

Федеральное государственное автономное образовательное учреждение
высшего образования «Северо-Кавказский федеральный университет»
Институт цифрового развития

ОТЧЕТ ПО ДИСЦИПЛИНЕ
ОСНОВЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ

Выполнила:

Мирзаева Камилла Мирзаевна

2 курс, группа ПИЖ-б-о-20-1

Принял:

Воронкин Роман Александрович

Ставрополь, 2021 г.

```
Git CMD

C:\Users\Камилла>cd moon
C:\Users\Камилла\moon>explorer .
C:\Users\Камилла\moon>git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

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:   README.md

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

C:\Users\Камилла\moon>git add .
C:\Users\Камилла\moon>git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md

C:\Users\Камилла\moon>git commit -m "изменения"
[main 91115ff] изменения
1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\Камилла\moon>git push
info: please complete authentication in your browser...
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 600 bytes | 600.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/mirzaevakamilla/moon.git
   6672718..91115ff  main -> main

C:\Users\Камилла\moon>_
```

Рисунок 1- скриншот терминала командами Git

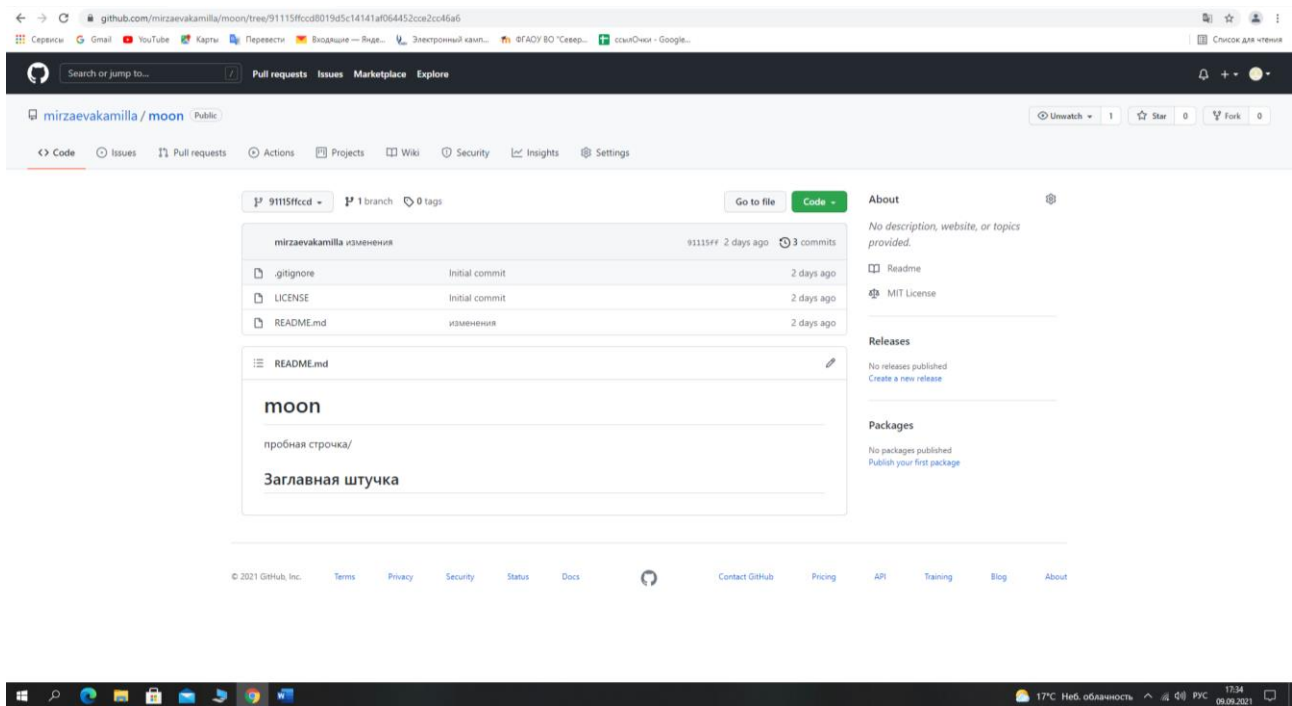


Рисунок 2 - скриншот репозитория после соответствующих изменений

```
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   README.md

C:\Users\Камилла\moon>git commit -m "изменения"
[main 91115ff] изменения
1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\Камилла\moon>git push
info: please complete authentication in your browser...
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 600 bytes | 600.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/mirzaevakamilla/moon.git
   6672718..91115ff  main -> main

C:\Users\Камилла\moon>git status
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:   README.md

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

C:\Users\Камилла\moon>git add .

C:\Users\Камилла\moon>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   README.md

C:\Users\Камилла\moon>git commit -m "change"
[main 20e4cdf] change
1 file changed, 2 insertions(+), 2 deletions(-)

C:\Users\Камилла\moon>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), 296 bytes | 296.00 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/mirzaevakamilla/moon.git
   91115ff..20e4cdf  main -> main

C:\Users\Камилла\moon>
```

