

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:  
Дмитриченко Марат Заурович  
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:**

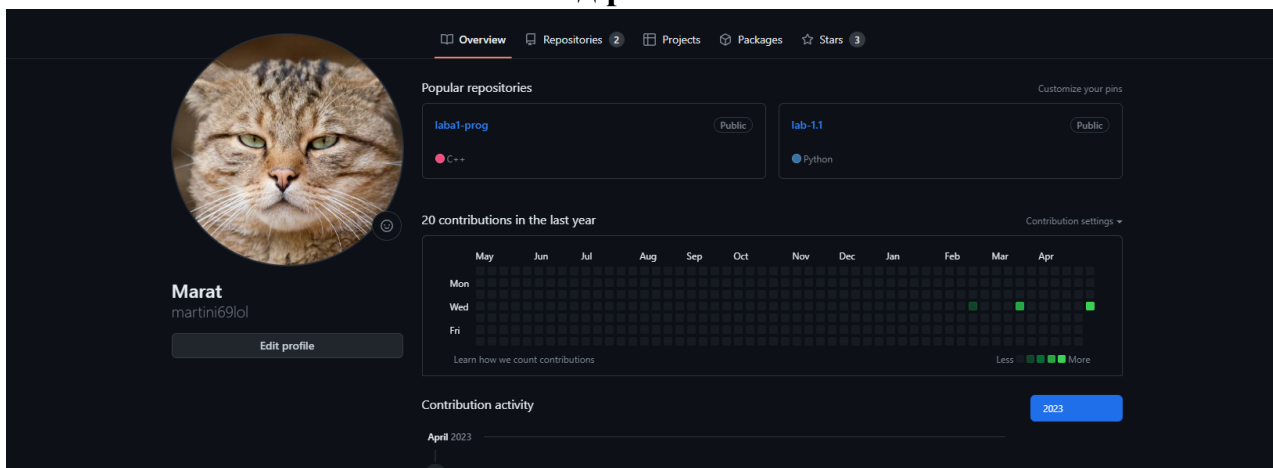


Рисунок 1. Создание аккаунта на GitHub

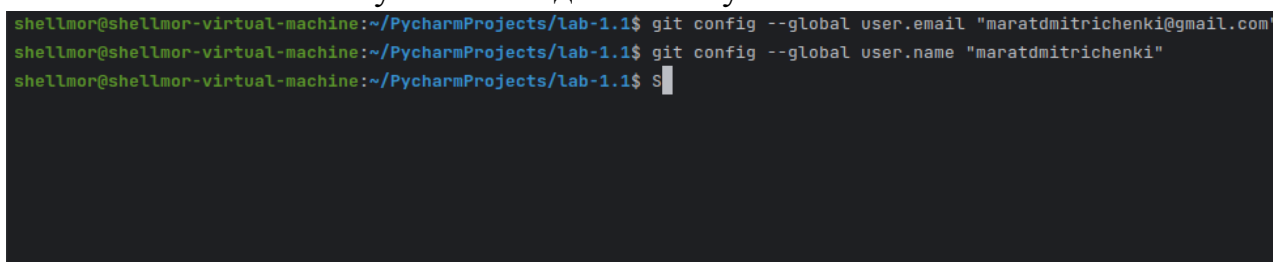


Рисунок 2. Добавление имени и электронной почты

**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 \***

martini69lol / lab-1 ✓

Great repository names are short and memorable. Need inspiration? How about **cuddly-fortnight?**

**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 3. Создание репозитория GitHub

```
PS C:\Users\Shellmor\PycharmProjects\lab-1> git clone https://github.com/martini69lol/lab-1.git
Cloning into 'lab-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.
PS C:\Users\Shellmor\PycharmProjects\lab-1> 
```

