DESKTOP-EH2DCJR MINGW64 ~/Desktop/Дина/Документы/СКФУ/Программирование/ticgit
$ git remote
origin
paul
```

Рисунок 10. Переименование удаленного репозитория

```

tag v1.4
Tagger: DinaDichenko <dinadichenko@mail.ru>
Date: Thu Mar 24 23:38:08 2022 +0300

my version 1.4

commit 847256809a3d518cd36b8f81859401416fe8d945 (HEAD -> master, tag: v1.4, origin/master, origin/HEAD)
Author: Jeff Welling <Jeff.Welling@Gmail.com>
Date: Tue Apr 26 17:29:17 2011 -0700

    Added note to clarify which is the canonical TicGit-ng repo

diff --git a/README.mkd b/README.mkd
index ab92035..9ea9ff9 100644
--- a/README.mkd
+++ b/README.mkd
@@ -1,3 +1,6 @@
+Note: the original TicGit author has pulled all the TicGit-ng changes into his
+repository, creating a potentially confusing situation. The schacon TicGit repo,
+this one, is not consistently maintained. For up to date TicGit-ng info and code,
+check the canonical TicGit-ng repository at https://github.com/jeffwelling/ticgit

```

Рисунок 11. Создание аннотированного тега

```

$ git tag -d v1.4
Deleted tag 'v1.4' (was b1fbdaf)

```

Рисунок 12. Удаление тега

4. Выполнила клонирование репозитория на рабочий компьютер.

```

C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование>git clone https://github.com/DinaDichenko/Lab1.2.git
Cloning into 'Lab1.2'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.

```

Рисунок 13. Клонирование репозитория

5. Дополнила файл .gitignore необходимыми правилами для выбранного языка программирования и интегрированной среды разработки.

```
# Prerequisites
*.d

# Compiled Object files
*.slo
*.lo
*.o
*.obj

# Precompiled Headers
*.gch
*.pch

# Compiled Dynamic libraries
*.so
*.dylib
*.dll

# Fortran module files
*.mod
*.smod

# Compiled Static libraries
```

Рисунок 14. Файл .gitignore

6. Дополнила файл README.md.

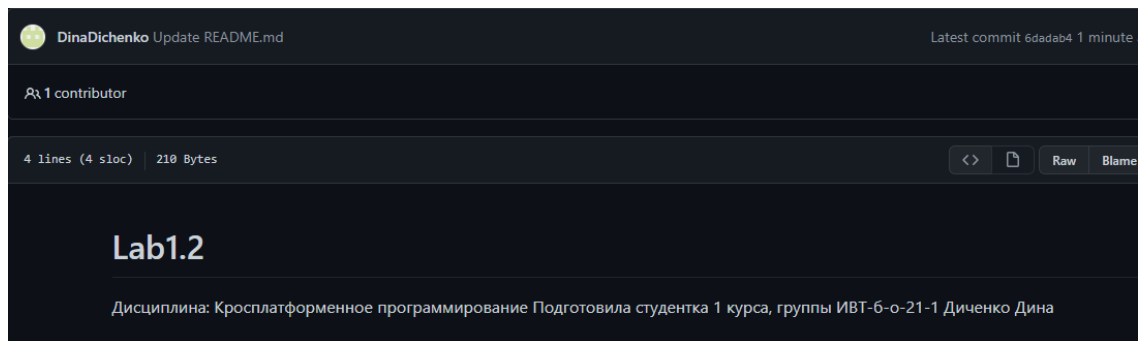


Рисунок 15. Дополнение файла README.md

7. Написала небольшую программу на выбранном языке.

```
C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab1.2>git commit -m "added Lab1.2"
[main 1174557] added Lab1.2
9 files changed, 237 insertions(+)
create mode 100644 Lab1.2/.vs/Lab1.2/project-colors.json
create mode 100644 Lab1.2/.vs/Lab1.2/v17/.suo
create mode 100644 Lab1.2/.vs/Lab1.2/v17/Browse.VC.db
create mode 100644 Lab1.2/.vs/Lab1.2/v17/ipch/AutoPCH/e8fb4e9154777b1f/LAB1.2.ipch
create mode 100644 Lab1.2/Lab1.2.sln
create mode 100644 Lab1.2/Lab1.2/Lab1.2.cpp
create mode 100644 Lab1.2/Lab1.2/Lab1.2.vcxproj
create mode 100644 Lab1.2/Lab1.2/Lab1.2.vcxproj.filters
create mode 100644 Lab1.2/Lab1.2/Lab1.2.vcxproj.user

