

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:  
Ермолович Иван Денисович  
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)

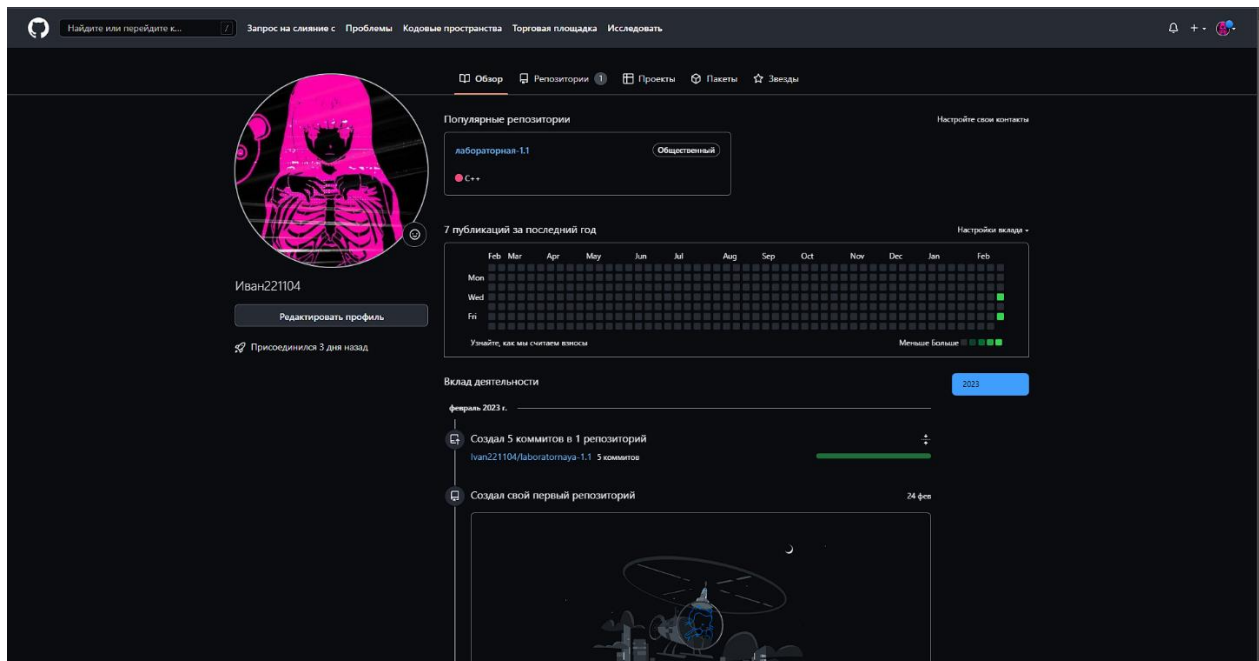
Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

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:**



Скриншот 1. Профиль GitHub.



Скриншот 2. Скачивание терминала Git Bash.

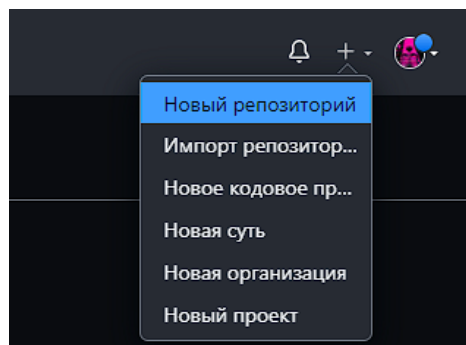
```
MINGW64:/c/laboratornaya-1.1

a234e@LAPTOP-PKMVGVLJ MINGW64 ~
$ git version
git version 2.39.2.windows.1

a234e@LAPTOP-PKMVGVLJ MINGW64 ~
$ git config --global user.name Ivan 221104

a234e@LAPTOP-PKMVGVLJ MINGW64 ~
$ git config --global user.email a234ef5@gmail.com
```


Скриншот 3. Проверка версии, синхронизация имени и адреса электронной почты.



Скриншот 4. Создание нового репозитория.


**Создать новый репозиторий**  
Репозиторий содержит все файлы проекта, включая историю изменений. У вас уже есть репозиторий проекта в другом месте? [Импорт репозитория.](#)


Владелец \* / Имя репозитория \*

 Иван221104 /

Отличные имена репозитория. Your new repository will be created as -1.1. Нужна вдохновение? Как насчет анимированного суккоташа ?

Описание (необязательно)

☒  **Общественный**  
Любой пользователь Интернета может видеть этот репозиторий. Вы выбираете, кто может совершить.


☐  **Частный**  
Вы выбираете, кто может видеть и фиксировать этот репозиторий.

Инициализируйте этот репозиторий с помощью:  
Пропустите этот шаг, если вы импортируете существующий репозиторий.

☐ **Добавьте файл README**  
Здесь вы можете написать подробное описание вашего проекта. [Узнать больше.](#)

**Добавить .gitignore**  
Выберите, какие файлы не отслеживать из списка шаблонов. [Узнать больше.](#)

**Выберите лицензию**  
Лицензия говорит другим, что они могут и не могут делать с вашим кодом. [Узнать больше.](#)

 You are creating a public repository in your personal account.

