

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 «Информационные технологии и программирование»

Выполнила:
Мурашко Анастасия Юрьевна
1 курс, группа ИТС-б-0-22-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети», очная форма обучения

(подпись)

Руководитель практики:
Братченко Н.Ю., канд. физ.-мат. наук,
доцент, доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: Исследование основных возможностей Giti GitHub.

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

Конспект теоретического материала:

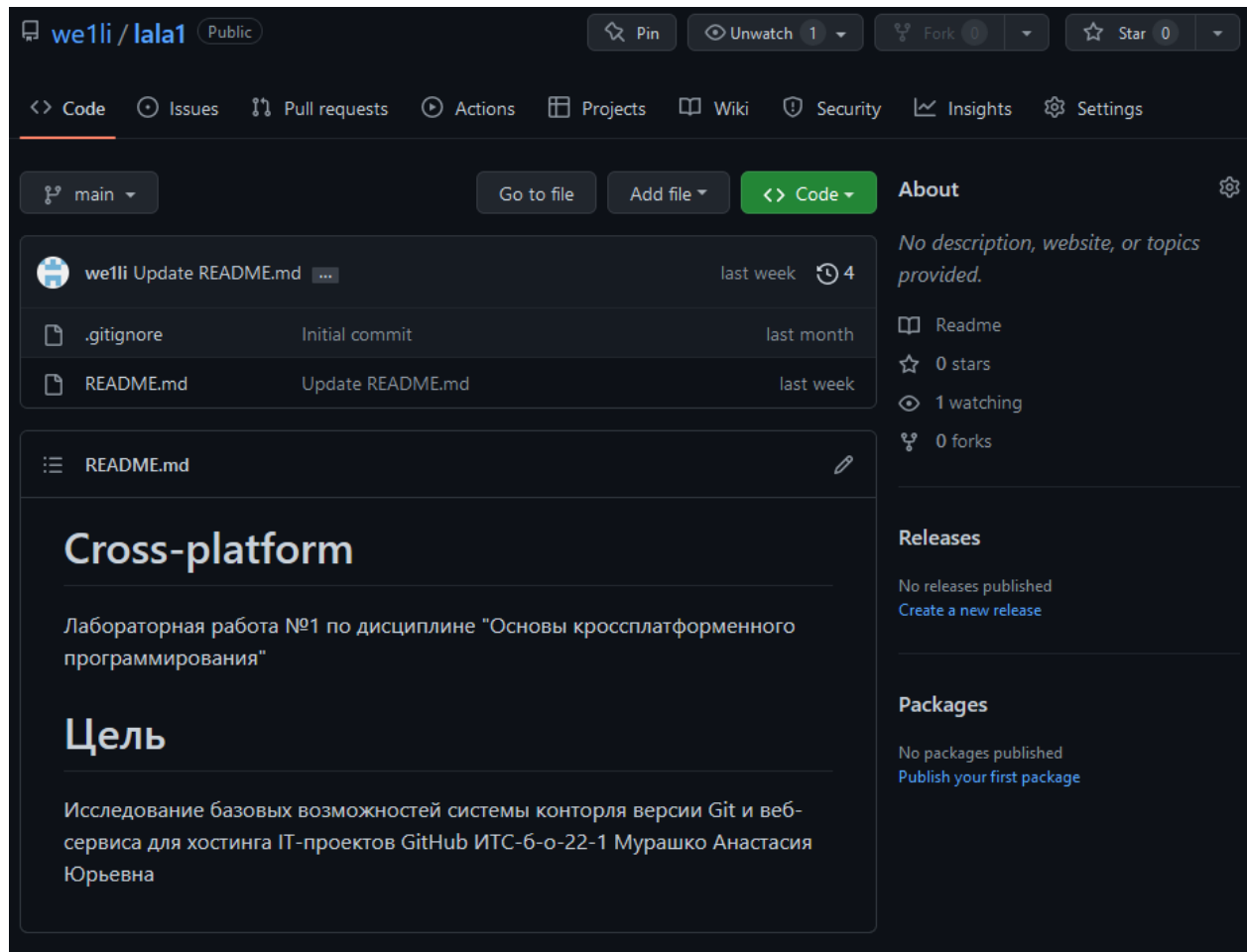
Система контроля версий (СКВ) – это система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов. Программисты обычно помещают в систему контроля версий исходные коды программ, но на самом деле под версионный контроль можно поместить файлы практически любого типа.

