

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 №1.1**  
**дисциплины**  
**«Основы кроссплатформенного программирования»**

Выполнил:  
Наумов Никита Викторович  
1 курс, группа ИТС-б-о-22-1,  
11.03.02 «Инфокоммуникационные  
технологии и системы связи»,  
направленность (профиль)  
«Инфокоммуникационные системы и  
сети», заочная форма обучения

---

(подпись)

Проверил:  
Воронкин Р. А., доцент кафедры  
инфокоммуникаций

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2023 г.

**Тема:** Исследование основных возможностей Git и GitHub

**Цель:** исследовать базовые возможности системы контроля версий Git и веб-сервиса для хостинга IT-проектов GitHub

**Ход работы:**

Рисунок 1. Репозиторий

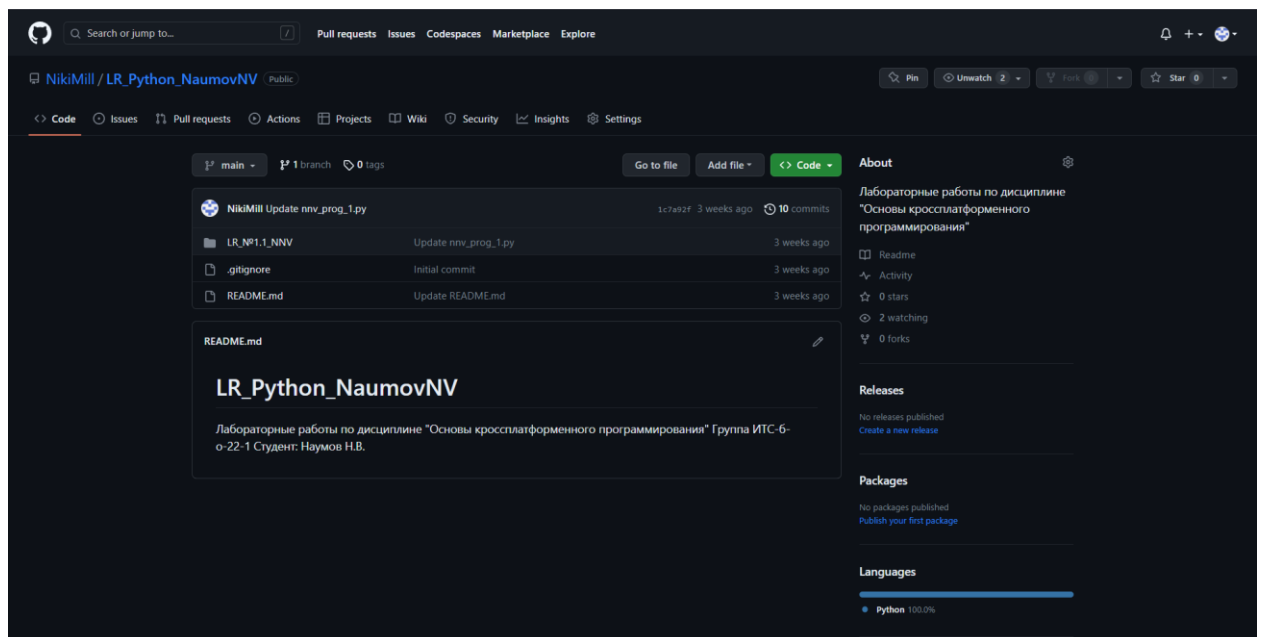


Рисунок 2. Клонирование репозитория.

