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 программной инженерии»

Выполнил:
Звездин Алексей Сергеевич
2 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

_____
(подпись)

Руководитель практики:
Воронкин Р. А., доцент кафедры
инфокоммуникаций

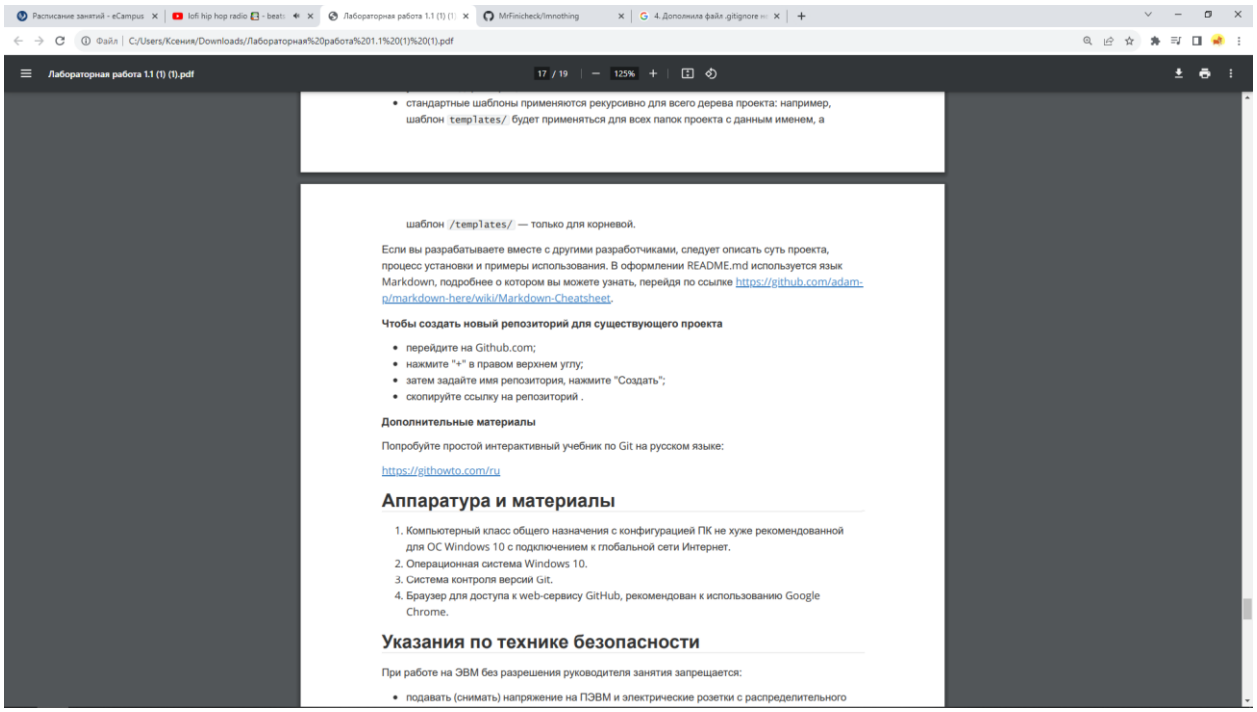
_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2023 г.

Ход работы

1. Я изучил теоретический материал работы



- стандартные шаблоны применяются рекурсивно для всего дерева проекта: например, шаблон `templates/` будет применяться для всех папок проекта с данным именем, а

шаблон `/templates/` — только для корневой.

Если вы разрабатываете вместе с другими разработчиками, следует описать суть проекта, процесс установки и примеры использования. В оформлении README.md используется язык Markdown, подробнее о котором вы можете узнать, перейдя по ссылке https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet.

Чтобы создать новый репозиторий для существующего проекта

- перейдите на Github.com;
- нажмите "+" в правом верхнем углу;
- затем задайте имя репозитория, нажмите "Создать";
- скопируйте ссылку на репозиторий .

Дополнительные материалы

Попробуйте простой интерактивный учебник по Git на русском языке:

https://githowto.com/ru

Аппаратура и материалы

1. Компьютерный класс общего назначения с конфигурацией ПК не хуже рекомендованной для ОС Windows 10 с подключением к глобальной сети Интернет.
2. Операционная система Windows 10.
3. Система контроля версий Git.
4. Браузер для доступа к web-сервису GitHub, рекомендован к использованию Google Chrome.

Указания по технике безопасности

При работе на ЭВМ без разрешения руководителя занятия запрещается:

- подавать (снимать) напряжение на ПЭВМ и электрические розетки с распределительного

Рисунок 1.1 – Изучение материала для лабораторной работы

2. Создал общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и выбранный мною язык программирования