Локальные системы контроля версий – Многие люди в качестве метода контроля версий применяют копирование файлов в отдельную директорию (возможно даже, директорию с отметкой по времени, если они достаточно сообразительны). Данный подход очень распространён из-за его простоты, однако он невероятно сильно подвержен появлению ошибок. Можно легко забыть, в какой директории вы находитесь, и случайно изменить не тот файл или скопировать не те файлы, которые вы хотели.

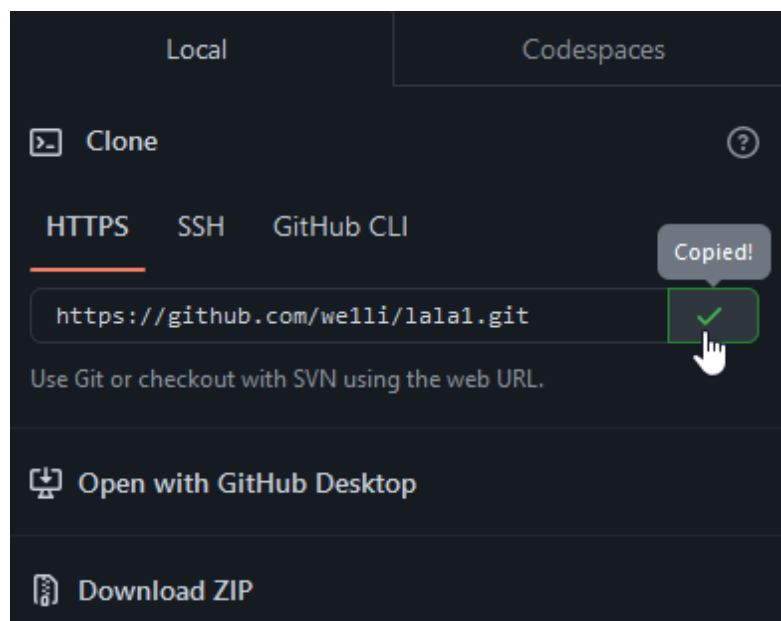
Для того, чтобы решить эту проблему, программисты давным-давно разработали локальные СКВ с простой базой данных, которая хранит записи о всех изменениях в файлах, осуществляя тем самым контроль ревизий.

Порядок выполнения работы:

1. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и выбранный Вами язык программирования.