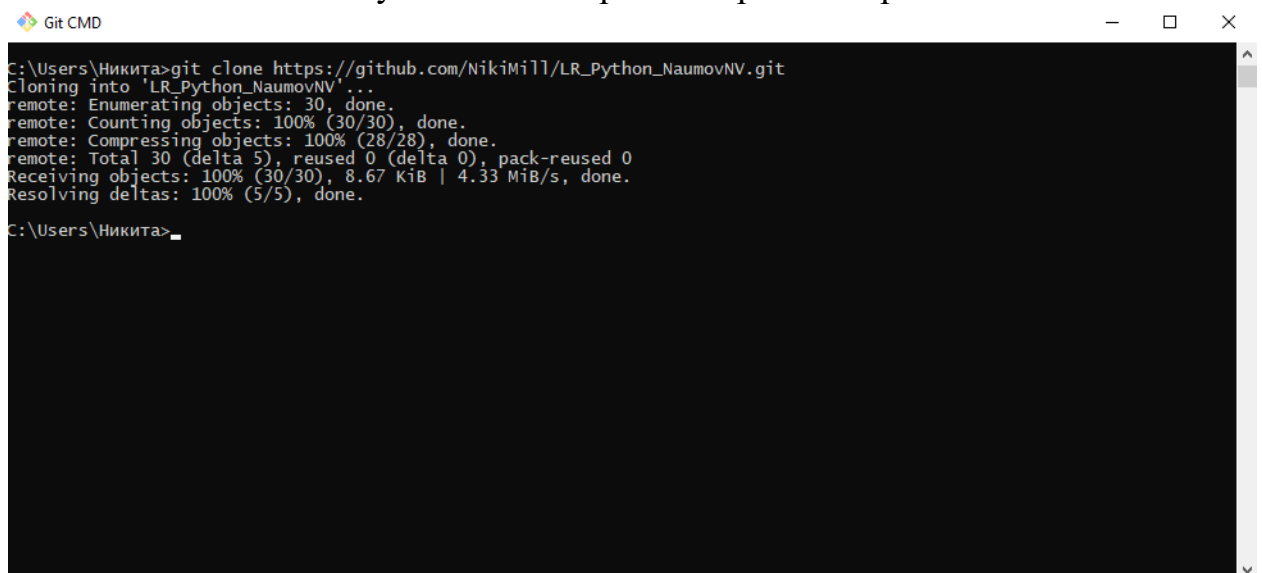
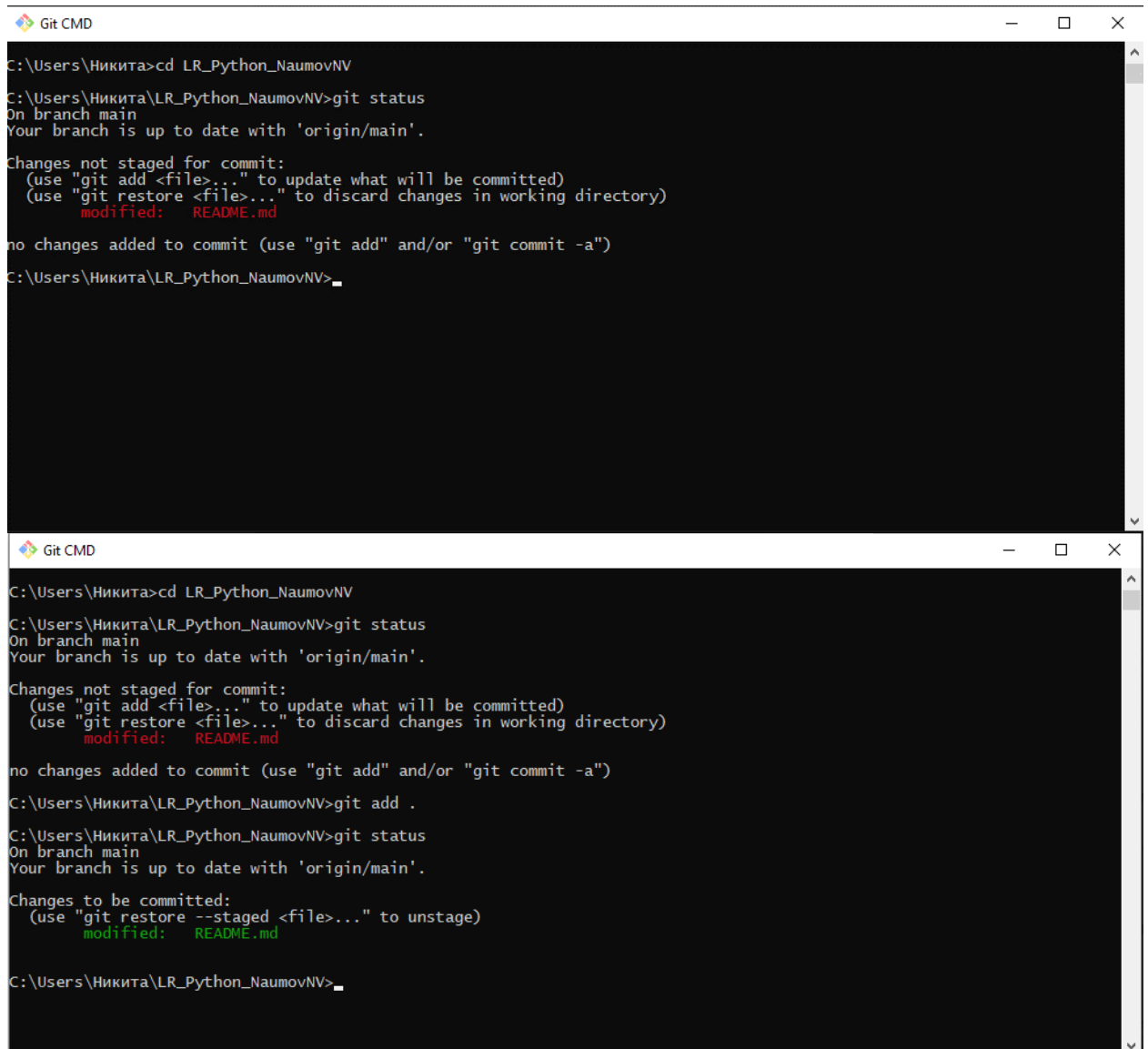


