

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития  
Кафедра инфокоммуникаций

**«Исследование основных возможностей Git и GitHub»**

**ОТЧЕТ**  
**по лабораторной работе №1**  
**дисциплины**  
**«Основы программной инженерии»**

Выполнил:  
Коньжев Максим Викторович  
2 курс, группа ПИЖ-б-о-21-1,  
09.03.04 «Программная инженерия»,  
направленность (профиль) «Разработка  
и сопровождение программного  
обеспечения», очная форма обучения

---

(подпись)

Проверил:

---

(подпись)

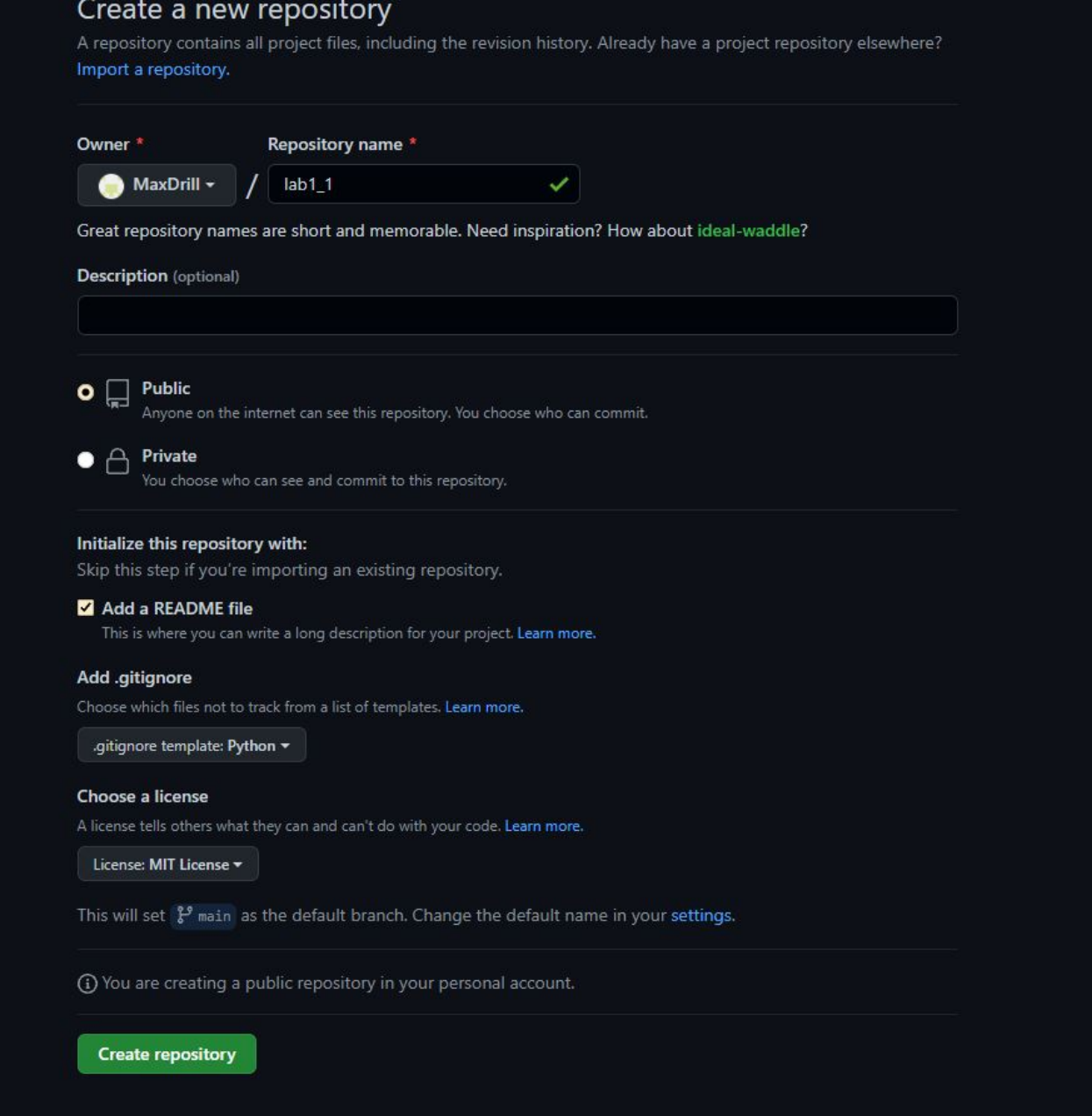
Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2022 г.