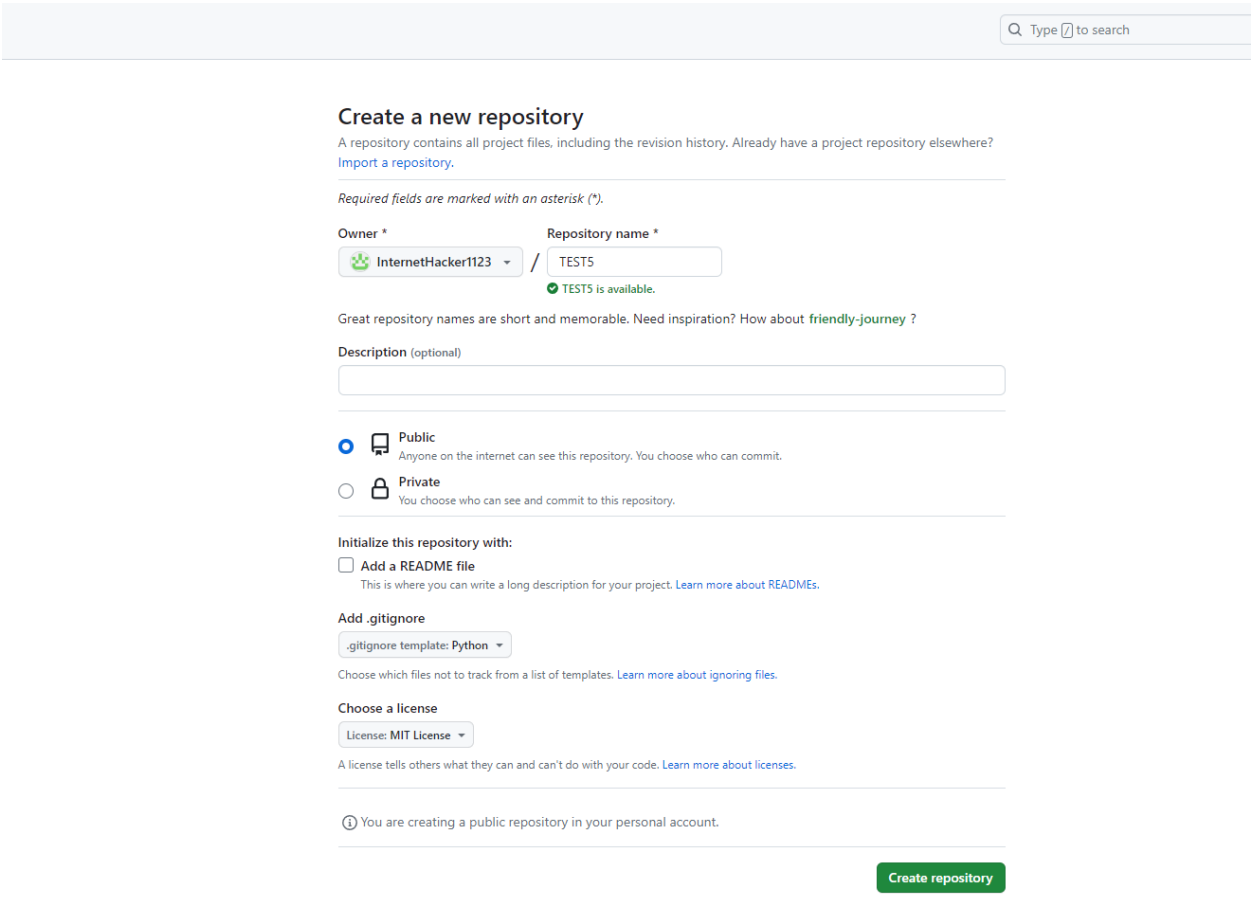


Рисунок 2.1 – Настройка репозитория

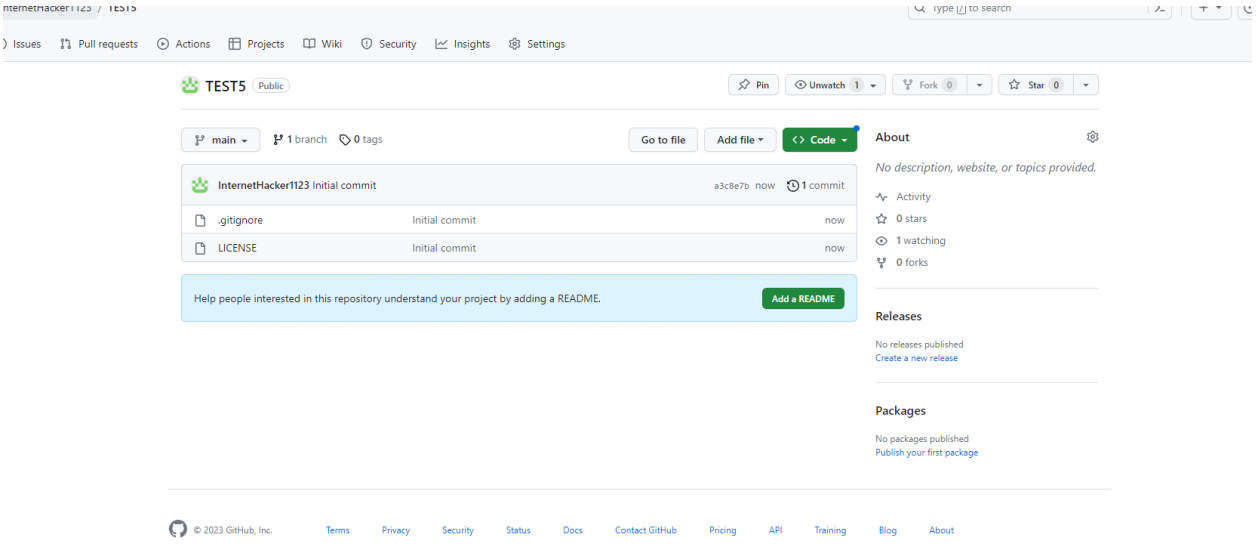


Рисунок 2.2 – Готовый репозиторий

3. Выполнил клонирование созданного репозитория на рабочий компьютер

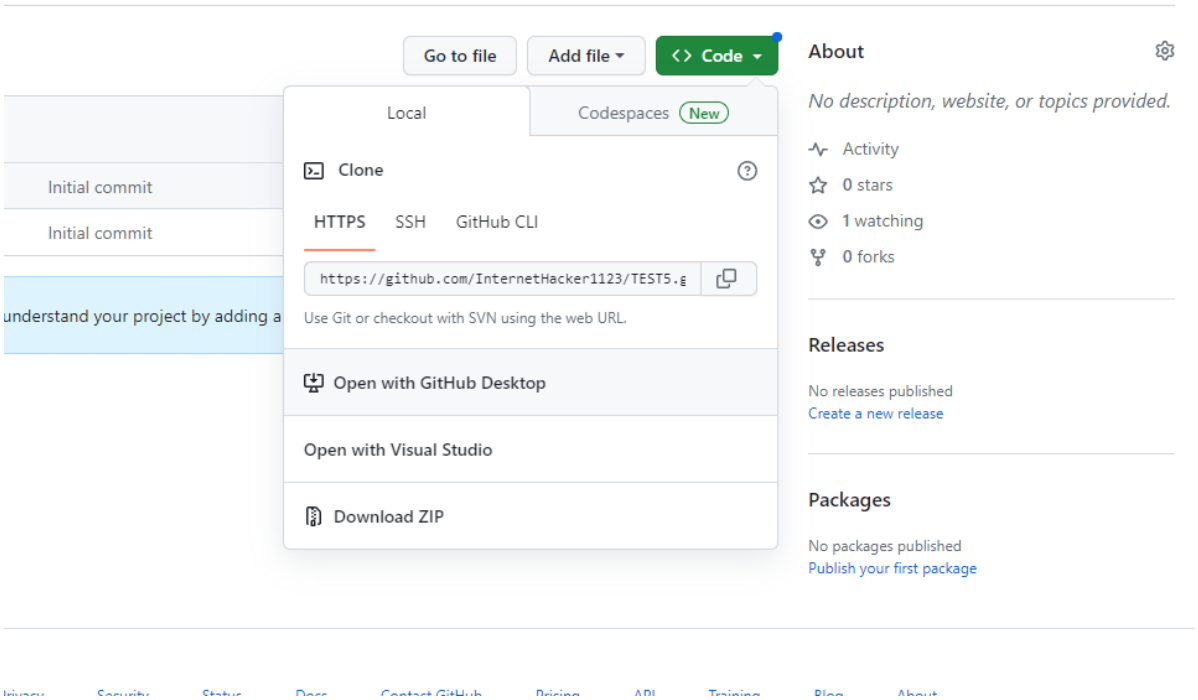