C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab1.2>git push
Enumerating objects: 20, done.
Counting objects: 100% (20/20), done.
Delta compression using up to 2 threads
Compressing objects: 100% (15/15), done.
Writing objects: 100% (19/19), 11.53 MiB | 1.73 MiB/s, done.
Total 19 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/DinaDichenko/Lab1.2.git
6dadab4..1174557 main -> main
```

Рисунок 16. Пуш программы на удаленный сервер







 DinaDichenko added Lab1.2	1174557 3 minutes ago	 3 commits
 Lab1.2	added Lab1.2	3 minutes ago
 .gitignore	Initial commit	3 hours ago
 LICENSE	Initial commit	3 hours ago
 README.md	Update README.md	1 hour ago

Рисунок 17. Изменение на удаленном сервере

```

4   #include <iostream>
5   using namespace std;
6   int main()
7   {
8       cout << "Hello World!\n";
9       cout << "Peace";
10      int a, b, c;
11      cin >> a >> b;

```

```

C:\Windows\System32\cmd.exe

C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab1.2>git commit -am "first comm"
[main f5ad38a] first comm
1 file changed, 0 insertions(+), 0 deletions(-)

C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab1.2>git commit -am "second comm"
[main 7b81cec] second comm
1 file changed, 0 insertions(+), 0 deletions(-)

10C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab1.2>

```

Рисунок 18. Первый и второй коммиты

```

4   #include <iostream>
5   using namespace std;
6   int main()
7   {
8       cout << "Hello World!\n";
9       cout << "Peace";
10      int a, b, c;
11      cin >> a >> b;
12      c = (a * b) / (a - b);

```

```

C:\Windows\System32\cmd.exe

:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab1.2>git commit -am "first comm"
main f5ad38a] first comm
1 file changed, 0 insertions(+), 0 deletions(-)

:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab1.2>git commit -am "second comm"
main 7b81cec] second comm
1 file changed, 0 insertions(+), 0 deletions(-)

:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab1.2>git commit -am "third comm"
main 2618602] third comm
1 file changed, 0 insertions(+), 0 deletions(-)

:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab1.2>

```

Рисунок 19. Третий коммит


```
5 using namespace std;
6 int main()
7 {
8     cout << "Hello World!\n";
9     cout << "Peace";
10    int a, b, c;
11    cin >> a >> b;
12    c = (a * b) / (a - b);
13    cout << "Результат: ";

C:\Windows\System32\cmd.exe

C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab1.2>git commit -am "first comm"
[main f5ad38a] first comm
1 file changed, 0 insertions(+), 0 deletions(-)

C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab1.2>git commit -am "second comm"
[main 7b81cec] second comm
1 file changed, 0 insertions(+), 0 deletions(-)

C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab1.2>git commit -am "third comm"
[main 2618602] third comm
1 file changed, 0 insertions(+), 0 deletions(-)

C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab1.2>git commit -am "Fourth comm"
[main 52adc52] Fourth comm
1 file changed, 0 insertions(+), 0 deletions(-)

C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab1.2>_
```

Рисунок 20. Четвертый коммит

```
C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab1.2>git commit -am "6 comm"
[main 4ed225a] 6 comm
1 file changed, 0 insertions(+), 0 deletions(-)

C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab1.2>git tag -a v1.2 -m "6 comm"

C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab1.2>git commit -am "7 comm"
[main 9f2fb02] 7 comm
1 file changed, 0 insertions(+), 0 deletions(-)

C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab1.2>git tag -a v1.3 -m "7 comm"
```

Рисунок 21. Пятый, шестой и седьмой коммиты, отмеченные тегами

```
C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab1.2>git tag
v1.1
v1.2
v1.3
```

Рисунок 22. Теги

8. Посмотрела историю хранилища.

```
C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab1.2>git log --graph --pretty=oneline --abbrev-commit
* 9f2fb02 (HEAD -> main, tag: v1.3) 7 comm
* 4ed225a (tag: v1.2) 6 comm
* ec583c5 (tag: v1.1) 5 comm
* 52adc52 Fourth comm
* 2618602 third comm
* 7b81cec second comm
* f5ad38a first comm
* 1174557 (origin/main, origin/HEAD) added Lab1.2
* 6dadab4 Update README.md
* a79383a Initial commit
```

Рисунок 23. История хранилища

9. Посмотрела содержимое коммитов.

