

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»

Выполнила:
Медяник Даниил Владимирович
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., 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:

Вариант №10

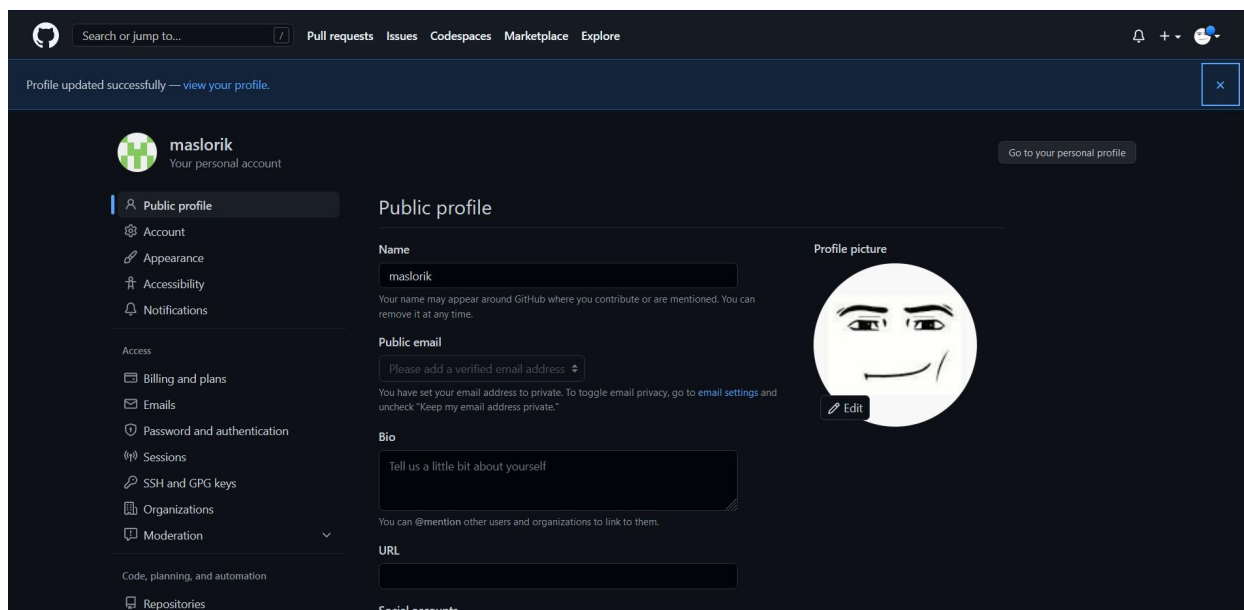


Рисунок 1. Создание аккаунта на GitHub

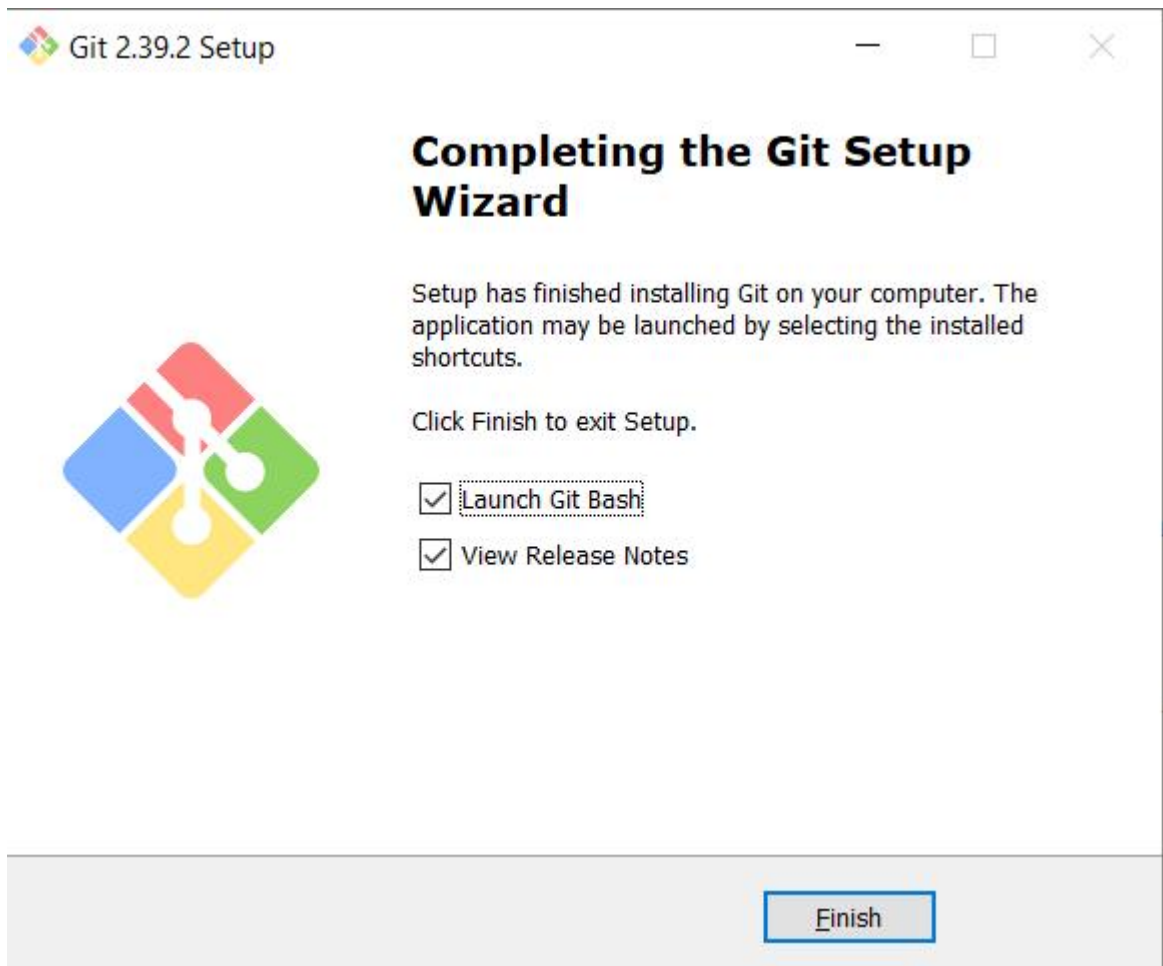


Рисунок 2. Установка Git

The image shows a terminal window with the title bar 'MINGW64:/c/Users/Даниил'. The prompt is 'Даниил@LAPTOP-4FPBSIPF MINGW64 ~'. The user has entered three commands: 'git version', 'git config --global user.name maslorik', and 'git config --global user.email medyanik.danechka@mail.ru'. The output of the first command is 'git version 2.39.2.windows.1'. The cursor is now at the end of the third command line.

