

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 РАБОТЕ №1
дисциплины «Основы кроссплатформенного программирования»

Выполнил:
Кондратенко Даниил Витальевич
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., канд. тех. наук, доцент,
доцент кафедры инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: исследование возможностей Git для работы с локальными репозиториями.

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

Порядок выполнения работы:

Задание 1.

Создал новый репозиторий и клонировал его на свой компьютер.

```
C:\Users\HUAWEI>git clone https://github.com/DaniiGit23/LabRab2.git
Cloning into 'LabRab2'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
C:\Users\HUAWEI>
```

Рисунок 1. Новый репозиторий.

Задание 2.

Добавил некоторое правило в файл *gitignore*, чтобы Git игнорировал файлы в формате *.png*.

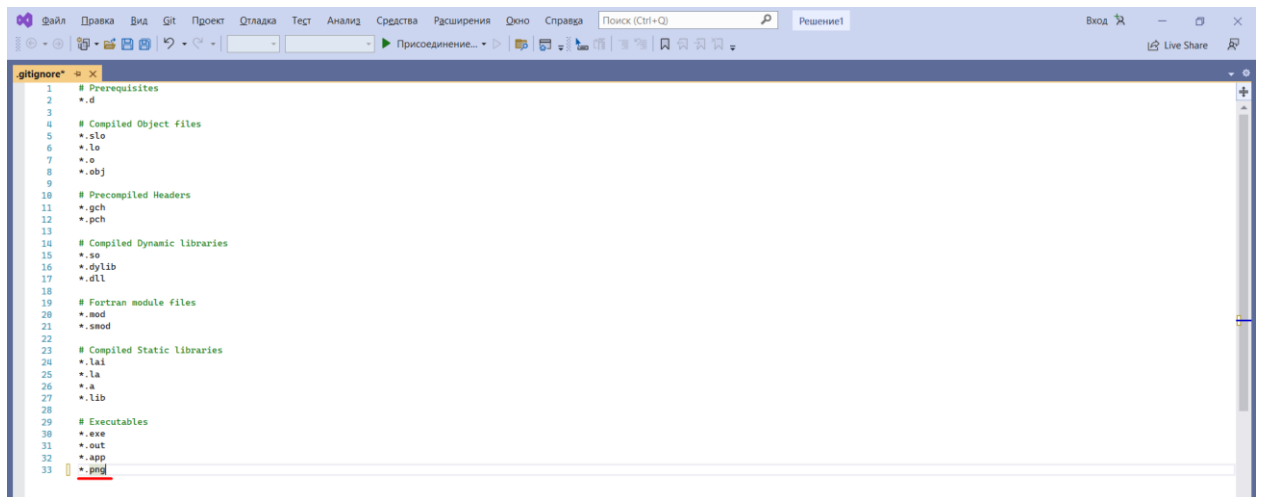


Рисунок 2. Работа с gitignore.

Задание 3.

Добавил информацию в файл README.md о дисциплине, группе и ФИО.