```
C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab1.2>git show HEAD
commit 9f2fb02c0258f78508c144771f527da1afbce16a (HEAD -> main, tag: v1.3)
Author: DinaDichenko <dinadichenko@mail.ru>
Date:   Fri Mar 25 01:51:20 2022 +0300

    7 comm

diff --git a/Lab1.2/.vs/Lab1.2/v17/Browse.VC.db b/Lab1.2/.vs/Lab1.2/v17/Browse.VC.db
index 7b21625..b7b13ed 100644
Binary files a/Lab1.2/.vs/Lab1.2/v17/Browse.VC.db and b/Lab1.2/.vs/Lab1.2/v17/Browse.VC.db differ
```

Рисунок 24. Содержимое седьмого коммита

10. Освоила возможности отката к заданной версии.

```
C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab1.2>git reset --hard
HEAD is now at 9f2fb02 7 comm

C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab1.2> git tag
v1.1
v1.2
v1.3

C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab1.2>git show
commit 9f2fb02c0258f78508c144771f527da1afbce16a (HEAD -> main, tag: v1.3)
```

Рисунок 25. Откат состояния хранилища к предыдущей версии

11. Зафиксировала сделанные изменения.

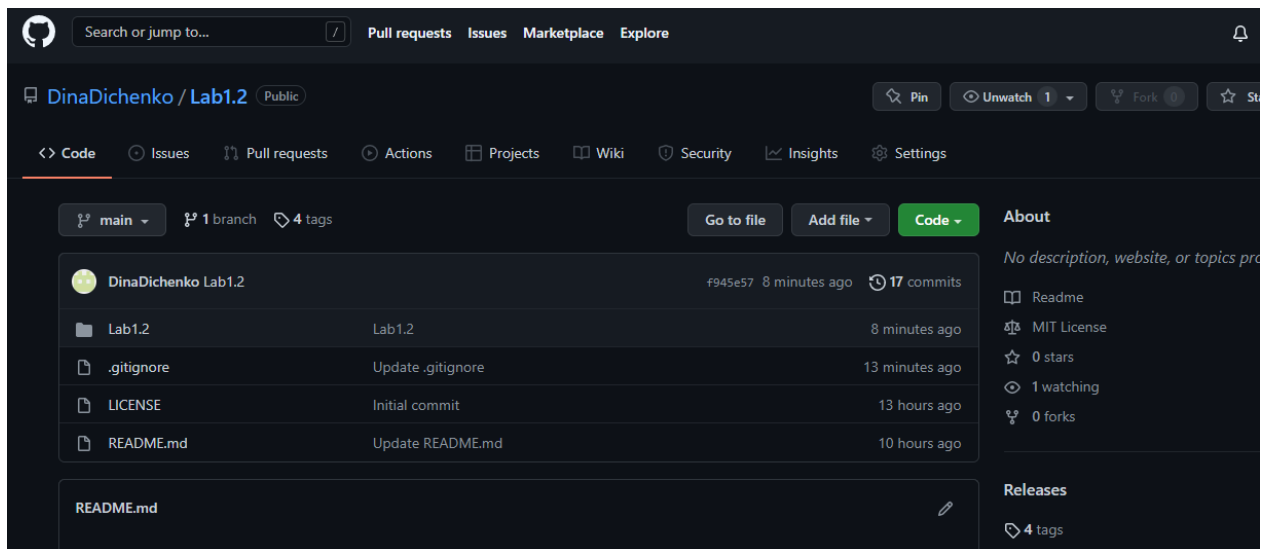


Рисунок 26. Изменение на удаленном репозитории

12. Добавила отчет по лабораторной работе в формате PDF в папку doc репозитория.

13. Зафиксировала изменения.

14. 13. Отправила изменения из локального репозитория в удаленный репозиторий GitHub.

15.

16. 14. Проконтровала изменения, произошедшие в репозитории GitHub.

Ответы на вопросы:

1. Как выполнить историю коммитов в Git? Какие существуют дополнительные опции для просмотра истории коммитов?

После того, как вы создали несколько коммитов или же клонировали репозиторий с уже существующей историей коммитов, вероятно Вам понадобится возможность посмотреть что было сделано — историю коммитов. Одним из основных и наиболее мощных инструментов для этого является команда `git log`.

Команда `git log` имеет очень большое количество опций для поиска коммитов по разным критериям. Рассмотрим наиболее популярные из них.

Одним из самых полезных аргументов является `-p` или `--patch`, который показывает разницу (выводит патч), внесенную в каждый коммит. Так же вы можете ограничить количество записей в выводе команды.

Если вы хотите увидеть сокращенную статистику для каждого коммита, вы можете использовать опцию `--stat`.

Следующей действительно полезной опцией является `--pretty`. Эта опция меняет формат вывода. Существует несколько встроенных вариантов отображения. Опция `oneline` выводит каждый коммит в одну строку, что может быть очень удобным если вы просматриваете большое количество коммитов. К тому же, опции `short`, `full` и `fuller` делают вывод приблизительно в том же формате, но с меньшим или большим количеством информации соответственно.

Наиболее интересной опцией является `format`, которая позволяет указать формат для вывода информации.

2. Как ограничить вывод при просмотре истории коммитов?

В дополнение к опциям форматирования вывода, команда `git log` принимает несколько опций для ограничения вывода — опций, с помощью которых можно увидеть определенное подмножество коммитов. Одна из таких опций — это опция `-2`, которая показывает только последние два коммита. В действительности вы можете использовать `-<n>`, где `n` — это любое натуральное число и представляет собой `n` последних коммитов. На практике вы не будете часто использовать эту опцию, потому что Git по умолчанию использует постраничный вывод и вы будете видеть только одну страницу за раз.