Рисунок 3.1 – Копирование ссылки репозитория

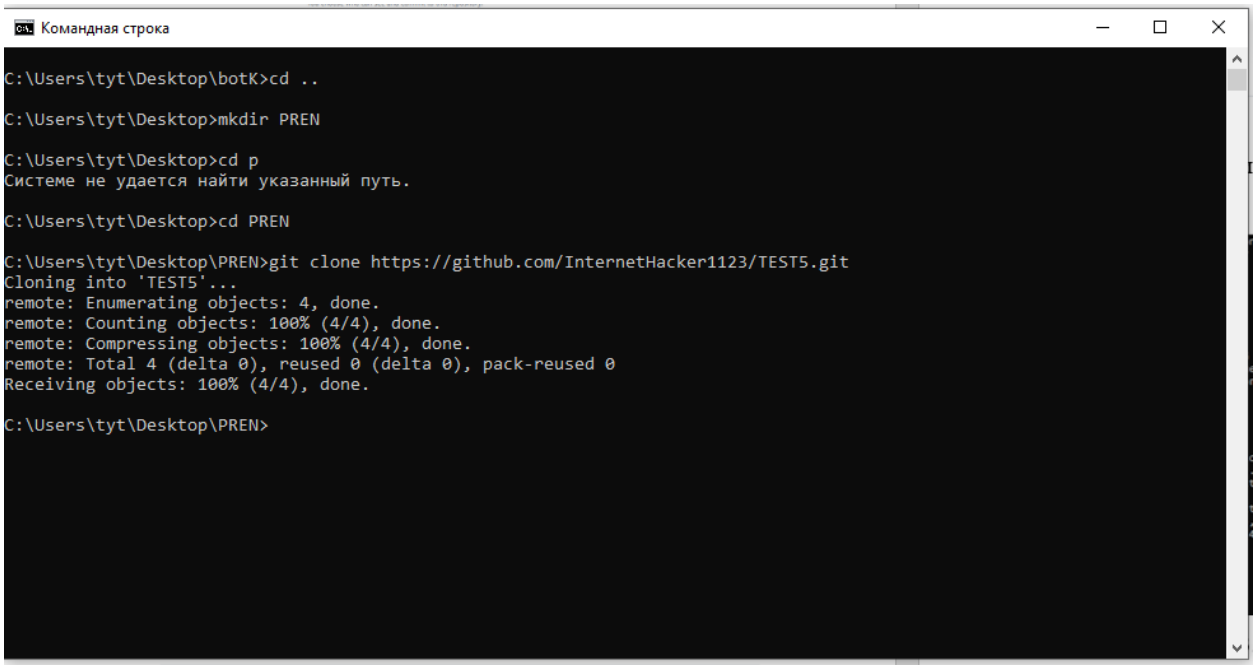


Рисунок 3.2 – Копирование репозитория на рабочий компьютер

4. Дополнил файл .gitignore необходимыми правилами для выбранного языка программирования и интегрированной среды разработки