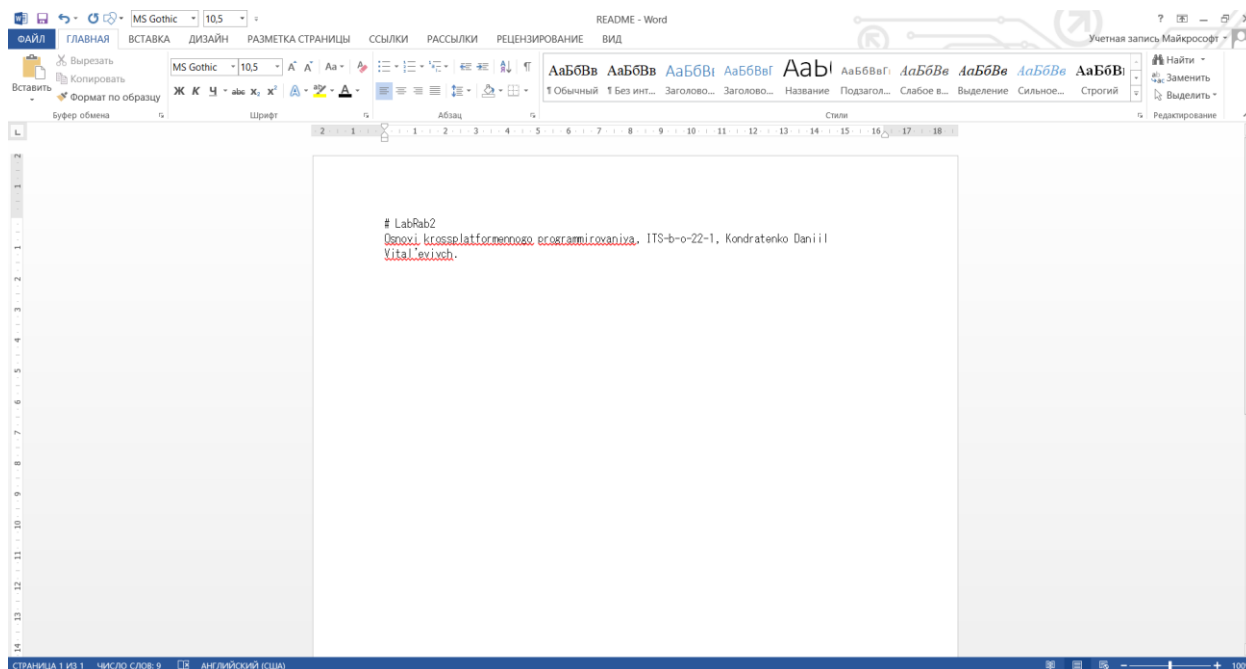
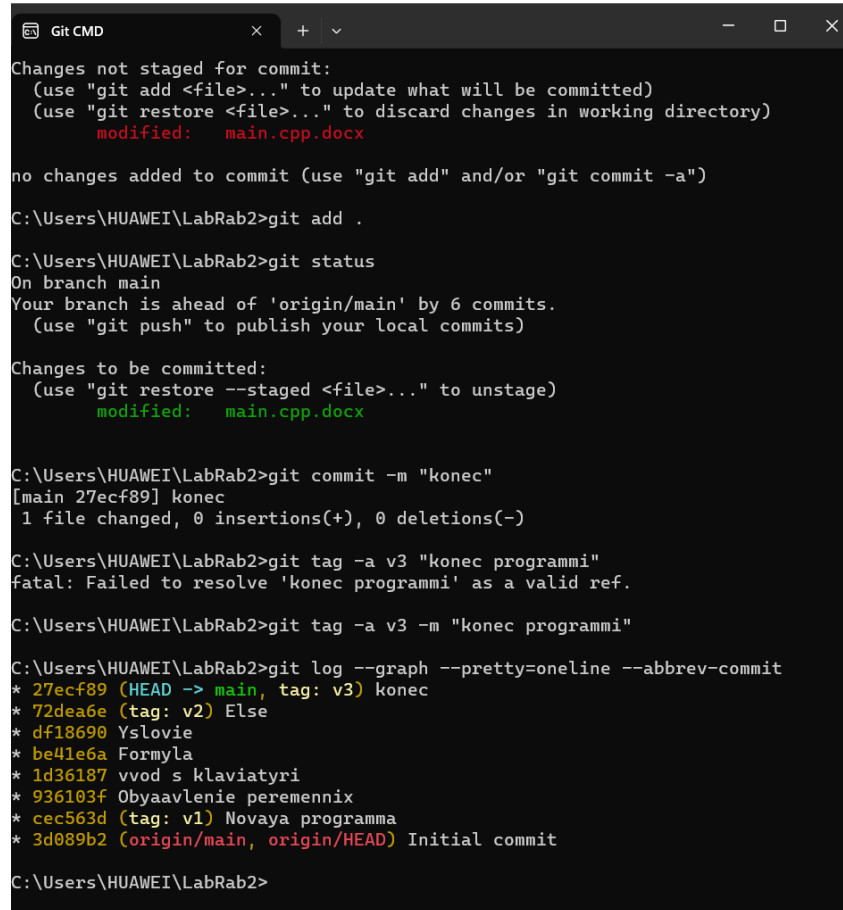


Рисунок 3. Работа с README.

Задание 4.