Рисунок 3 - скриншот терминала командами Git

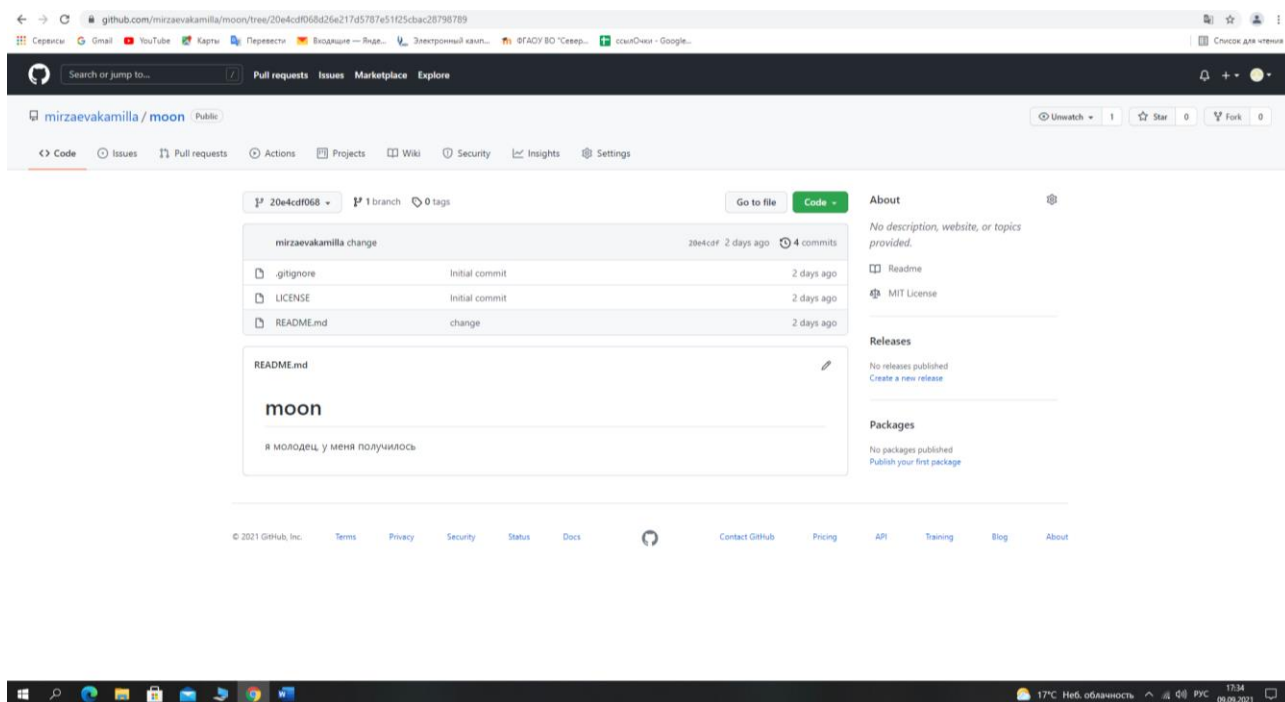


Рисунок 4 - скриншот репозитория после соответствующих изменений

Git CMD

```
(use "git restore --staged <file>..." to unstage)
modified: README.md

C:\Users\Камилла\moon>git commit -m "change"
[main 20e4cdf] change
1 file changed, 2 insertions(+), 2 deletions(-)

C:\Users\Камилла\moon>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), 296 bytes | 296.00 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/mirzaevakamilla/moon.git
91115ff..20e4cdf main -> main

C:\Users\Камилла\moon>git status
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:   README.md

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

C:\Users\Камилла\moon>git add .

C:\Users\Камилла\moon>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   README.md

C:\Users\Камилла\moon>git push
Everything up-to-date

C:\Users\Камилла\moon>git commit -m "change evrything"
[main 37870f8] change evrything
1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\Камилла\moon>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/mirzaevakamilla/moon.git
20e4cdf..37870f8 main -> main

C:\Users\Камилла\moon>
```