Рисунок 4. Клонирование репозитория

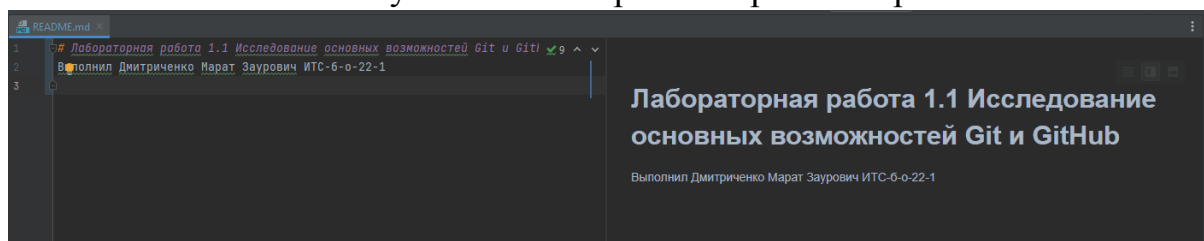


Рисунок 5. Изменение файла README

```
commit baa518bf4e81322c4a8f503055f670c3e3b1e94f (HEAD -> main, origin/main, origin/HEAD)
Author: maratdmitrichenki <maratdmitrichenki@gmail.com>
Date: Wed Apr 26 13:39:26 2023 +0300

    add test

commit 65a401e6ab8e11e1ab0312db9a43ec1d04e47695
Author: maratdmitrichenki <maratdmitrichenki@gmail.com>
Date: Wed Apr 26 13:38:29 2023 +0300

    add return

commit 4f6b385ab9f5a28e86fd94348d683941334c41ef
Author: maratdmitrichenki <maratdmitrichenki@gmail.com>
Date: Wed Apr 26 13:38:11 2023 +0300

    add end fun

commit 0479c84bcbcc7da7229544acb0c451deb083ef15
Author: maratdmitrichenki <maratdmitrichenki@gmail.com>
Date: Wed Apr 26 13:37:34 2023 +0300

    add if array

commit 2f26e622d0065b47e94d4e6cc7b7ba2fc2a7e232
Author: maratdmitrichenki <maratdmitrichenki@gmail.com>
Date: Wed Apr 26 13:36:42 2023 +0300

    add for range

commit 1ccbbade2346130de21436df22ad077b9cd9317d
Author: maratdmitrichenki <maratdmitrichenki@gmail.com>
Date: Wed Apr 26 13:36:11 2023 +0300

    add for

commit e3750a139a0ce4ce7b001f479cf768db852a1aad
Author: maratdmitrichenki <maratdmitrichenki@gmail.com>
Date: Wed Apr 26 13:35:48 2023 +0300

    add length

commit faa9db7af7a4e61a1c07de4f01667204105e099e
Author: maratdmitrichenki <maratdmitrichenki@gmail.com>
Date: Wed Apr 26 13:35:11 2023 +0300

    add def b_sort

commit 1a9c72bd29382e49e2e5d705b0610875109e4b3e
Author: maratdmitrichenki <maratdmitrichenki@gmail.com>
Date: Wed Apr 26 13:32:43 2023 +0300

    README add

commit 006444d5508546221517791e13337515378a97f6
Author: Marat <125448969+martini69lol@users.noreply.github.com>
Date: Wed Apr 26 13:13:00 2023 +0300

    Initial commit
```