Однако, опции для ограничения вывода по времени, такие как `--since` и `--until`, являются очень удобными.

Опция `--author` дает возможность фильтровать по автору коммита, а опция `--grep` искать по ключевым словам в сообщении коммита.

3. Как внести изменения в уже сделанный коммит?

Помните, все что попало в коммит почти всегда Git может восстановить. Можно восстановить даже коммиты из веток, которые были удалены, или коммиты, перезаписанные параметром `--amend`. Но всё, что не было включено в коммит и потеряно — скорее всего, потеряно навсегда.

4. Как отменить индексацию файла в Git?

Использовать `git reset HEAD <file>...` для исключения из индекса.

5. Как отменить изменения в файле?

Использовать `"git reset HEAD <file>..."` для возвращения к версии из последнего коммита.

6. Что такое удаленный репозиторий Git?

Удалённые репозитории представляют собой версии вашего проекта, сохранённые в интернете или ещё где-то в сети.

7. Как выполнить просмотр удаленных репозиториях данного локального репозитория?

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. Как добавить удаленный репозиторий для данного локального репозитория?

Для того, чтобы добавить удалённый репозиторий и присвоить ему имя (`shortname`), просто выполните команду `git remote add <shortname> <url>`.

9. Как выполнить отправку/получение изменений с удаленного репозитория?

Для получения данных из удалённых проектов, следует выполнить `$ git fetch [remote-name]`.

Когда вы хотите поделиться своими наработками, вам необходимо отправить их в удалённый репозиторий. Команда для этого действия простая: `git push <remote-name> <branch-name>`.

10. Как выполнить просмотр удаленного репозитория?

Если хотите получить побольше информации об одном из удалённых репозиториях, вы можете использовать команду `git remote show <remote>`. Она выдаёт URL удалённого репозитория, а также информацию об отслеживаемых ветках.

11. Каково назначение тэгов Git?

Как и большинство СКВ, Git имеет возможность пометить определённые моменты в истории как важные. Как правило, эта функциональность используется для отметки моментов выпуска версий (v1.0, и т. п.). Такие пометки в Git называются тегами.

12. Как осуществляется работа с тэгами Git?

Просмотреть список имеющихся тегов в Git можно очень просто. Достаточно набрать команду `git tag` (параметры `-l` и `--list` опциональны).

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

По умолчанию, команда `git push` не отправляет теги на удалённые сервера. После создания тега нужно отправлять явно на удалённый сервер. Процесс аналогичен отправке веток — достаточно выполнить команду `git push origin <tagname>`.

Для удаления тега в локальном репозитории достаточно выполнить команду `git tag -d <tagname>`.

Если вы хотите получить версии файлов, на которые указывает тег, то вы можете сделать `git checkout` для тега. Однако, это переведёт репозиторий в состояние «detached HEAD», которое имеет ряд неприятных побочных эффектов.

13. Самостоятельно изучите назначение флага `--prune` в командах `git fetch` и `git push`. Каково назначение этого флага?

Вывод: в результате выполнения работы были исследованы базовые возможности системы контроля версий Git для работы с локальными репозиториями.