Рисунок 5 - скриншот терминала командами Git

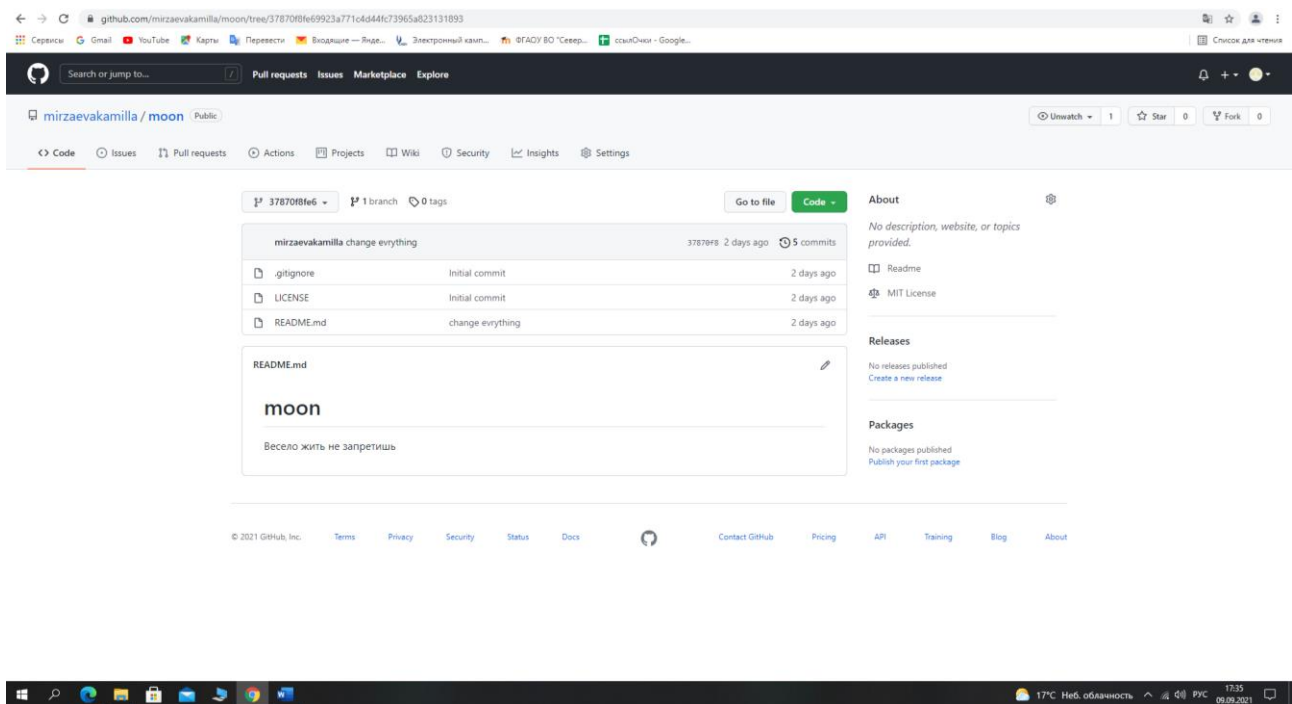


Рисунок 6 - скриншот репозитория после соответствующих изменений

```
Git CMD
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   README.md

C:\Users\Камилла\moon>git push
Everything up-to-date

C:\Users\Камилла\moon>git commit -m "change evrything"
[main 37870f8] change evrything
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\Камилла\moon>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/mirzaevakamilla/moon.git
   20e4cdf..37870f8  main -> main

C:\Users\Камилла\moon>git status
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:   README.md

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

C:\Users\Камилла\moon>git add .

C:\Users\Камилла\moon>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   README.md

C:\Users\Камилла\moon>git commit -m "change readme"
[main dbd71d9] change readme
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\Камилла\moon>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), 300 bytes | 300.00 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/mirzaevakamilla/moon.git
   37870f8..dbd71d9  main -> main

C:\Users\Камилла\moon>
```

