

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
дисциплины «Основы кроссплатформенного программирования»

Вариант ____

Выполнила:
Маньшина Дарья Алексеевна
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 кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

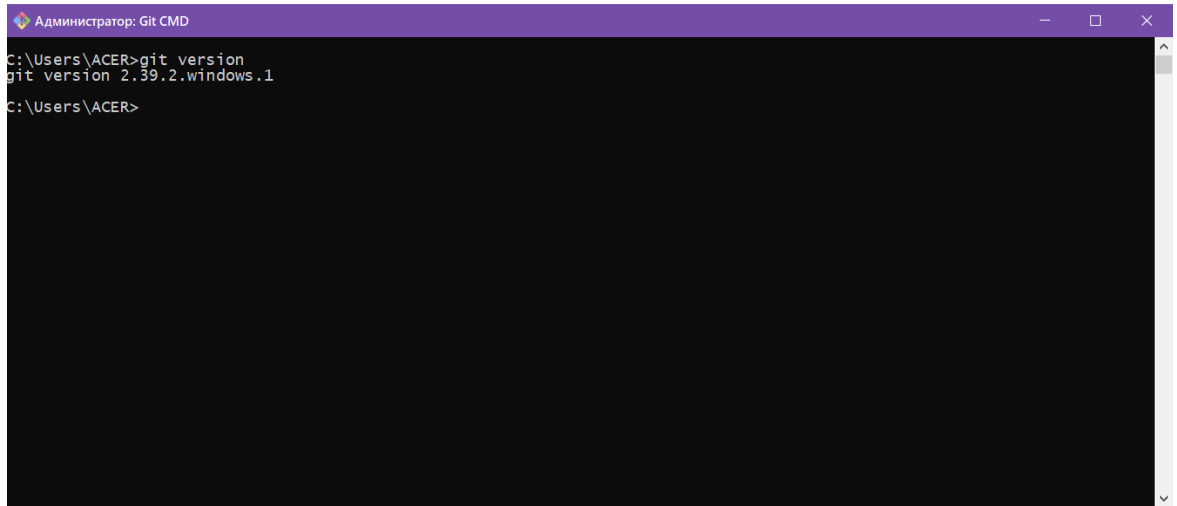
Ставрополь, 2023 г.

Тема: Исследование основных возможностей Git и GitHub

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

Ход работы:

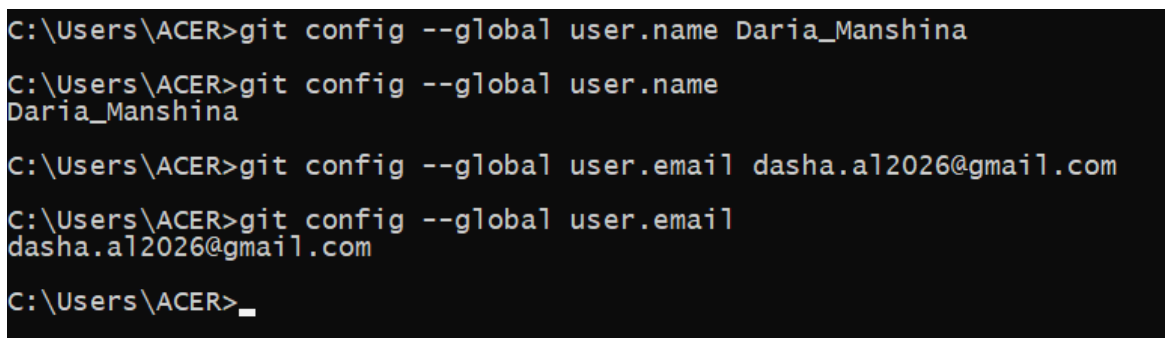
1. Открыв Git CDM, проверяем его работу, задав команду git version.



```
Администратор: Git CMD
C:\Users\ACER>git version
git version 2.39.2.windows.1
C:\Users\ACER>
```

Рисунок 1 – Использование команды git version.