Рисунок 6. Список добавленных коммитов

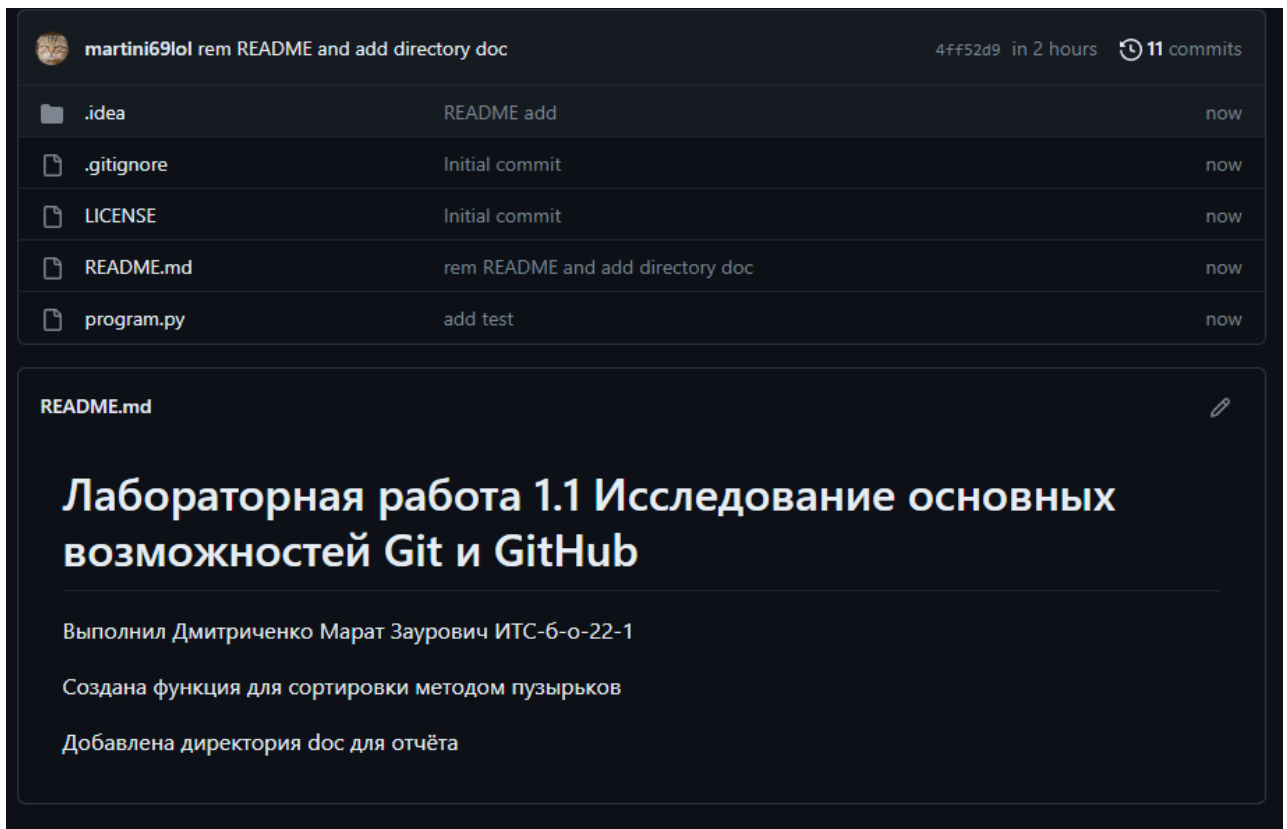


Рисунок 7. Обновленный удаленный репозиторий

Ссылка: <https://github.com/martini69lol/lab-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 распределенным 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 всех данных вычисляется хеш-сумма, и только потом происходит сохранение. В дальнейшем обращение к сохраненным 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 и брать что либо себе или участвовать в вашем проекте.

8. *Какие бывают репозитории в GitHub?*

Репозиторий — это хранилище всех версий кода. Он бывает трех видов:

1. Локальный — расположен на одном компьютере, и работать с ним может только один человек.
2. Централизованный — расположен на сервере, куда имеют доступ сразу несколько программистов.
3. Распределенный — самый удобный вариант с облачным хранилищем. Главный репозиторий хранится в облаке, а его локальные копии у разработчиков на компьютерах. Когда программист вносит правки в локальную версию, ее можно синхронизировать с удаленной. Получается, что в облаке всегда актуальный код.

9. *Укажите основные этапы модели работы с GitHub.*

- 1) Регистрация;
- 2) Подключение вашего профиля в используемой IDE
- 3) Создание проекта
- 4) Публикация начального проекта на GitHub
- 5) Создание дополнительных веток (тестовый стенд и прод)
- 6) Создание коммитов в ветке, которая была унаследована от основной
- 7) Мердж дополнительной ветки с основной

10. *Как осуществляется первоначальная настройка Git после установки?*

Необходимо убедиться, какой версии установленный Git используя команду: `git version`. Далее необходимо клонировать репозиторий командой: `git clone ссылка`. Далее в папке с локальным репозиторием копируем ссылку расположения папки и в git используем команду: `cd /d <Расположения папки на компьютере>`. Свяжем локальный репозиторий и удалённый командами: `git config --global user.name .`

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 компьютер. Для этого на странице репозитория необходимо найти кнопку Clone или Code и щелкнуть по ней, чтобы отобразить адрес репозитория для клонирования. Затем открыть командную строку или терминал и перейти в каталог, куда вы хотите скопировать хранилище. Затем написать `git clone` и ввести адрес. Клонировать репозиторий нужно чтобы он был на компьютере со всеми созданными и измененными файлами.

Дальше для синхронизации с проектом используется `git pull`

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` ?*

Все выполненные действия будут сохранены в локальном 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.*

Если проект уже был загружен на локальный ПК то для синхронизации всех изменений используется команда

1. `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: я исследовал базовые возможности системы контроля версий Git и веб-сервиса для хостинга IT-проектов GitHub.

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