Рисунок 7 - скриншот терминала командами Git

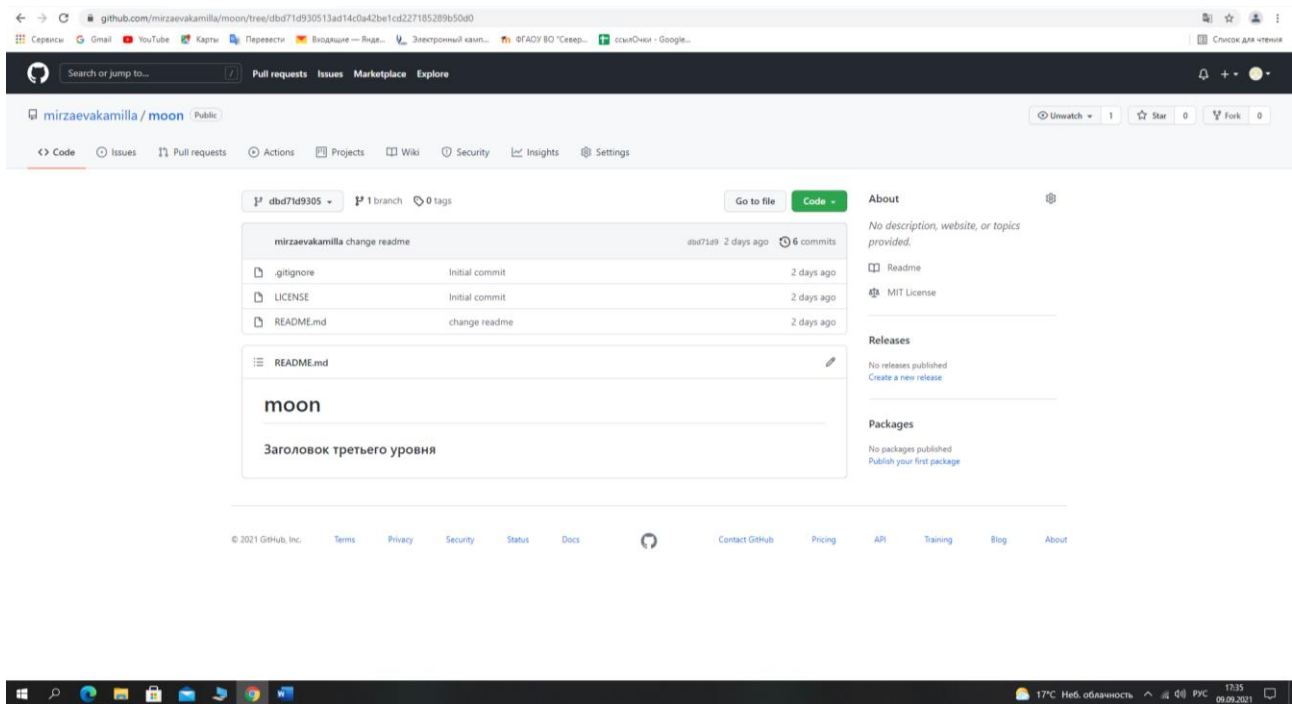


Рисунок 8 - скриншот репозитория после соответствующих изменений

```
C:\Users\Камилла\moon>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md

C:\Users\Камилла\moon>git commit -m "change readme"
[main dbd71d9] change readme
1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\Камилла\moon>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), 300 bytes | 300.00 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/mirzaevakamilla/moon.git
   37870f8..dbd71d9  main -> main

C:\Users\Камилла\moon>git status
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:   README.md

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

C:\Users\Камилла\moon>git add .

C:\Users\Камилла\moon>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md

C:\Users\Камилла\moon>git commit -m "изменение ридми"
[main c8ceafa] изменение ридми
1 file changed, 1 insertion(+), 2 deletions(-)

C:\Users\Камилла\moon>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), 323 bytes | 323.00 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/mirzaevakamilla/moon.git
   dbd71d9..c8ceafa  main -> main

C:\Users\Камилла\moon>
```