Написал небольшую программу в новом файле `main.cpp`, сделал не менее 7-ми коммитов с 3-мя тегами.



```
Git CMD
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:   main.cpp.docx

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\HUAWEI\LabRab2>git add .

C:\Users\HUAWEI\LabRab2>git status
On branch main
Your branch is ahead of 'origin/main' by 6 commits.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   main.cpp.docx

C:\Users\HUAWEI\LabRab2>git commit -m "konec"
[main 27ecf89] konec
 1 file changed, 0 insertions(+), 0 deletions(-)

C:\Users\HUAWEI\LabRab2>git tag -a v3 "konec programmi"
fatal: Failed to resolve 'konec programmi' as a valid ref.

C:\Users\HUAWEI\LabRab2>git tag -a v3 -m "konec programmi"

C:\Users\HUAWEI\LabRab2>git log --graph --pretty=oneline --abbrev-commit
* 27ecf89 (HEAD -> main, tag: v3) konec
* 72dea6e (tag: v2) Else
* df18690 Yslovie
* be41e6a Formyla
* 1d36187 vvod s klaviatyri
* 936103f Obyaavlenie peremennix
* cec563d (tag: v1) Novaya programma
* 3d089b2 (origin/main, origin/HEAD) Initial commit

C:\Users\HUAWEI\LabRab2>
```

Рисунок 4. История хранилища.

С помощью команды `git log --graph --pretty=oneline --abbrev-commit` можно проанализировать все свои коммиты и теги. `--graph` отображает ASCII граф с ветвлениями и историей слияний, `--pretty=oneline` показывает коммиты в альтернативном формате и `--abbrev-commit` показывает только несколько символов SHA-1 чек-суммы вместо всех 40.

Задание 5.

Посмотрел содержимое коммитов командой `git show <ref>`, где `<ref>`:

- 1) HEAD : последний коммит;

```
C:\Users\HUAWEI\LabRab2>git show HEAD
commit 27ecf89be1970b5d4b1ae667745fff882f21052 (HEAD -> main, tag: v3)
Author: Daniil <kondratenko_danil.23@mail.ru>
Date: Sun Feb 19 15:42:13 2023 +0300

    konec

diff --git a/main.cpp.docx b/main.cpp.docx
index e485147..af14a4c 100644
--- a/main.cpp.docx
+++ b/main.cpp.docx
@@ -26,4 +26,7 @@ My programm:
         x3 = (-b) / (2 * a);
         printf("%lf\n", x3);
     }
+    else printf("Net sasheniy");
+    }
+    }

C:\Users\HUAWEI\LabRab2>
```

Рисунок 5. Последний коммит.

- 2) HEAD~1 : предпоследний коммит (и т. д.);

```
C:\Users\HUAWEI\LabRab2>git show HEAD~1
commit 72dea6e248df8691aa4001ab8f7b75c6541c479b (tag: v2)
Author: Daniil <kondratenko_danil.23@mail.ru>
Date: Sun Feb 19 15:39:43 2023 +0300

    Else

diff --git a/main.cpp.docx b/main.cpp.docx
index 115d85a..e485147 100644
--- a/main.cpp.docx
+++ b/main.cpp.docx
@@ -19,4 +19,11 @@ My programm:
     printf("%lf\n", x1);
     printf("%lf\n", x2);
 }
+ else
+ {
+     if (D == 0)
+     {
+         x3 = (-b) / (2 * a);
+         printf("%lf\n", x3);
+     }
+ }

C:\Users\HUAWEI\LabRab2>
```

Рисунок 5. Предпоследний коммит.

3) b34a0e : коммит с указанным хэшем.

```
C:\Users\HUAWEI\LabRab2>git show 1d36187
commit 1d3618702ae9a16efca01cd2d74930334ce7c029
Author: Daniil <kondratenko_danil.23@mail.ru>
Date:   Sun Feb 19 15:37:32 2023 +0300

    vvod s klaviatyri

diff --git a/main.cpp.docx b/main.cpp.docx
index b70e8a7..311e207 100644
--- a/main.cpp.docx
+++ b/main.cpp.docx
@@ -5,4 +5,10 @@ My programm:
     int main()
     {
         double a, b, c, D, x1, x2, x3;
+    printf("Write a ");
+    scanf_s("%lf", &a);
+    printf("Write b ");
+    scanf_s("%lf", &b);
+    printf("Write c ");
+    scanf_s("%lf", &c);

C:\Users\HUAWEI\LabRab2>
```

Рисунок 6. Коммит с указанным хэшем.

Задание 6. Откат к заданной версии.

1.1. Удалил весь программный код с файла main.cpp и сохранил его.