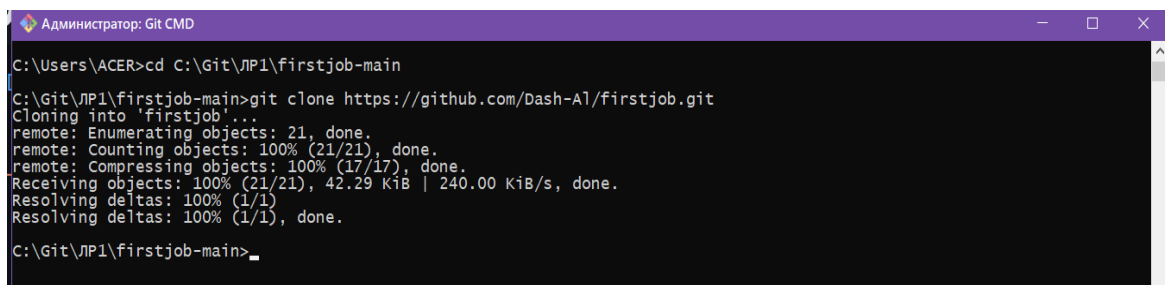
2. При помощи git config --global добавим своё имя и адрес электронной почты.



```
C:\Users\ACER>git config --global user.name Daria_Manshina
C:\Users\ACER>git config --global user.name
Daria_Manshina
C:\Users\ACER>git config --global user.email dasha.a12026@gmail.com
C:\Users\ACER>git config --global user.email
dasha.a12026@gmail.com
C:\Users\ACER>_
```

Рисунок 2 – Добавление имени и электронной почты.

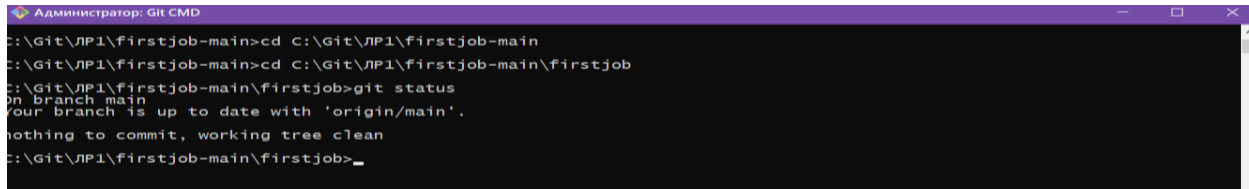
3. Выполнение клонирования репозитория



```
Администратор: Git CMD
C:\Users\ACER>cd C:\Git\ЛР1\firstjob-main
C:\Git\ЛР1\firstjob-main>git clone https://github.com/Dash-A1/firstjob.git
Cloning into 'firstjob'...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (17/17), done.
Receiving objects: 100% (21/21), 42.29 KiB | 240.00 KiB/s, done.
Resolving deltas: 100% (1/1)
Resolving deltas: 100% (1/1), done.
C:\Git\ЛР1\firstjob-main>_
```

Рисунок 3 – Клонирование.

4. Проверим состояния репозитория при помощи git status.



```
Администратор: Git CMD
C:\Git\ЛР1\firstjob-main>cd C:\Git\ЛР1\firstjob-main
C:\Git\ЛР1\firstjob-main>cd C:\Git\ЛР1\firstjob-main\firstjob
C:\Git\ЛР1\firstjob-main\firstjob>git status
On branch main
Your branch is up to date with 'origin/main'.
nothing to commit, working tree clean
C:\Git\ЛР1\firstjob-main\firstjob>
```

Рисунок 4 – Использование «git status».

5. Добавление .gitignore

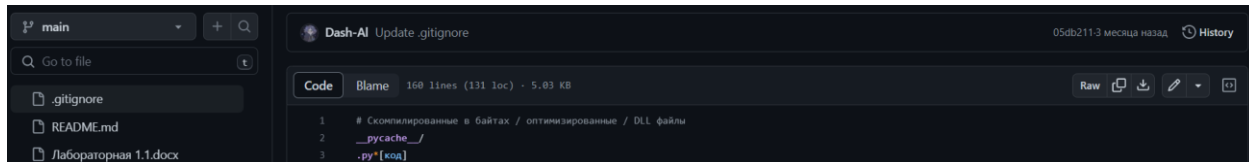


Рисунок 5 - .gitignore

6. Добавим в файл README.md информацию о своей группе и ФИО.