Рисунок 9 - скриншот терминала командами Git

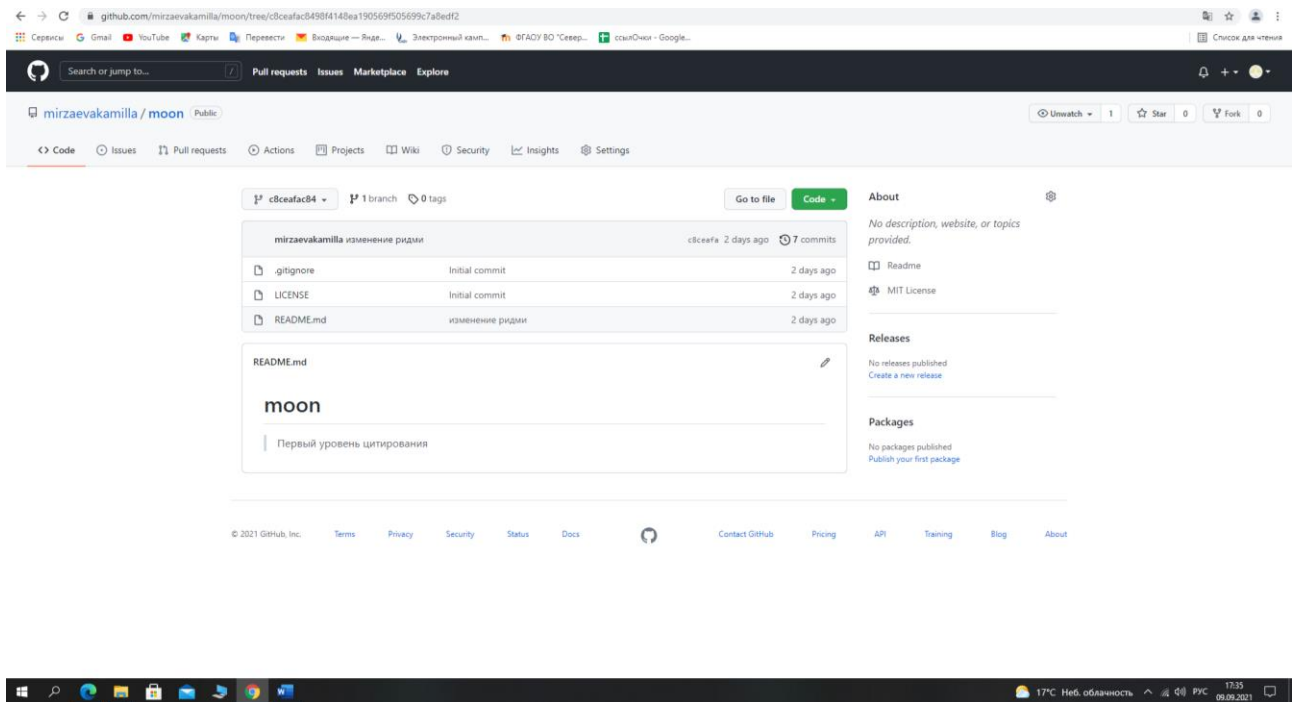


Рисунок 10 - скриншот репозитория после соответствующих изменений

```
Git CMD
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   README.md

C:\Users\Камилла\moon>git commit -m "изменение ридми"
[main c8ceafa] изменение ридми
1 file changed, 1 insertion(+), 2 deletions(-)

C:\Users\Камилла\moon>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), 323 bytes | 323.00 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/mirzaevakamilla/moon.git
   dbd71d9..c8ceafa  main -> main

C:\Users\Камилла\moon>git status
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:   README.md

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

C:\Users\Камилла\moon>git add .

C:\Users\Камилла\moon>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   README.md

C:\Users\Камилла\moon>git commit -m "change"
[main d5af1bf] change
1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\Камилла\moon>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), 297 bytes | 297.00 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/mirzaevakamilla/moon.git
   c8ceafa..d5af1bf  main -> main

C:\Users\Камилла\moon>
```