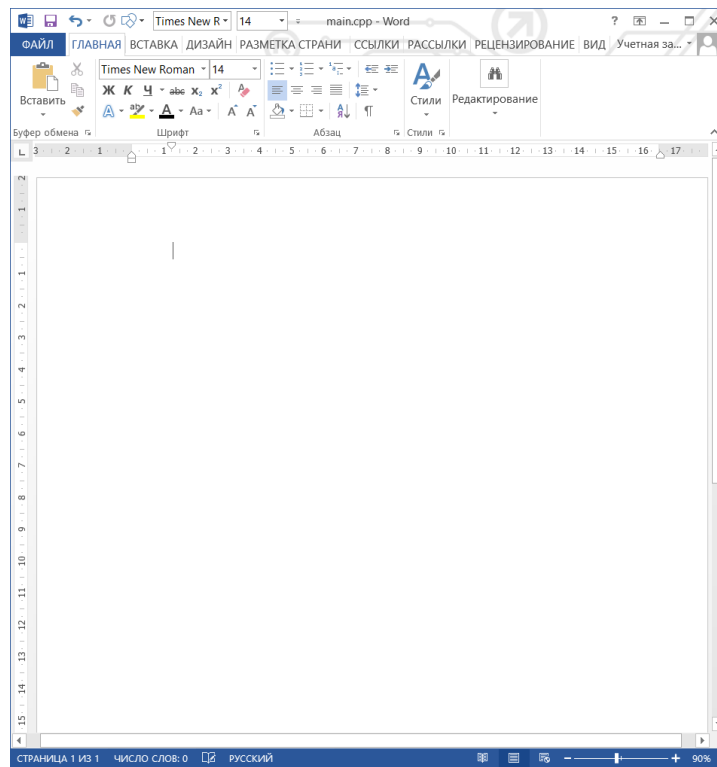


Рисунок 7. Удаление программ

1.2. Удалил это изменение с помощью команды `git checkout --main.cpp.docx`.

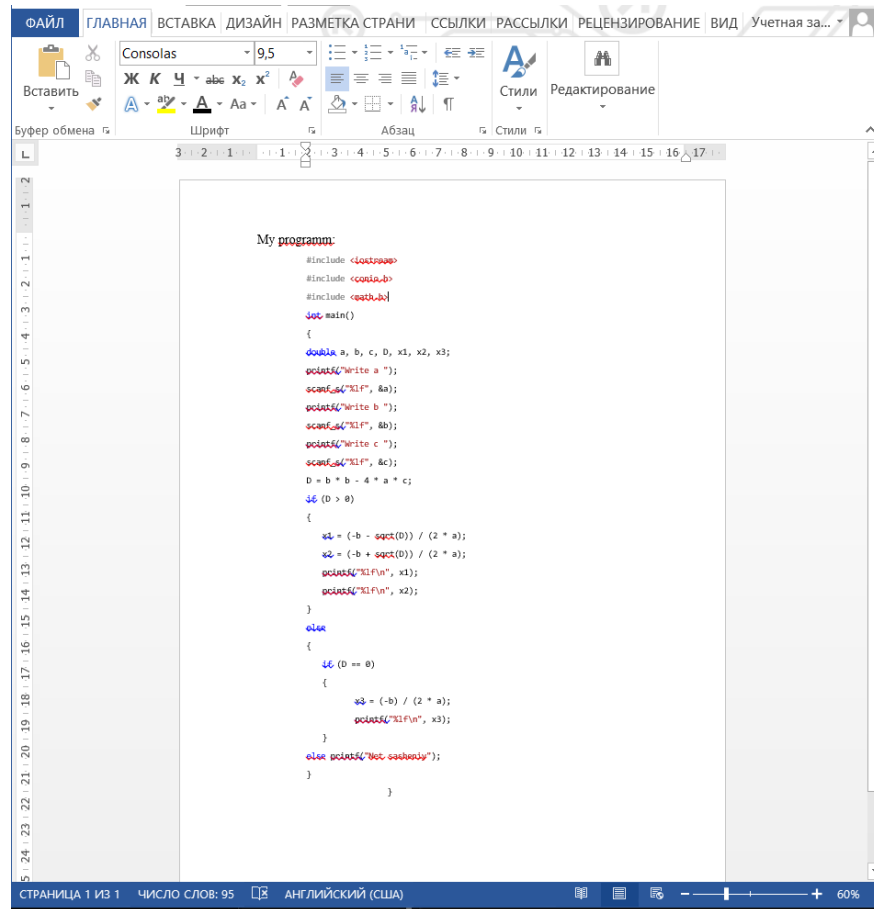


Рисунок 8. Восстановление программы.