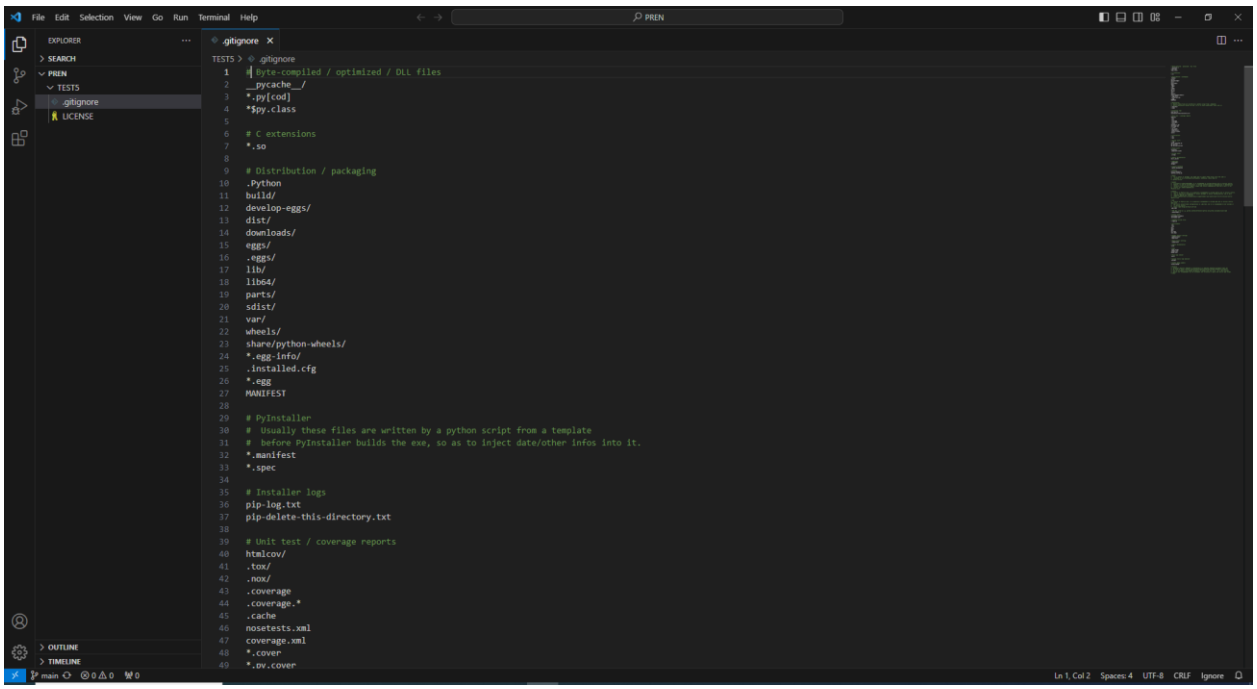


Рисунок 4.1 – Файл с необходимыми правилами для языка Python

5. Добавил в файл README.md информацию о своей группе и моём ФИО

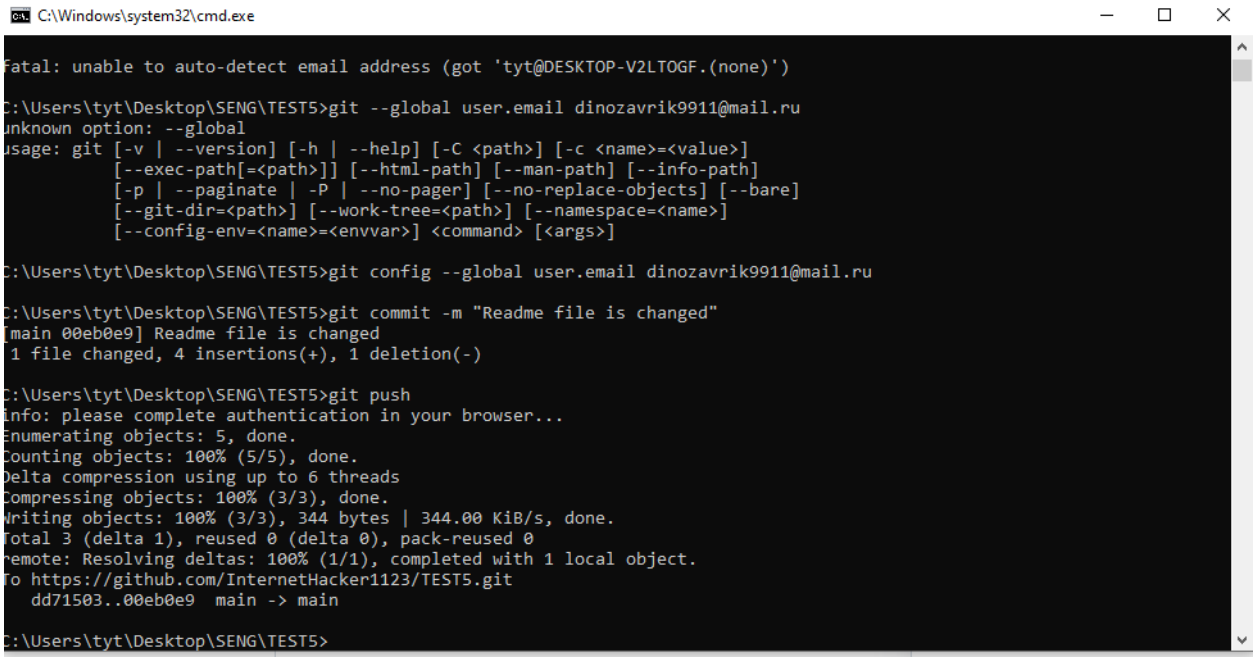


Рисунок 5.1 – Добавляю в файл READ.md информацию о своей группе и моём ФИО на GitHub через командную строку