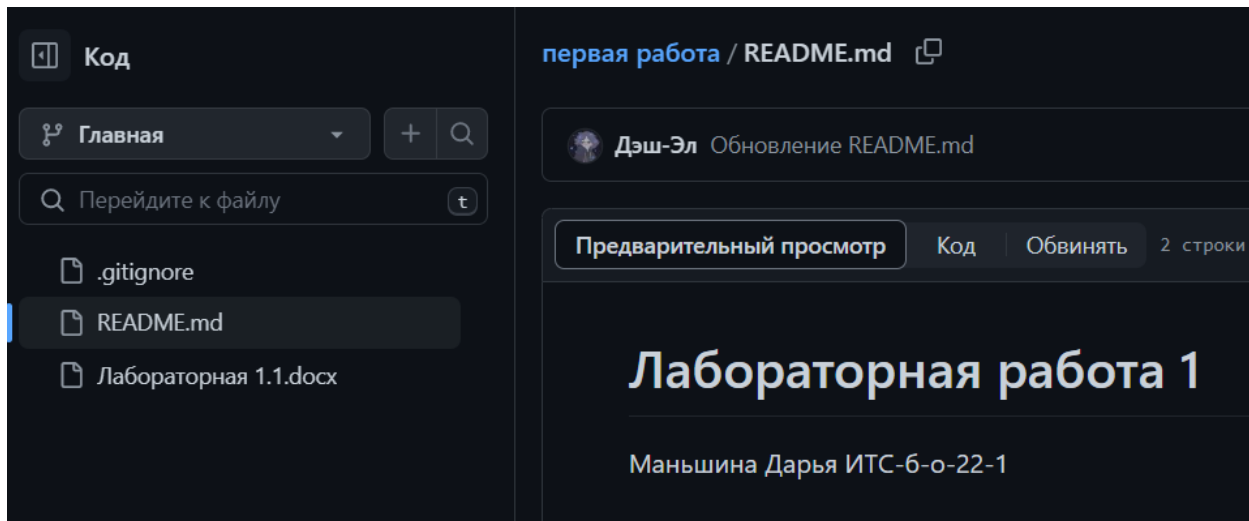


Рисунок 6 - файл README.md с информацией о себе.

7. Написание небольшой программы. Сделаны 7 коммитов.

```
Администратор: Git CMD
C:\Git\LP1\firstjob-main\firstjob>git add programma
fatal: pathspec 'programma' did not match any files
C:\Git\LP1\firstjob-main\firstjob>git add programma.txt
warning: in the working copy of 'programma.txt', LF will be replaced by CRLF the next time Git touches it
C:\Git\LP1\firstjob-main\firstjob>git commit -m "Add main"
[main 8661176] Add main
1 file changed, 1 insertion(+), 1 deletion(-)
C:\Git\LP1\firstjob-main\firstjob>git add programma.txt
warning: in the working copy of 'programma.txt', LF will be replaced by CRLF the next time Git touches it
C:\Git\LP1\firstjob-main\firstjob>git commit -m "Add main"
[main 5eac8fe] Add main
1 file changed, 2 insertions(+), 2 deletions(-)
C:\Git\LP1\firstjob-main\firstjob>git add programma.txt
warning: in the working copy of 'programma.txt', LF will be replaced by CRLF the next time Git touches it
C:\Git\LP1\firstjob-main\firstjob>git commit -m "Add set"
[main b9a1941] Add set
1 file changed, 4 insertions(+), 3 deletions(-)
C:\Git\LP1\firstjob-main\firstjob>git add programm.txt
fatal: pathspec 'programm.txt' did not match any files
C:\Git\LP1\firstjob-main\firstjob>git add programma.txt
warning: in the working copy of 'programma.txt', LF will be replaced by CRLF the next time Git touches it
C:\Git\LP1\firstjob-main\firstjob>git commit -m "Add p"
[main 3b9b0b6] Add p
1 file changed, 2 insertions(+), 1 deletion(-)
C:\Git\LP1\firstjob-main\firstjob>git add programma.txt
warning: in the working copy of 'programma.txt', LF will be replaced by CRLF the next time Git touches it
C:\Git\LP1\firstjob-main\firstjob>git commit -m "Add m"
[main 2bb12a1] Add m
1 file changed, 2 insertions(+), 1 deletion(-)
C:\Git\LP1\firstjob-main\firstjob>git add programma.txt
warning: in the working copy of 'programma.txt', LF will be replaced by CRLF the next time Git touches it
C:\Git\LP1\firstjob-main\firstjob>git commit -m "Add dd"
[main 6023cce] Add dd
1 file changed, 2 insertions(+), 1 deletion(-)
C:\Git\LP1\firstjob-main\firstjob>git add programma.txt
warning: in the working copy of 'programma.txt', LF will be replaced by CRLF the next time Git touches it
C:\Git\LP1\firstjob-main\firstjob>git commit -m "Add dr"
[main 127d60d] Add dr
1 file changed, 5 insertions(+), 6 deletions(-)
```