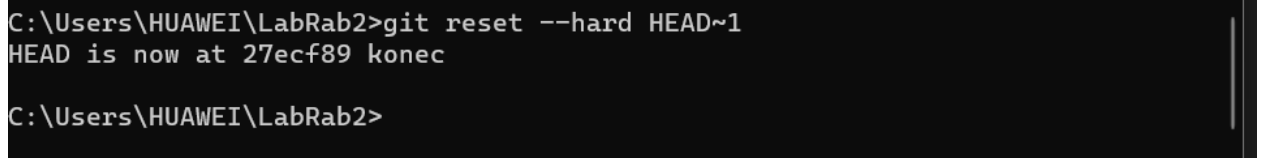
Код вновь вернулся.

1.3. Вновь повторил пункт 1.1. и сделал коммит.

```
C:\Users\HUAWEI\LabRab2>git add .
C:\Users\HUAWEI\LabRab2>git commit -m "Ydalil kod v main.cpp"
[main 1fbc5a3] Ydalil kod v main.cpp
2 files changed, 1 insertion(+), 1 deletion(-)
C:\Users\HUAWEI\LabRab2>
```

Рисунок 9. Коммит.

1.4. Откатить состояние хранилища к предыдущей версии командой:
git reset --hard HEAD~1 .



```
C:\Users\HUAWEI\LabRab2>git reset --hard HEAD~1
HEAD is now at 27ecf89 konec
C:\Users\HUAWEI\LabRab2>
```

Рисунок 10. Возвращение к предпоследней версии коммита.

Ссылка на репозиторий: <https://github.com/DaniiGit23/LabRab2.git>

После сделанных пунктов можно сделать вывод, что можно отменять ненужные изменения, в случае если коммит не произошел и изменения не нужны. Также, если коммит был уже сделан, то можно вернуться к предпоследней версии коммита (где было сохранение до сохранения с ненужными изменениями).

Ответы на контрольные вопросы:

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

Историю коммитов можно выполнить с помощью команды *git log*.

Дополнительные опции для просмотра истории:

%N, %h, %T, %t, %P, %p тд.

-p, --stat, --shortstat, --name-only, --name-status и тд.

2) Как ограничить вывод при просмотре истории коммитов?

Ограничить вывод при просмотре истории коммитов можно с помощью команды *git log -n*, где n – число последних коммитов.

3) Как внести изменения в уже сделанный коммит?

Если вы хотите переделать коммит — внесите необходимые изменения, добавьте их в индекс и сделайте коммит ещё раз, указав параметр *--amend* : *git commit --amend*.

4) Как отменить индексацию файла в Git?

Отменить индексацию файла можно с помощью команды: *git reset HEAD <file>*.

5) Как отменить изменения в файле?

Отменить изменения в файле можно с помощью команды: *git checkout -* *<file>*

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?

Выполнить просмотр удаленных репозиториях данного локального репозитория можно с помощью команды: *git remote*.

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]*.

Для отправки изменений в удаленный репозиторий используется команда: *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>*.

11) Каково назначение тэгов Git?

Git имеет возможность пометить определённые моменты в истории как важные. Для таких случаев были придуманы тэги.

12) Как осуществляется работа с тэгами Git?

Просмотреть список имеющихся тегов в Git можно очень просто. Достаточно набрать команду *git tag*.

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

С помощью команды *git show* вы можете посмотреть данные тега вместе с коммитом.

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

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

Если вы хотите получить версии файлов, на которые указывает тег, то вы можете сделать *git checkout <tagname>* для тега.

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

Git *prune* — это команда, которая удаляет все файлы, недоступные из текущей ветки. Команда *prune* полезна, когда в вашем рабочем каталоге много файлов, которые вы не хотите хранить.

git fetch --prune делает то же самое: удалит ссылки на ветки, которые не существуют на удалённом компьютере.

Опция *--prune* в команде *git push* удалит ветку из удалённого репозитория, если в локальной репозитории не существует ветки с таким именем.

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