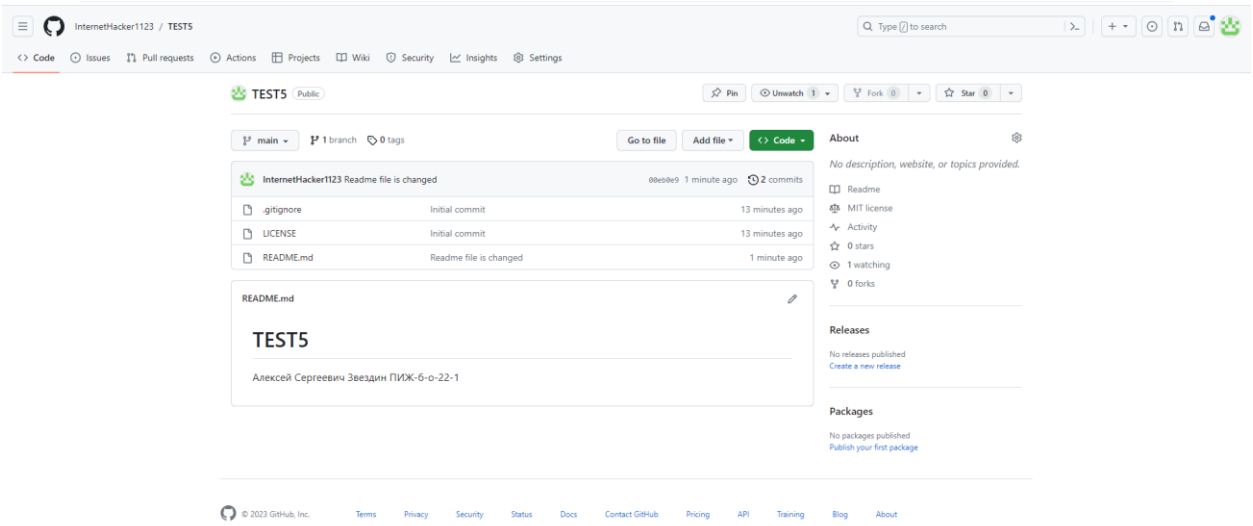


Рисунок 5.2 – Мой файл README.md на GitHub

6. Написал программу на выбранном мною языке программирования. Зафиксировала изменения при написании программы в локальном репозитории. Сделал не менее 7 коммитов