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

### Выполнение работы:

Создание репозитория:



The screenshot shows the 'Create a new repository' page on GitHub. At the top, it says 'Create a new repository' and provides a brief explanation of what a repository is. Below this, there are two main input fields: 'Owner' and 'Repository name'. The 'Owner' field is set to 'MaxDrill' and the 'Repository name' field is set to 'lab1\_1'. Below these fields, there is a section for 'Description (optional)' with a text input area. Further down, there are two radio button options for repository visibility: 'Public' (selected) and 'Private'. Below these, there is a section for 'Initialize this repository with:' which includes a checkbox for 'Add a README file' (checked) and a dropdown for '.gitignore template' set to 'Python'. There is also a section for 'Choose a license' with a dropdown set to 'MIT License'. At the bottom, there is a green button labeled 'Create repository'.

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 \*

MaxDrill / lab1\_1

Great repository names are short and memorable. Need inspiration? How about [ideal-waddle?](#)

Description (optional)

☒ 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: Python

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

License: MIT License

This will set `main` as the default branch. Change the default name in your [settings](#).

☐ You are creating a public repository in your personal account.

Create repository

Рисунок 2 – Настройка репозитория

Клонирование репозитория:

```
C:\Users\UESR>git clone https://github.com/MaxDrill/lab1_1.git
Cloning into 'lab1_1'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
C:\Users\UESR>
```

Рисунок 3 – Создание локального хранилища проекта

Проверка состояния репозитория:

```
C:\Users\UESR>cd C:\Users\UESR\lab1_1

C:\Users\UESR\lab1_1>git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
C:\Users\UESR\lab1_1>
```

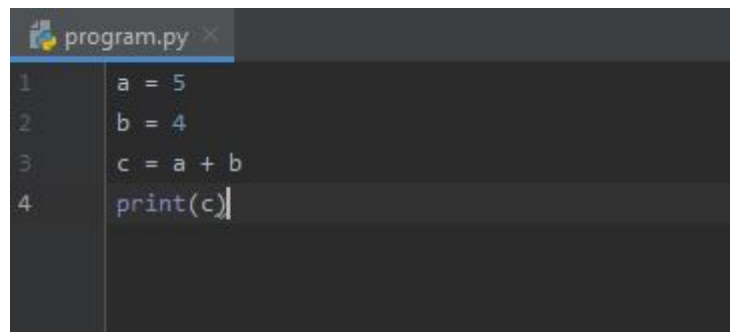
Рисунок 4 – Статус репозитория

Добавление в файл README.md информации о группе и ФИО студента, выполняющего лабораторную работу:



Рисунок 5 - Окно блокнота

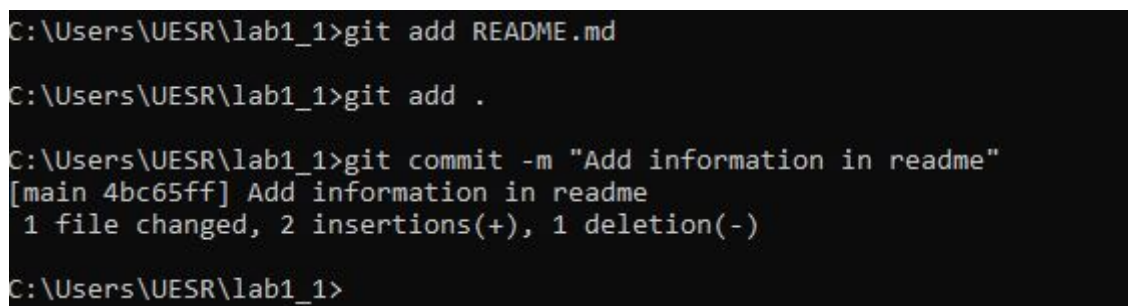
Написание небольшой программы с фиксацией изменений и созданием КОММИТОВ



```
program.py x
1 a = 5
2 b = 4
3 c = a + b
4 print(c)
```