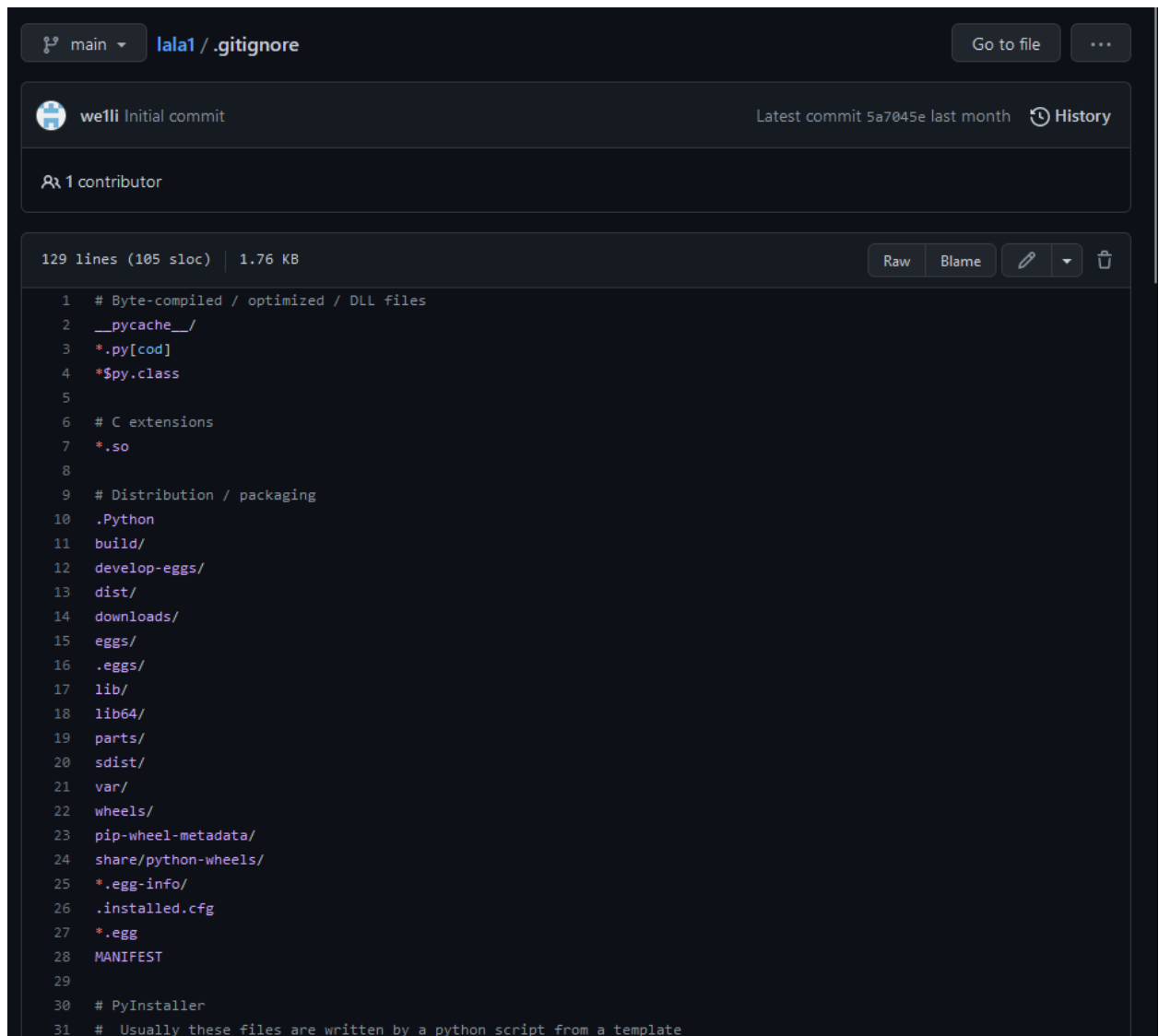


2. Выполните клонирование созданного репозитория на рабочий компьютер.



```
user@DESKTOP-KII2ANN MINGW64 ~  
$ git clone https://github.com/we1li/lala1.git  
Cloning into 'lala1'...  
remote: Enumerating objects: 17, done.  
remote: Counting objects: 100% (17/17), done.  
remote: Compressing objects: 100% (16/16), done.  
remote: Total 17 (delta 2), reused 0 (delta 0), pack-reused 0  
Receiving objects: 100% (17/17), 5.08 KiB | 866.00 KiB/s, done.  
Resolving deltas: 100% (2/2), done.
```

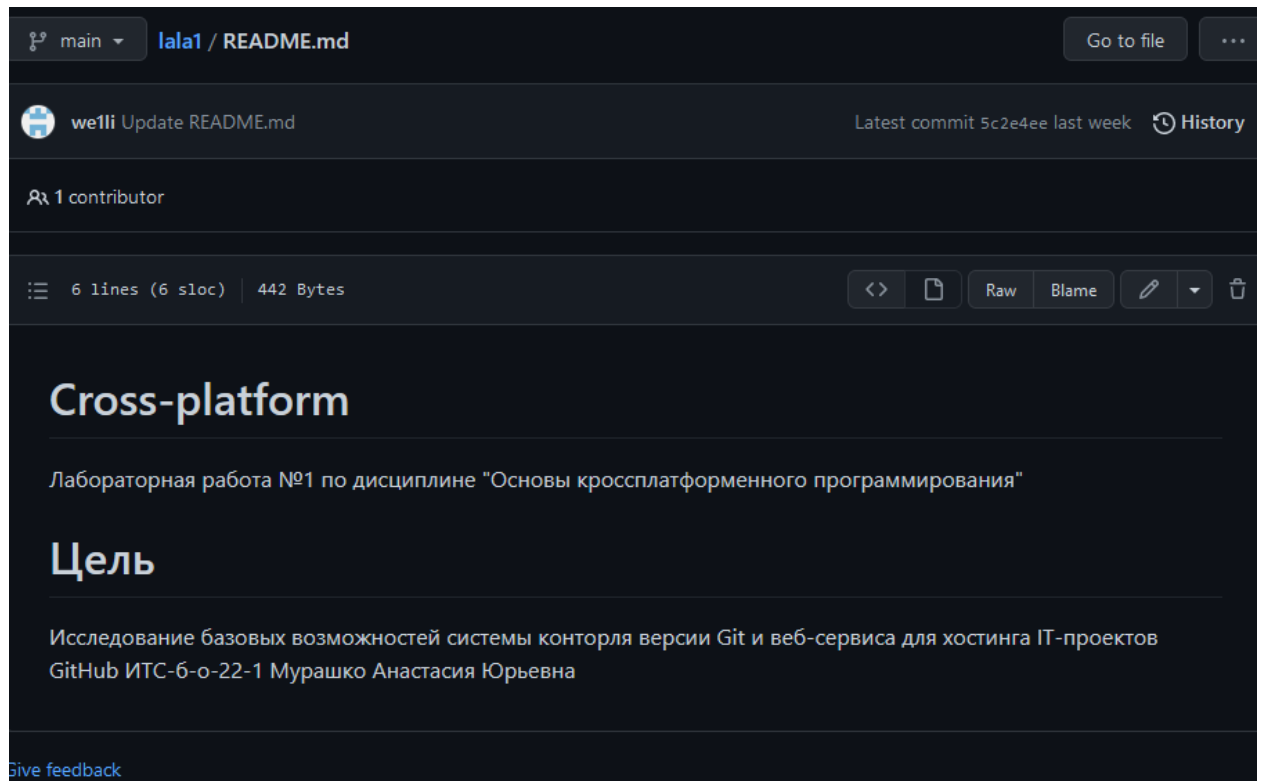
3. Дополните файл `.gitignore` необходимыми правилами для выбранного языка программирования и интегрированной среды разработки.