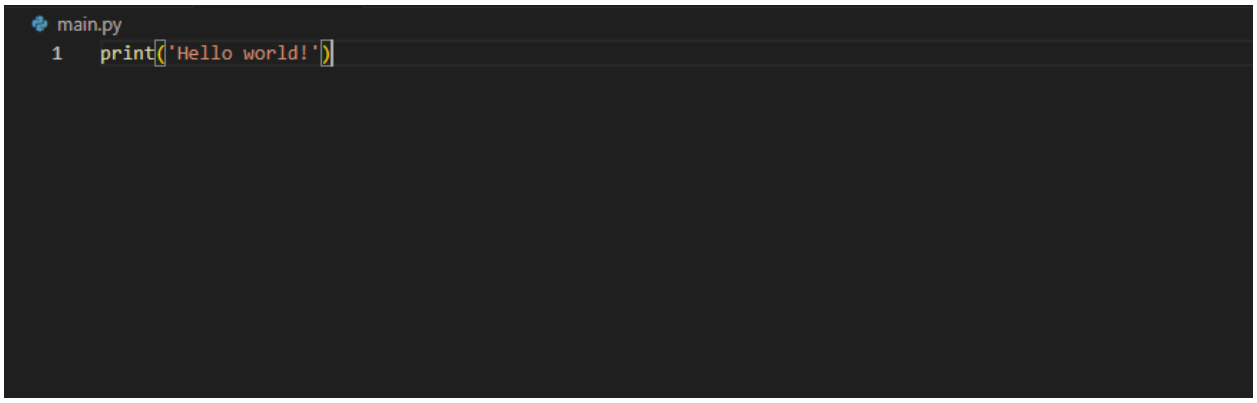


Рисунок 6.1 – Моя программа на Python

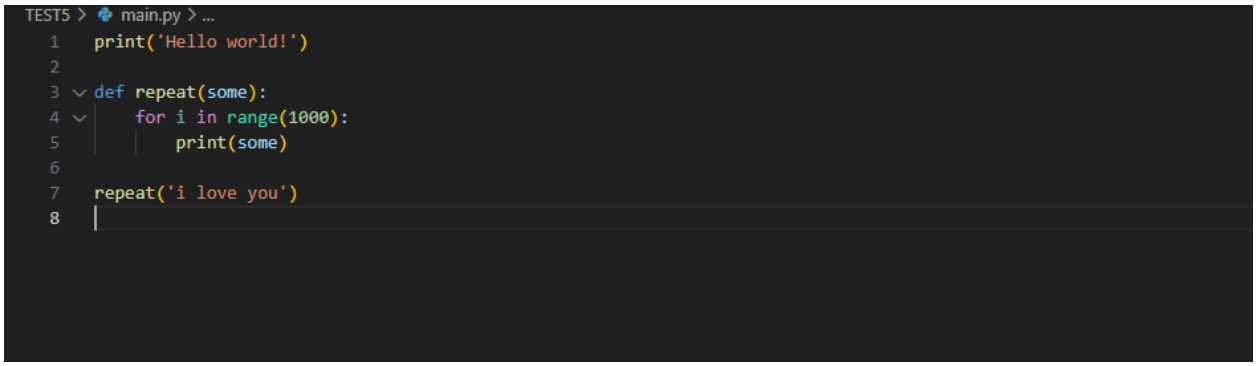


Рисунок 6.2 – Изменяю программу

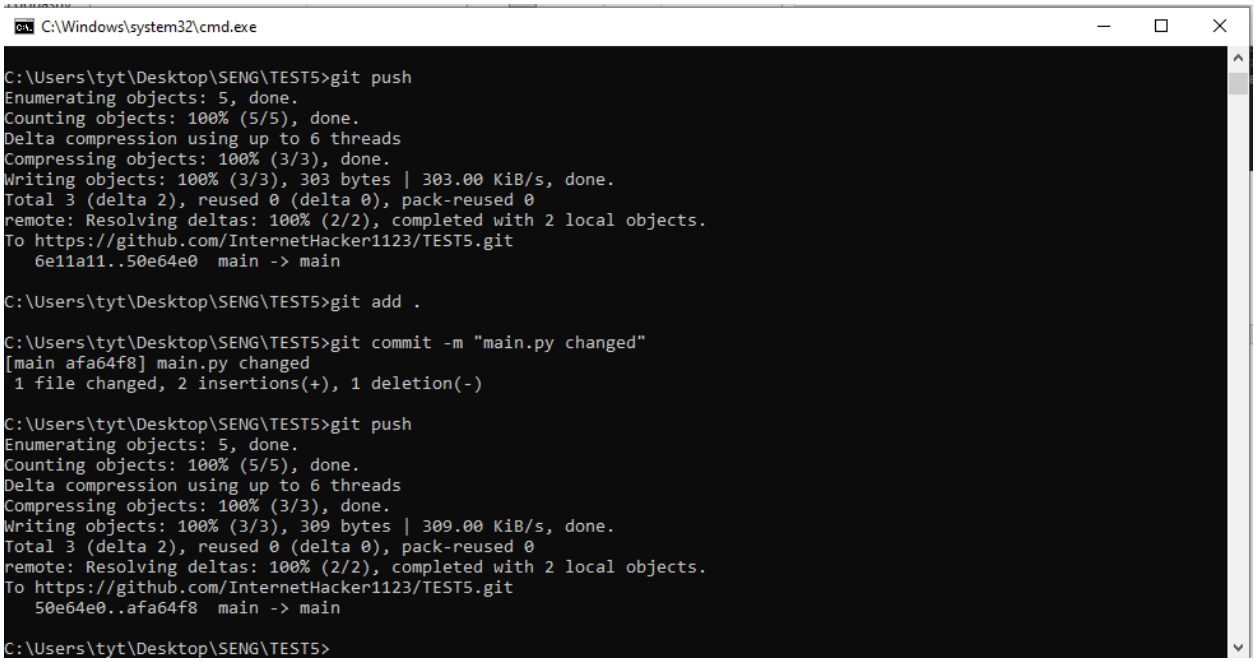


Рисунок 6.3 – Делаю 7 коммитов

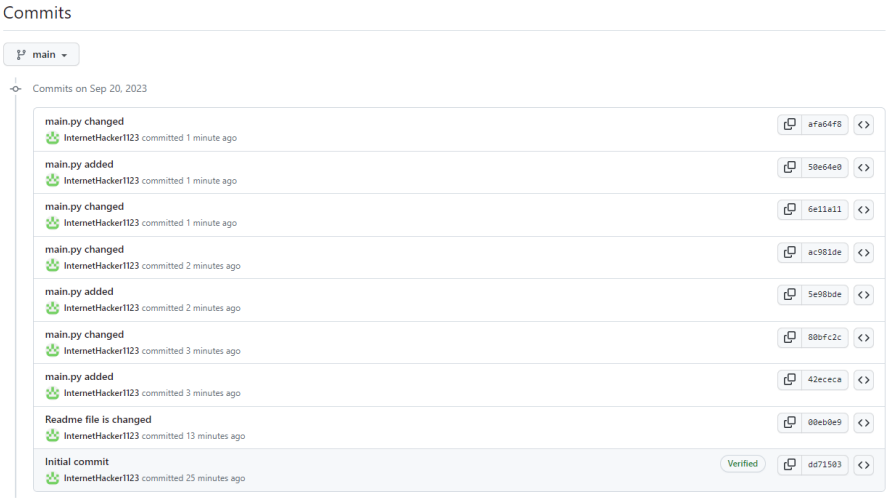


Рисунок 6.4 – Изменения, отображённые на GitHub

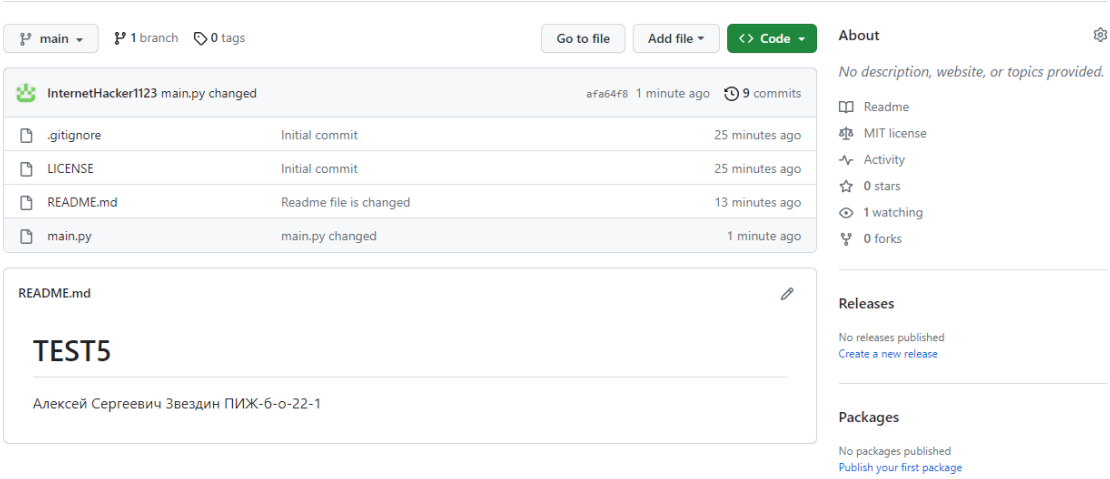


Рисунок 6.5 – Мой репозиторий

7. Добавил файл README и зафиксировала сделанные изменения.