Рисунок 3. Проверка состояния.



The image consists of two screenshots of a Windows command prompt window titled "Git CMD". The window shows the user navigating to the directory "C:\Users\Никита\LR\_Python\_NaumovNV" and running the command "git status". The output indicates that the branch "main" is up to date with "origin/main" and that there are changes not staged for commit, specifically "modified: README.md". The user then runs "git add ." to stage the changes. The second screenshot shows the output of "git status" after the staging process, indicating that the changes are now staged for commit.

```
C:\Users\Никита>cd LR_Python_NaumovNV
C:\Users\Никита\LR_Python_NaumovNV>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

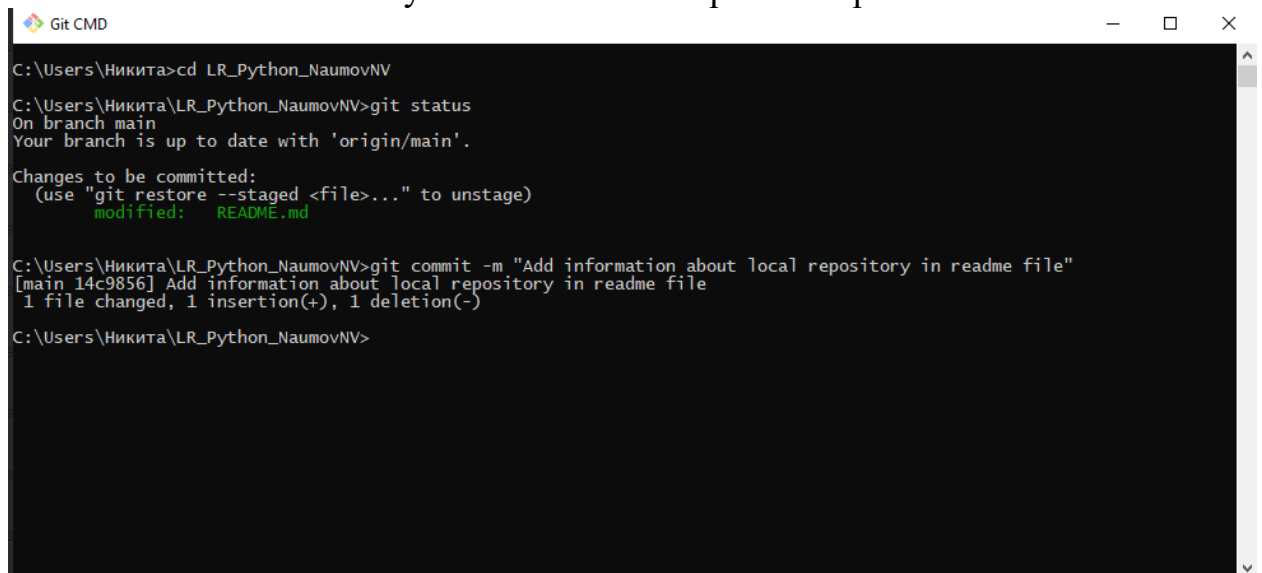
no changes added to commit (use "git add" and/or "git commit -a")
C:\Users\Никита\LR_Python_NaumovNV>

C:\Users\Никита\LR_Python_NaumovNV>git add .
C:\Users\Никита\LR_Python_NaumovNV>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\Никита\LR_Python_NaumovNV>
```

Рисунок 4. Изменение репозитория.