The screenshot shows the GitHub interface for the repository 'lala1' by user 'we1li'. The file '.gitignore' is selected, showing its content. The file is 129 lines long, 1.76 KB, and was initially committed. The content of the file is as follows:


```
1 # Byte-compiled / optimized / DLL files  
2 __pycache__/  
3 *.py[cod]  
4 *$py.class  
5  
6 # C extensions  
7 *.so  
8  
9 # Distribution / packaging  
10 .Python  
11 build/  
12 develop-eggs/  
13 dist/  
14 downloads/  
15 eggs/  
16 .eggs/  
17 lib/  
18 lib64/  
19 parts/  
20 sdist/  
21 var/  
22 wheels/  
23 pip-wheel-metadata/  
24 share/python-wheels/  
25 *.egg-info/  
26 .installed.cfg  
27 *.egg  
28 MANIFEST  
29  
30 # PyInstaller  
31 # Usually these files are written by a python script from a template
```


4. Добавьте в файл README.md информацию о группе и ФИО студента, выполняющего Лабораторную работу:




The screenshot shows a GitHub repository interface for a file named `README.md` in the `lala1` branch. The file was last updated by `we1li` with commit `5c2e4ee` last week. It has 1 contributor. The file details show it is 6 lines (6 sloc) and 442 Bytes. The README content includes a title **Cross-platform**, a subtitle "Лабораторная работа №1 по дисциплине "Основы кроссплатформенного программирования"", a section **Цель**, and a description: "Исследование базовых возможностей системы контроля версии Git и веб-сервиса для хостинга IT-проектов". The author is listed as "GitHub ИТС-6-о-22-1 Мурашко Анастасия Юрьевна". A "Give feedback" link is at the bottom left.

main `lala1` / README.md [Go to file](#) [...](#)

 `we1li` Update README.md Latest commit `5c2e4ee` last week [History](#)

 1 contributor

 6 lines (6 sloc) | 442 Bytes [<>](#) [Raw](#) [Blame](#) [Edit](#) [Delete](#)

Cross-platform

Лабораторная работа №1 по дисциплине "Основы кроссплатформенного программирования"

Цель

Исследование базовых возможностей системы контроля версии Git и веб-сервиса для хостинга IT-проектов
GitHub ИТС-6-о-22-1 Мурашко Анастасия Юрьевна

[Give feedback](#)