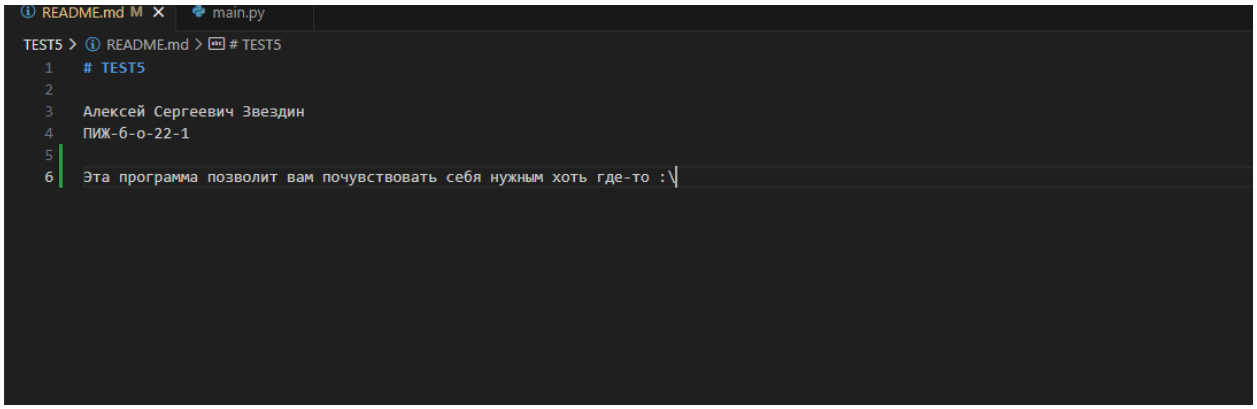


Рисунок 7.1 – Добавляю файл README с описанием программы

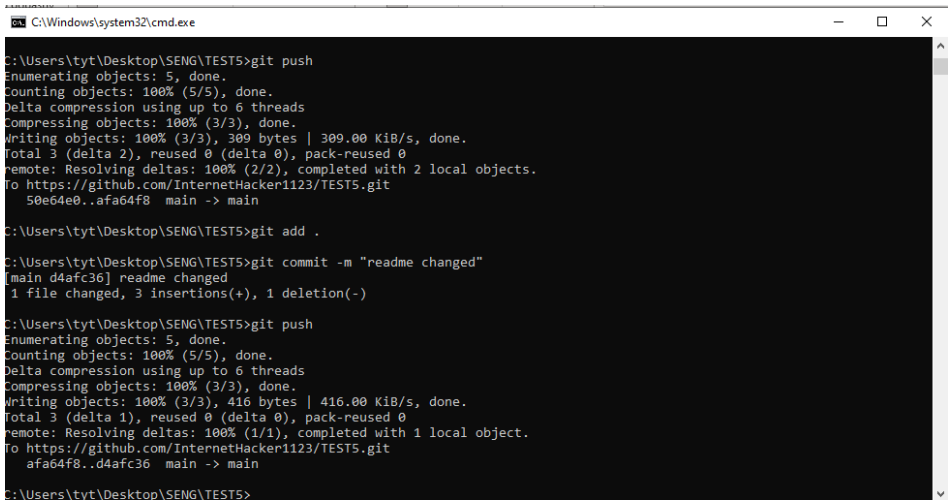


Рисунок 7.2 – Фиксирую сделанные изменения

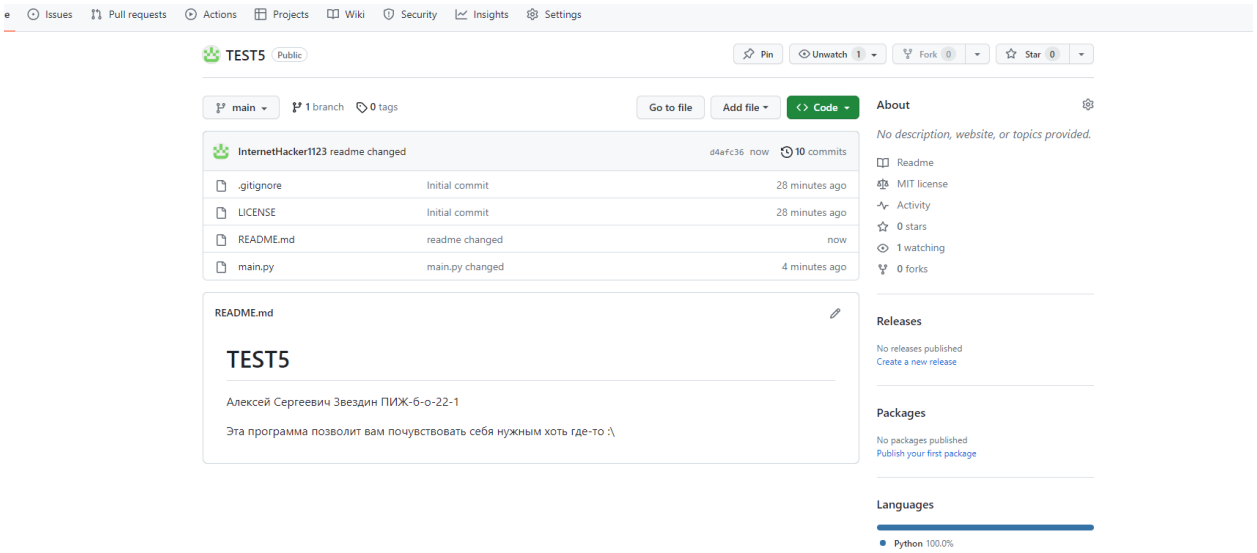


Рисунок 7.3 – Мой репозиторий с измененным файлом README

Контрольные вопросы

1. Что такое СКВ и каково ее назначение?

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.

2. В чем недостатки локальных и централизованных СКВ?

В локальных СКВ можно легко забыть, в какой директории вы находитесь, и случайно изменить не тот файл или скопировать не те файлы, которые вы хотели. В централизированных СКВ 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. Если жёсткий диск, на котором хранится центральная БД, повреждён, а своевременные бэкапы отсутствуют, вы потеряете всё.

3. К какой СКВ относится Git?

   Распределённая система контроля версий

4. В чем концептуальное отличие Git от других СКВ?

Основное отличие Git от любой другой СКВ — это подход к работе со своими данными.

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. Это значит, что невозможно изменить содержимое файла или директории так, чтобы Git не узнал об этом.

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

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. Что такое профиль пользователя в GitHub?

На странице профиля отображаются сведения о вашей работе через репозитории, которые вас интересуют, вклад, который вы сделали, и беседы, в которых вы участвовали.

8. Какие бывают репозитории в GitHub?

   Публичные и приватные

9. Укажите основные этапы модели работы с GitHub.

Создание аккаунта, создание репозитория, клонирование репозитория на локальный диск, отправка изменений на GitHub с помощью git push.

10. Как осуществляется первоначальная настройка Git после установки?

git config --global <user.name>

git config --global <user.email>

11. Опишите этапы создания репозитория в GitHub.

На GitHub в правом верхнем углу, рядом с аватаром есть кнопка с плюсиком, нажимая которую мы переходим к созданию нового репозитория. В результате будет выполнен переход на страницу создания репозитория. Наиболее важными на ней являются следующие поля: Имя репозитория, описание Public/private, gitignore и LICENSE. После заполнения этих полей нажимаем кнопку Create repository.

12. Какие типы лицензий поддерживаются GitHub при создании репозитория?

Список лицензий можно просмотреть при создании нового репозитория