The image shows a single screenshot of a Windows command prompt window titled "Git CMD". The window shows the user running the command "git status" in the directory "C:\Users\Никита\LR\_Python\_NaumovNV". The output indicates that the branch "main" is up to date with "origin/main" and that there are changes to be committed, specifically "modified: README.md". The user then runs the command "git commit -m 'Add information about local repository in readme file'". The output shows the commit hash "[main 14c9856]" and the commit message "Add information about local repository in readme file". It also shows that 1 file changed, with 1 insertion and 1 deletion.

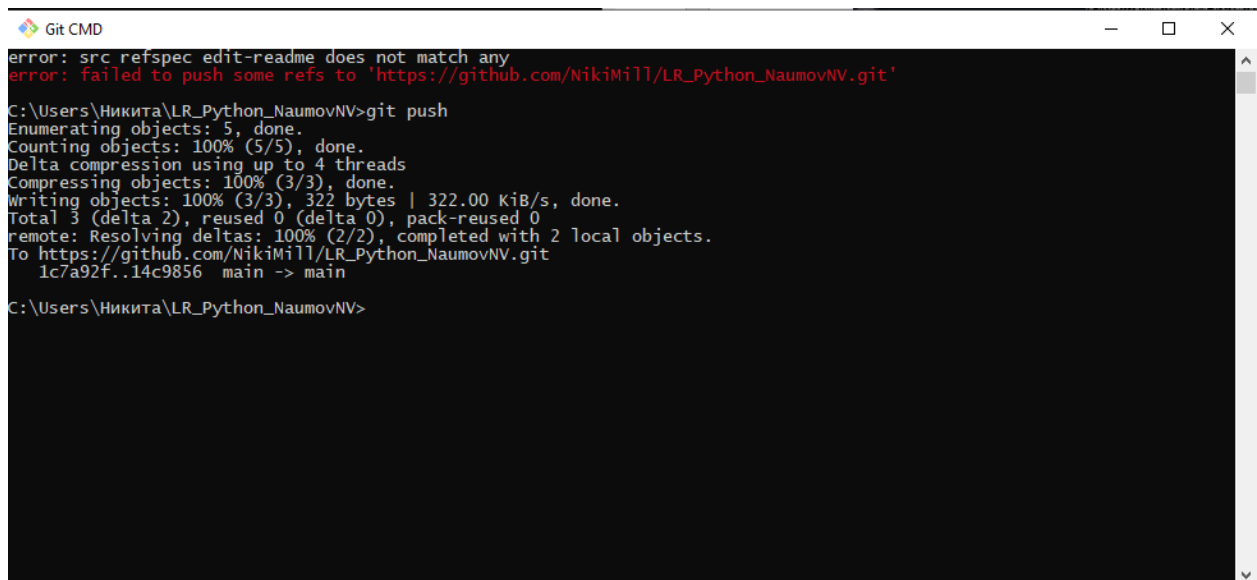
```
C:\Users\Никита>cd LR_Python_NaumovNV
C:\Users\Никита\LR_Python_NaumovNV>git status
On branch main
Your branch is up to date with 'origin/main'.

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

C:\Users\Никита\LR_Python_NaumovNV>git commit -m "Add information about local repository in readme file"
[main 14c9856] Add information about local repository in readme file
1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\Никита\LR_Python_NaumovNV>
```

Рисунок 6. Коммит.



```
Git CMD
error: src refspec edit-readme does not match any
error: failed to push some refs to 'https://github.com/NikiMill/LR_Python_NaumovNV.git'

C:\Users\Никита\LR_Python_NaumovNV>git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 322 bytes | 322.00 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/NikiMill/LR_Python_NaumovNV.git
1c7a92f..14c9856 main -> main

C:\Users\Никита\LR_Python_NaumovNV>
```

Рисунок 5. Git push.