```
Даниил@LAPTOP-4FPBSIPF MINGW64 ~
$ git version
git version 2.39.2.windows.1

Даниил@LAPTOP-4FPBSIPF MINGW64 ~
$ git config --global user.name maslorik

Даниил@LAPTOP-4FPBSIPF MINGW64 ~
$ git config --global user.email medyanik.danechka@mail.ru

Даниил@LAPTOP-4FPBSIPF MINGW64 ~
$ |
```

Рисунок 3. Текущая версия Git + добавление имени и электронной почты

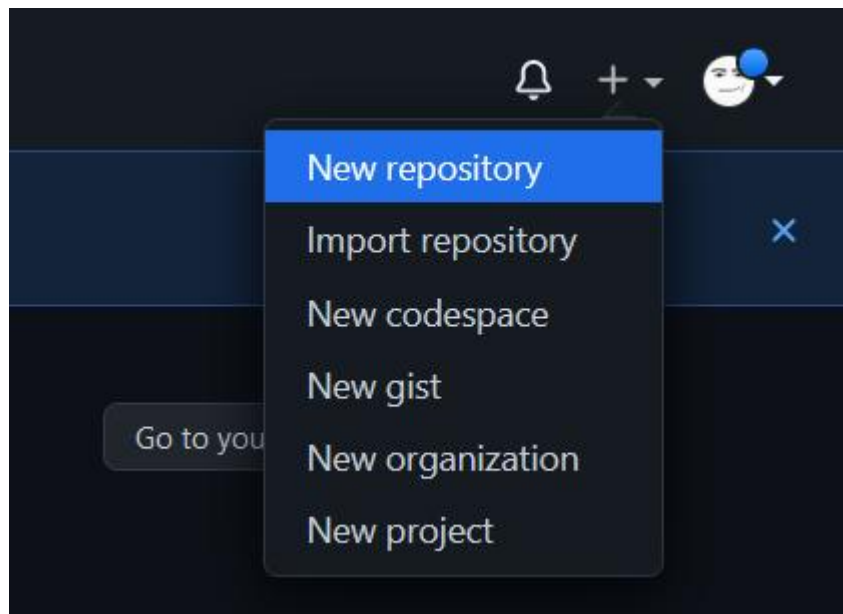




Рисунок 4. Начало создания репозитория

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)


Owner ^{*} / Repository name ^{*}


 maslorik / Лабораторная работа 1.1 

Great repository names ⁴ Your new repository will be created as `-1.1`. How about [super-duper-guide](#)?