Скриншот 5. Даём название новому репозиторию.

```

a234e@LAPTOP-PKMVGVLJ MINGW64 ~
$ cd ..

a234e@LAPTOP-PKMVGVLJ MINGW64 /c/Users
$ cd ..

a234e@LAPTOP-PKMVGVLJ MINGW64 /c
$ git clone https://github.com/Ivan221104/laboratornaya-1.1.git
Cloning into 'laboratornaya-1.1'...
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.

a234e@LAPTOP-PKMVGVLJ MINGW64 /c
$ cd c:/laboratornaya-1.1

a234e@LAPTOP-PKMVGVLJ MINGW64 /c/laboratornaya-1.1 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

```

Скриншот 6. Клонирование репозитория.

```

a234e@LAPTOP-PKMVGVLJ MINGW64 /c/laboratornaya-1.1 (main)
$ git add README.md

a234e@LAPTOP-PKMVGVLJ MINGW64 /c/laboratornaya-1.1 (main)
$ git add .

a234e@LAPTOP-PKMVGVLJ MINGW64 /c/laboratornaya-1.1 (main)
$ git commit -m "Add information about local repository in readme file"
[main cda8a7a] Add information about local repository in readme file
1 file changed, 2 insertions(+), 1 deletion(-)

a234e@LAPTOP-PKMVGVLJ MINGW64 /c/laboratornaya-1.1 (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 6 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 335 bytes | 335.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Ivan221104/laboratornaya-1.1.git
8a9fbd7..cda8a7a main -> main

```

Скриншот 7. Изменение файла README, коммит и Git Push README.

```

a234e@LAPTOP-PKMVGVLJ MINGW64 /c/laboratornaya-1.1 (main)
$ git add sfregreg.gitignore

a234e@LAPTOP-PKMVGVLJ MINGW64 /c/laboratornaya-1.1 (main)
$ git add .

a234e@LAPTOP-PKMVGVLJ MINGW64 /c/laboratornaya-1.1 (main)
$ git commit -m "Add a gitignore file"
[main 4c3fc33] Add a gitignore file
1 file changed, 38 insertions(+)
create mode 100644 sfregreg.gitignore

a234e@LAPTOP-PKMVGVLJ MINGW64 /c/laboratornaya-1.1 (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 6 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 520 bytes | 520.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Ivan221104/laboratornaya-1.1.git
cda8a7a..4c3fc33 main -> main

```

Скриншот 8. Добавление файла Gitignore.

```

a234e@LAPTOP-PKMVGVLJ MINGW64 /c/laboratornaya-1.1 (main)
$ git add progrkod.cpp

a234e@LAPTOP-PKMVGVLJ MINGW64 /c/laboratornaya-1.1 (main)
$ git add .

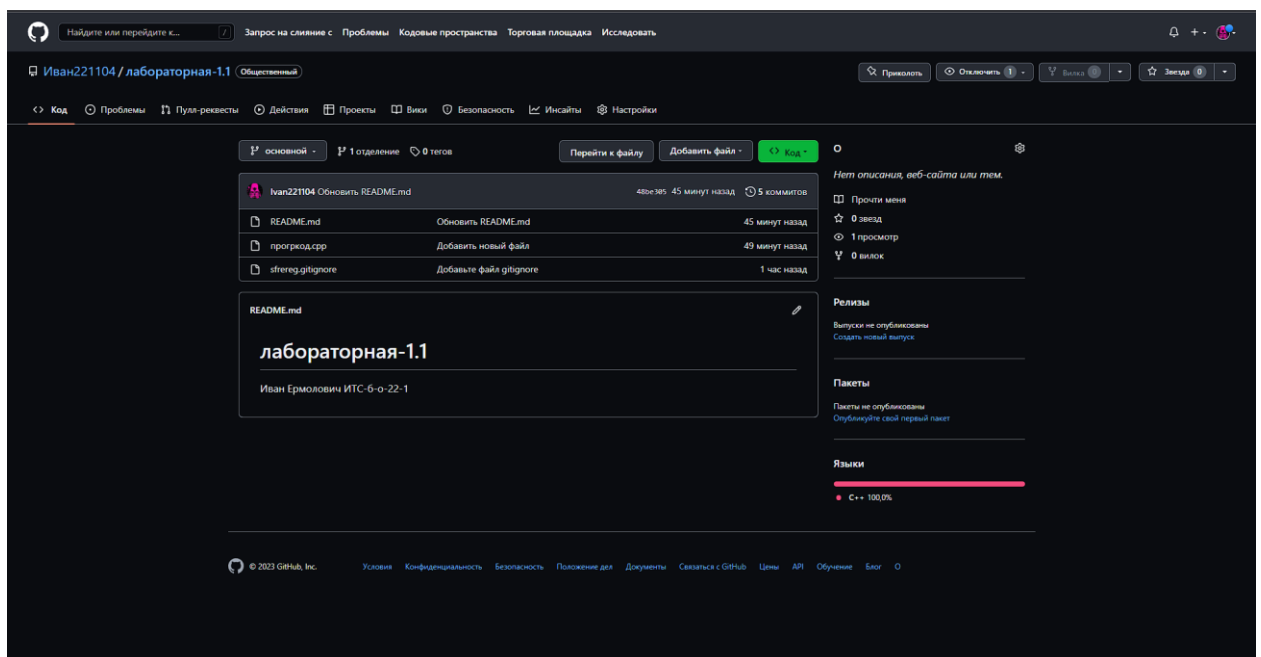
a234e@LAPTOP-PKMVGVLJ MINGW64 /c/laboratornaya-1.1 (main)
$ git commit -m "Add a new file"
[main 07968cf] Add a new file
1 file changed, 32 insertions(+)
create mode 100644 progrkod.cpp

a234e@LAPTOP-PKMVGVLJ MINGW64 /c/laboratornaya-1.1 (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 6 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 816 bytes | 816.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Ivan221104/laboratornaya-1.1.git
4c3fc33..07968cf main -> main

a234e@LAPTOP-PKMVGVLJ MINGW64 /c/laboratornaya-1.1 (main)
$

```

Скриншот 9. Добавление файла с программой, коммита и Git Push.



Скриншот 10. Удалённый репозиторий.

Ссылка: <https://github.com/Ivan221104/laboratornaya-1.1>

## **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 (CPO), 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.