Ссылка: [https://github.com/NikiMill/LR\\_Python\\_NaumovNV.git](https://github.com/NikiMill/LR_Python_NaumovNV.git)

### Ответы на контрольные вопросы:

1. Что такое СКВ и каково ее назначение?

Система контроля версий (СКВ) — это система, регистрирующая изменения в одном или нескольких файлах этих файлов.

2. В чем недостатки локальных и централизованных 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, над которыми работает, а также никто не сможет обмениваться этими изменениями с другими разработчиками.

3. К какой СКВ относится Git?

К распределённым системам контроля версий.

4. В чем концептуальное отличие Git от других СКВ?

Git не хранит и не обрабатывает данные таким же способом как другие СКВ.

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.

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

1) Зафиксированный значит, что файл уже сохранён в вашей локальной базе;

- 2) К изменённым относятся файлы, которые поменялись, но ещё не были зафиксированы;
- 3) Подготовленные файлы — это изменённые файлы, отмеченные для включения в следующий коммит.

7. Что такое профиль пользователя в GitHub?

Профиль - это наша публичная страница на GitHub, как и в социальных сетях. В нем другие пользователи могут посмотреть ваши работы.

8. Какие бывают репозитории в GitHub?

9. Укажите основные этапы модели работы с GitHub.

- 1) Регистрация;
- 2) Создание репозитория;
- 3) Клонирование репозитория;
- 4) Добавление новых файлов.

10. Как осуществляется первоначальная настройка Git после установки?

Убедимся, что Git установлен используя команду: `git version`. Перейдём в папку с локальным репозиторием используя команду: `cd /d <Расположения папки на компьютере>`. Свяжем локальный репозиторий и удалённый командами: `git config --global user.name <YOUR_NAME>` `git config --global user.email <EMAIL>`.

11. Опишите этапы создания репозитория в GitHub.

- 1) В правом верхнем углу, рядом с аватаром есть кнопка с плюсиком, нажимая которую мы переходим к созданию нового репозитория;
- 2) В результате будет выполнен переход на страницу создания репозитория. Наиболее важными на ней являются следующие поля: Имя репозитория. Описание (Description). Public/private. “Initialize this repository with a README” .gitignore и LICENSE.

12. Какие типы лицензий поддерживаются GitHub при создании репозитория?

Microsoft Reciprocal License, The Code Project Open License (CPOL), The Common Development and Distribution License (CDDL), The Microsoft Public License (Ms-PL), The Mozilla Public License 1.1 (MPL 1.1), The Common Public License Version 1.0 (CPL), The Eclipse Public License 1.0, The MIT License, The BSD License, The Apache License, Version 2.0, The Creative Commons Attribution-ShareAlike 2.5 License, The zlib/libpng License, A Public Domain dedication, The Creative Commons Attribution 3.0 Unported License, The Creative Commons).

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

После создания репозитория его необходимо клонировать на ваш компьютер. 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.

14. Как проверить состояние локального репозитория Git?

`git status`

15. Как изменяется состояние локального репозитория Git после выполнения следующих операций: добавления/изменения файла в локальный репозиторий Git; добавления нового/измененного файла под версионный контроль с помощью команды `git add`; фиксации (коммита) изменений с помощью команды `git commit` и отправки изменений на сервер с помощью команды `git push`?

Файлы обновятся на репозитории.

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.

`git clone`.

`git pull`.

17. GitHub является не единственным сервисом, работающим с Git.

Какие сервисы еще Вам известны? Приведите сравнительный анализ одного из таких сервисов с GitHub.

1) GitLab — альтернатива GitHub номер один. GitLab предоставляет не только веб-сервис для совместной работы, но и программное обеспечение с открытым исходным кодом;

2) BitBucket — это служба хостинга репозитория и управления версиями от Atlassian. Она тесно интегрирована с другими инструментами Atlassian — Jira, HipChat и Confluence.

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 это совершенно бесплатное приложение с открытым исходным кодом, разработанное GitHub. С его помощью можно взаимодействовать с GitHub (что и не удивительно), а также с другими платформами (включая Bitbucket и GitLab).

**Вывод:** исследовала базовые возможности системы контроля версий Git и веб-сервиса для хостинга IT-проектов GitHub