5. Напишите небольшую программу на выбранном Вами языке программирования. Фиксируйте изменения при написании программы в локальном репозитории. Должно быть сделано не менее 7 коммитов.

```
commit b79e0bd0d95e4a1bb3374ad19550e2b8a9ccdc74
Author: Nastya <nastya.murashko1@yandex.ru>
Date: Tue Mar 14 20:34:57 2023 +0300

    3 commit
...skipping...
commit 644b4a58c1d1f70a26cd2df4846eefdc5ba18305 (HEAD -> main, origin/main, origin/HEAD)
Author: Nastya <nastya.murashko1@yandex.ru>
Date: Tue Mar 14 20:38:34 2023 +0300

    7 commit

commit 283df580ecfa0d0d2f64ef394bb88eb368b78a03
Author: Nastya <nastya.murashko1@yandex.ru>
Date: Tue Mar 14 20:37:56 2023 +0300

    6 commit

commit eab1849ff007023a48e03f86cee3f86d066c59b0
Author: Nastya <nastya.murashko1@yandex.ru>
Date: Tue Mar 14 20:37:17 2023 +0300

    5 commit

commit f53c8a04e27a394cb361bf9db27b662f03f328a7
Author: Nastya <nastya.murashko1@yandex.ru>
Date: Tue Mar 14 20:36:20 2023 +0300

    4 commit

commit b79e0bd0d95e4a1bb3374ad19550e2b8a9ccdc74
Author: Nastya <nastya.murashko1@yandex.ru>
Date: Tue Mar 14 20:34:57 2023 +0300

    3 commit

commit 9118c3f0f8d7725bde005569701d5cc4caf708e7
Author: Nastya <nastya.murashko1@yandex.ru>
Date: Tue Mar 14 20:33:12 2023 +0300

    2 commit

commit bf98361428912a100dac1aa33a76a12fd5ed4080
Author: Nastya <nastya.murashko1@yandex.ru>
Date: Tue Mar 14 20:31:24 2023 +0300

Add file program.py
```

1. Что такое СКВ и каково ее назначение?

Система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов.

2. В чем недостатки локальных и централизованных СКВ?

Локальные: возможность потери данных вследствие возникновения физических поломок оборудования; отсутствие возможности совместной

разработки. Централизованные: отсутствие доступа к данным при сбое работы сервера; довольно низкая скорость работы (из-за возникновения сетевых задержек).

3. К какой СКВ относится Git?

В Git каждая рабочая копия кода сама по себе является репозиторием.

4. В чем концептуальное отличие Git от других СКВ?

Бесплатный и open-source. Можно бесплатно скачать и вносить любые изменения в исходный код; Небольшой и быстрый. Выполняет все операции локально, что увеличивает его скорость. Кроме того, Git локально сохраняет весь репозиторий в небольшой файл без потери качества данных; Простое ветвление. В других системах контроля версий создание веток— утомительная и трудоёмкая задача, так как весь код копируется в новую ветку. В Git управление ветками реализовано гораздо проще и эффективнее.

5. Как обеспечивается целостность хранимых данных в Git?

Git обеспечивает целостность хранимых данных, используя контрольные суммы в качестве идентификаторов.

6. В каких состояниях могут находиться файлы в Git? Как связаны эти состояния?

Отслеживаемые файлы могут находиться в 3 состояниях: Не изменено (Unmodified), изменено (Modified), подготовленное (Staged).

7. Что такое профиль пользователя в GitHub?

У каждого пользователя есть публичный профиль, который помогает в поиске работы. Чтобы показать свой опыт потенциальному работодателю, нужно оставить в резюме ссылку на профиль. Когда рекрутер или другой специалист перейдут по ней, то увидят информацию о вас.

8. Какие бывают репозитории в GitHub?

Репозиторий Git бывает локальный и удалённый.

9. Укажите основные этапы модели работы с GitHub.

Установка Git; добавление имени, фамилии и адреса электронной почты; ввод определенных команд для Git; загрузка изменений в состояние (staged); добавление коммита; отправка в репозиторий на сервис GitHub.

10. Как осуществляется первоначальная настройка Git после установки?

Добавление имени, фамилии и адреса электронной почты: `git config --global user.name` – указывает ваше имя, фамилию. `git config --global user.email` – указывает вашу электронную почту. `git init` – создает новый репозиторий Git.

11.Опишите этапы создания репозитория в GitHub.

Ввод имени для репозитория, добавление описания проекта (выборочно), выбор приватности данного репозитория, добавление дополнительных файлов, как README.md и .gitignore.

12.Какие типы лицензий поддерживаются GitHub при создании репозитория?

Academic Free License v3.0; Boost Software License 1.0; Creative Commons license family; Eclipse Public License 1.0; ISC; MIT и многие другие.

13.Как осуществляется клонирование репозитория GitHub?

Зачем нужно клонировать репозиторий? С помощью команд `git clone/git push`.

Чтобы упростить устранение конфликтов слияния, добавление или удаление файлов и отправку больших фиксаций.

14.Как проверить состояние локального репозитория Git? Используйте команду `git status` , чтобы проверить текущее состояние репозитория