Рисунок 11 - скриншот терминала командами Git

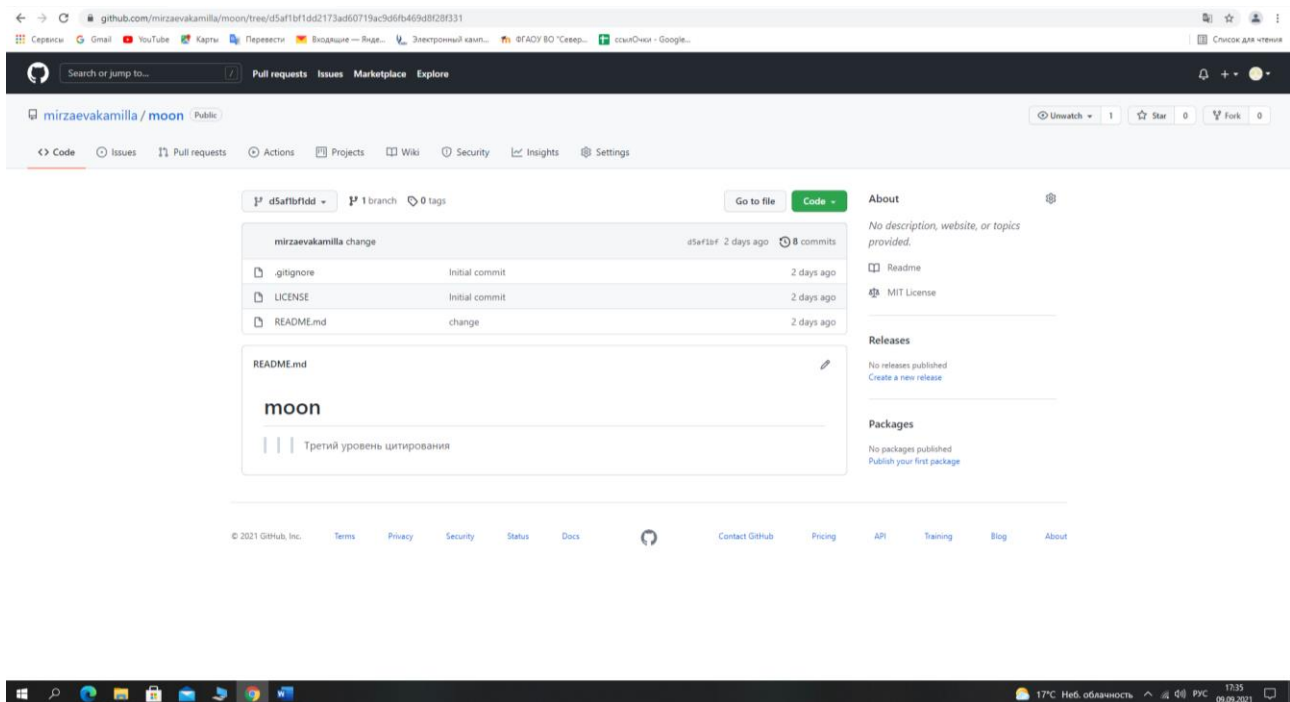


Рисунок 12 - скриншот репозитория после соответствующих изменений

Ответы на вопросы:

1. Что такое СКВ и каково ее назначение?

Система контроля версий (СКВ) — это система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов.

2. В чем недостатки локальных и централизованных СКВ?

Проблемой локальной СКВ является основное свойство — локальность. Она совершенно не предназначена для коллективного использования.

Самый очевидный минус централизованной СКВ - это единая точка отказа, представленная централизованным сервером. Если этот сервер выйдет из строя на час, то в течение этого времени никто не сможет использовать контроль версий для сохранения изменений.

3. К какой СКВ относится Git?

Распределенная система контроля версий.

4. В чем концептуальное отличие Git от других СКВ?

Основное отличие Git от любой другой СКВ— это подход к работе со своими данными. Подход Git к хранению данных больше похож на набор снимков

миниатюрной файловой системы. Каждый раз, когда вы делаете коммит, то есть сохраняете состояние своего проекта в Git, система запоминает, как выглядит каждый файл в этот момент, и сохраняет ссылку на этот снимок.

5. Как обеспечивается целостность хранимых данных в Git?

В Git для всего вычисляется хеш-сумма, и только потом происходит сохранение. В дальнейшем обращение к сохранённым объектам происходит по этой хеш-сумме. Это значит, что невозможно изменить содержимое файла или директории так, чтобы Git не узнал об этом.

6. В каких состояниях могут находиться файлы в Git? Как связаны эти состояния?

У Git есть три основных состояния, в которых могут находиться файлы: зафиксированное (committed), изменённое (modified) и подготовленное (staged).

Зафиксированный значит, что файл уже сохранён в вашей локальной базе. К изменённым относятся файлы, которые поменялись, но ещё не были зафиксированы. Подготовленные файлы — это изменённые файлы, отмеченные для включения в следующий коммит.

7. Что такое профиль пользователя в GitHub?

Профиль - это публичная страница на GitHub, как и в социальных сетях. Когда вы ищете работу в качестве программиста, работодатели могут посмотреть ваш профиль GitHub и принять его во внимание, когда будут решать, брать вас на работу или нет.

8. Какие бывают репозитории в GitHub?

В репозитории можно хранить код, конфигурации, наборы данных, изображения и другие файлы, включенные в проект.