Description (optional)

Исследование основных возможностей Git и GitHub

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

`.gitignore template: None`

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

`License: None`

Рисунок 5. Создание репозитория GitHub

```

MINGW64:/c/Users/Даниил/Dmitri/lab1.1

Даниил@LAPTOP-4FPBSIPF MINGW64 ~
$ cd C:/Users/Даниил

Даниил@LAPTOP-4FPBSIPF MINGW64 ~
$ cd C:/Users/Даниил/Dmitri

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri
$ git clone https://github.com/maslorik/lab1.1.git
Cloning into 'lab1.1'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri
$ cd C:/Users/Даниил/Dmitri/lab1.1

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

```

Рисунок 6. Клонирование репозитория

```

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ 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")

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git add README.md

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git add .

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git commit -m "Добавил README"
[main f073a06] Добавил README
1 file changed, 3 insertions(+), 1 deletion(-)

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)

```

Рисунок 7. Изменение файла README, добавление и коммит

```
Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git push
info: please complete authentication in your browser...
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 318 bytes | 159.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/maslorik/lab1.1.git
8c8fa03..f073a06  main -> main
```

Рисунок 8. Git push файла README

```
$ git commit -m "about me"
[main (root-commit) afea295] about me
1 file changed, 1 insertion(+)
create mode 100644 README.md
```

```
Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git add ignore.gitignore

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git add .

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git commit -m "Добавил игнор"
[main 1bed8f7] Добавил игнор
1 file changed, 38 insertions(+)
create mode 100644 ignore.gitignore

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 546 bytes | 182.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/maslorik/lab1.1.git
f073a06..1bed8f7  main -> main
```

Рисунок 9. Добавление файла в репозиторий через команду git add


```

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git add prog1.1.cpp

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git add .

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git commit -m "Добавил библиотеки"
[main a529683] Добавил библиотеки
1 file changed, 3 insertions(+)
create mode 100644 prog1.1.cpp

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 389 bytes | 389.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/maslorik/lab1.1.git
1bed8f7..a529683 main -> main

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)

```

Рисунок 10. Добавление program.cpp, commit 1 и push

```

Даниил@LAPTOP-4FPBSIPF MINGW64 ~
$ cd C:/Users/Даниил/Dmitri/lab1.1

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ vim prog1.1.cpp

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git add .

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ 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:   prog1.1.cpp

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git commit -m "Добавил главную функцию"
[main 18caf8e] Добавил главную функцию
1 file changed, 4 insertions(+), 1 deletion(-)

```

Рисунок 11. commit 2

```
Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git add .

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git commit -m "Поставил русский язык"
[main 51eaa73] Поставил русский язык
1 file changed, 4 insertions(+), 1 deletion(-)
```

Рисунок 12. commit 3

```
Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git add .

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git commit -m "Добавил x"
[main 482e603] Добавил x
1 file changed, 3 insertions(+)
```

Рисунок 13. commit 4

```
Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git add .

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git commit -m "Добавил y"
[main e4fdc13] Добавил y
1 file changed, 2 insertions(+)
```

Рисунок 14. commit 5

```
Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git add .

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git commit -m "Ввёл значения"
[main dc9b27b] Ввёл значения
1 file changed, 5 insertions(+)
```

Рисунок 15. commit 6

```
Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git add .

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git commit -m "Добавил условие if"
[main 4b6d9ef] Добавил условие if
1 file changed, 9 insertions(+)
```

Рисунок 16. commit 7


```
Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git add .

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git commit -m "Добавил условие else"
[main b0bba03] Добавил условие else
1 file changed, 11 insertions(+)

Даниил@LAPTOP-4FPBSIPF MINGW64 ~/Dmitri/lab1.1 (main)
$ git push origin main
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 12 threads
Compressing objects: 100% (21/21), done.
Writing objects: 100% (21/21), 2.08 KiB | 1.04 MiB/s, done.
Total 21 (delta 13), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (13/13), completed with 1 local object.
To https://github.com/maslorik/lab1.1.git
a529683..b0bba03 main -> main
```

Рисунок 17. commit 8 и push origin main

Ссылка:

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 (CPOPL), 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: исследовал базовые возможности системы контроля версий Git и веб-сервиса для хостинга IT-проектов GitHub.