Рисунок 7 – Добавление коммитов.

Вывод: в ходе лабораторной работы я исследовала базовые возможности системы контроля версий Git и веб-сервиса для хостинга IT-проектов GitHub.

Ответы на вопросы:

1. **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. *СКВ даёт возможность* возвращать отдельные файлы к прежнему виду, возвращать к прежнему состоянию весь проект, просматривать происходящие со временем изменения, определять, кто последним вносил изменения во внезапно переставший работать модуль, кто и когда внёс в код какую-то ошибку, и многое другое.

2. Самый очевидный минус — 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. Локальные СКВ страдают от той же самой проблемы: когда вся история проекта хранится в одном месте, вы рискуете потерять всё.

3. Git относится к распределённой системе контроля версий (РСКВ).

4. Основное отличие Git от любой другой СКВ (включая Subversion и её собратьев) — 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. 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. Механизм, которым пользуется Git при вычислении хеш-сумм, называется SHA-1 хеш. Это строка длиной в 40 шестнадцатеричных символов (0-9 и a-f), она вычисляется на основе содержимого файла или структуры каталога.

6. У Git есть три основных состояния, в которых могут находиться ваши 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. Профиль - это ваша 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. Репозитории в GitHub бывают локальными и удалёнными.

9. Основные этапы модели работы с GitHub: регистрация, создание репозитория, клонирование репозитория, добавление новых файлов.

10. Чтобы убедиться, что Git был успешно установлен, введём команду `git version` в терминале, чтобы отобразить текущую версию вашего Git. Если она сработала, добавим в настройки Git наше имя, фамилию и адрес электронной почты, связанный с нашей учетной записью GitHub: `git config --global user.name <YOUR_NAME>; git config --global user.email <EMAIL>`

11. В правом верхнем углу, рядом с аватаром есть кнопка с плюсом, нажимая которую мы переходим на страницу к созданию нового репозитория. Наиболее важными на ней являются следующие поля: имя репозитория, описание (Description), public/private. “Initialize this repository with a README” .gitignore и LICENSE.

12. 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. После создания репозитория его необходимо клонировать на наш 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. Чтобы проверить состояние локального репозитория Git, нужно ввести: `git status`

15. После: 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. Последовательность команд, с помощью которых оба локальных репозитория, связанных с репозиторием GitHub будут находиться в синхронизированном состоянии: `git clone`; `git pull`.

17. GitLab — альтернатива GitHub номер один. GitLab предоставляет не только веб-сервис для совместной работы, но и программное обеспечение с открытым исходным кодом; BitBucket — это служба хостинга репозитория и управления версиями от Atlassian. Она тесно интегрирована с другими инструментами Atlassian — Jira, HipChat и Confluence.

18. 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).