Рисунок 6 - Окно IDE

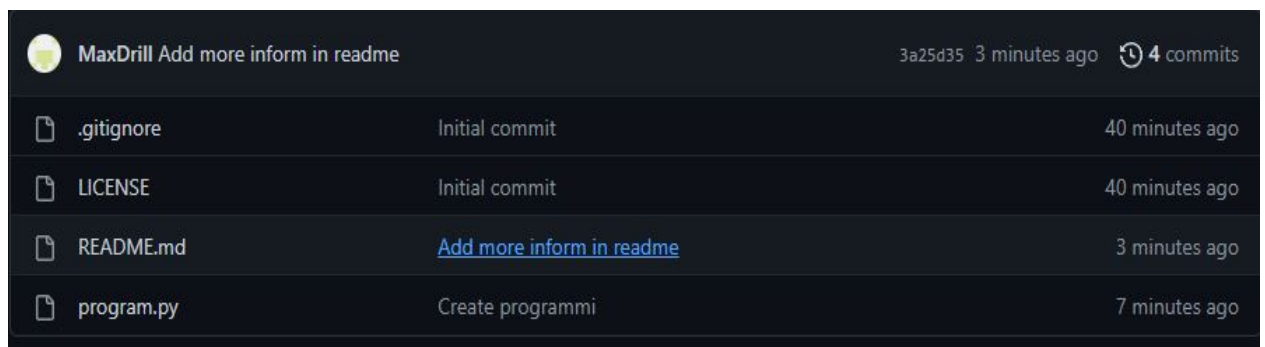
Добавление коммита в README.md:



```
C:\Users\UESR\lab1_1>git add README.md
C:\Users\UESR\lab1_1>git add .
C:\Users\UESR\lab1_1>git commit -m "Add information in readme"
[main 4bc65ff] Add information in readme
1 file changed, 2 insertions(+), 1 deletion(-)
C:\Users\UESR\lab1_1>
```

Рисунок 7 – Статус репозитория после редактирования

Скриншот окна браузера изменённого репозитория:








 MaxDrill	Add more inform in readme	3a25d35 3 minutes ago	🕒 4 commits
	.gitignore	Initial commit	40 minutes ago
	LICENSE	Initial commit	40 minutes ago
	README.md	<a href="#">Add more inform in readme</a>	3 minutes ago
	program.py	Create programmi	7 minutes ago

Рисунок 8 – Изменённый репозиторий

Ответы на вопросы:

1. Система контроля версий — это система, записывающая изменения в файл или набор файлов в течение времени и позволяющая вернуться позже к определённой версии.

2. Недостатки локальной системы: возможность потери данных вследствие возникновения физических поломок оборудования. Отсутствие возможности совместной разработки.

Недостатки централизованной системы: отсутствие доступа к данным при сбое работы сервера. Снижение скорости за счёт сетевых задержек.

3. Git относится к распределённой система контроля версий.

4. Подход Git к хранению данных больше похож на набор снимков миниатюрной файловой системы. Каждый раз, когда вы делаете коммит, то есть сохраняем состояние своего проекта в Git, система запоминает, как выглядит каждый файл в этот момент, и сохраняет ссылку на этот снимок. Для увеличения эффективности, если файлы не были изменены, Git не запоминает эти файлы вновь, а только создаёт ссылку на предыдущую версию идентичного файла, который уже сохранён.

6. Файл может находится в двух состояниях, отслеживаемом и не отслеживаемом. В первом случае это те файлы, которые были в последнем снимке состояния проекта, а во втором файлы, которые не входили в последний снимок или не были подготовлены к коммиту.

7. Профиль – это публичная страница на GitHub, как и в социальных сетях. Работодатели могут посмотреть профиль GitHub и принять его во внимание, когда будут решать, брать вас на работу или нет.

8. Публичные и приватные.

9. Модель 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. После того, как мы что-то изменили в локальном, мы можем отправить свои изменения в удаленный репозиторий, чтобы сделать их видимыми для других разработчиков.

10. При помощи команды «git version» можно убедиться в том, что Git был установлен. Далее нужно установить связь с профилем в GitHub для этого используем команды «git config --global user.name <YOUR\_NAME>» и «git config --global user.email <EMAIL>»

11. Для создания репозитория на GitHub нужно нажать на кнопку «New repository» после чего задать его имя, выбрать будет ли репозиторий приватным или публичным и при необходимости установить флажки в пунктах создания «.gitignore» и «LICENSE»
12. MIT, Академическая бесплатная лицензия, Apache license 2.0, Artistic license 2.0 и множество других лицензий.
13. Создание локального хранилища проекта происходит при помощи команды «git clone <Ссылка на репозиторий GitHub>»
14. При помощи команды «git status»
15. Добавится/изменится файл в локальном репозитории.
16. Оба компьютера должны создать локальную версию репозитория при помощи команды «git clone» после чего при изменении или добавлении файлов с одного компьютера следует обновить отслеживаемые файлы путем использования команды «git add .», добавлением коммита командой «git commit» и выгрузкой во внешний репозиторий командой «git push», на другом компьютере необходимо использовать команду «git pull» для обновления файлов в локальном репозитории.
17. GitLab и BitBucket
18. GitKraken и SourceTree

Ссылка на репозиторий: [https://github.com/MaxDrill/lab1\\_1](https://github.com/MaxDrill/lab1_1)