13. Как осуществляется клонирование репозитория GitHub? Зачем нужно клонировать репозиторий?

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. Это нужно для того, чтобы сохранить наши данные на локальный диск.

14. Как проверить состояние локального репозитория Git?

В командной строке с помощью команды 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?

1. В репозитории появляется изменённый, но не подготовленный файл.
2. Появляются изменённые, но подготовленные файлы.
3. После того как выполнен коммит происходит фиксация изменений всех подготовленных файлов, в результате репозиторий не содержит подготовленных файлов. После этого с помощью команды git push фиксированные файлы отправляются на GitHub.

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. Примечание: описание необходимо начать с команды git clone.

На обоих компьютерах выполняется команда git clone. Оба пользователя вносят в программу необходимые изменения, фиксируют их с помощью команды коммит. Перед командой push для синхронизации обоих компьютеров с GitHub выполняют команду git pull и только потом git push.

17. GitHub является не единственным сервисом, работающим с Git. Какие сервисы еще Вам известны? Приведите сравнительный анализ одного из таких сервисов с GitHub.

Преимущества и недостатки GitLab

Преимущества:

– Бесплатный план без ограничений, но есть планы оплаты.
– Это лицензия с открытым исходным кодом.
– Разрешает самостоятельный хостинг на любом плане.
– Он очень хорошо интегрирован с Git.

Недостатки:

– Его интерфейс может быть несколько медленнее по сравнению с конкурентами.

Преимущества и недостатки GitHub

Преимущества:

- Бесплатное обслуживание, хотя есть и платные.
- Очень быстрый поиск в структуре репозиториев.
- Большое сообщество и легко найти помощь.
- Он предлагает практические инструменты для сотрудничества и хорошую интеграцию с Git.
- Легко интегрируется с другими сторонними сервисами.
- Он также работает с TFS, HG и SVN.

Недостатки:

- У него есть ограничения по пространству, так как вы не можете превышать 100 МБ в одном файле, в то время как репозитории ограничены 1 ГБ в бесплатной версии.

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.

На графическом интерфейсе программы GitHub Desktop выбираем необходимый репозиторий. Указываем название коммита и его описание, затем нажимаем кнопку коммит. Происходит фиксация изменений на локальной машине. После этого нажимаем кнопку push и происходит передача данных на GitHub.