9. Укажите основные этапы модели работы с GitHub.

Стандартный подход к работе с проектом состоит в том, чтобы иметь локальную копию репозитория и фиксировать изменения в этой копии, а не в удаленном репозитории, размещенном на GitHub. Этот локальный репозиторий имеет полную историю версий проекта, которая может быть полезна при разработке без подключения к интернету.

10. Как осуществляется первоначальная настройка Git после установки?

Для этого необходимо перейти на официальный сайт Git <https://git-scm.com/> и скачать версию для операционной системы. После чего необходимо выполнить установку на свой компьютер. Чтобы убедиться, что Git был успешно установлен, нужно ввести команду `git version` ниже в терминале, чтобы отобразить текущую версию Git.

11. Опишите этапы создания репозитория в GitHub.

Имя репозитория. Оно может быть любое, необязательно уникальное во всем github, потому что привязано к аккаунту, но уникальное в рамках тех репозиторий, которые создавали.

Описание (Description). Можно оставить пустым.

Public/private. Выбираем открытый (Public), НЕ ставим галочку “Initialize this repository with a README” (В README потом будет лежать какая-то основная информация, что же такое ваш проект и как с ним работать).

gitignore и LICENSE можно сейчас не выбирать.

После заполнения этих полей нажимаем кнопку Create repository.

12. Какие типы лицензий поддерживаются GitHub при создании репозитория?

Apache, GPL, MIT и другие

13. Как осуществляется клонирование репозитория GitHub? Зачем нужно клонировать репозиторий?

Для этого на странице репозитория необходимо найти кнопку Clone или Code и щелкнуть по ней, чтобы отобразить адрес репозитория для клонирования.

Откройте командную строку или терминал и перейдите в каталог, куда вы хотите скопировать хранилище. Затем напишите git clone и введите адрес. Это необходимо, чтобы иметь локальное хранилище.

14. Как проверить состояние локального репозитория Git?

Необходимо ввести команду git status.

15. Как изменяется состояние локального репозитория Git после выполнения следующих операций:

добавления/изменения файла в локальный репозиторий Git:

```
C:\Users\Камилла\moon>git status
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:   README.md
```

добавления нового/ измененного файла под версионный контроль с помощью команды git add :

```
C:\Users\Камилла\moon>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md
```

фиксации (коммита) изменений с помощью команды `git commit` :

```
C:\Users\Камилла\moon>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
```

отправки изменений на сервер с помощью команды `git push` :

```
C:\Users\Камилла\moon>git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
C:\Users\Камилла\moon>_
```

16. У Вас имеется репозиторий на GitHub и два рабочих компьютера, с помощью которых Вы можете осуществлять работу над некоторым проектом с использованием этого репозитория. Опишите последовательность команд, с помощью которых оба локальных репозитория, связанных с репозиторием GitHub будут находиться в синхронизированном состоянии. Примечание: описание необходимо начать с команды `git clone` .

Необходимо клонировать (`git clone`) исходный репозиторий на каждый из компьютеров

Также на каждом компьютере добавить сразу все измененные файлы в версионный контроль посредством команды `git add`

Зафиксировать изменения с описательным комментарием (`git commit -m`)

Чтобы внести изменения, сделанные кем-то другим, использовать `git pull`

И отправить изменения на сервер (`git push`)

17. GitHub является не единственным сервисом, работающим с Git. Какие сервисы еще Вам известны? Приведите сравнительный анализ одного из таких сервисов с GitHub.

GitLab, SourceForge, BitBucket, Launchpad, Apache Allura, Cloud Source, AWS code commit, FogCreek/DevHub, BeanStalk, GitKraken.

18. Интерфейс командной строки является не единственным и далеко не самым удобным способом работы с Git. Какие Вам известны программные средства с графическим интерфейсом пользователя для работы с Git? Приведите как реализуются описанные в лабораторной работе операции Git с помощью одного из таких программных средств.

GitHub Desktop, GitKraken, SmartGit, SourceTree.

GitHub Desktop

Рабочий стол GitHub в основном является расширением рабочего процесса GitHub. Можно входить в систему, используя свои учетные данные GitHub и начинать работу с репозиториями. Есть возможность создавать новые репозитории, добавлять локальные репозитории и выполнять большинство операций Git из пользовательского интерфейса. GitHub Desktop является полностью открытым исходным кодом, и он доступен для MacOS и Windows.