МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГТУ», ВГТУ)

Факультет информационных технологий и компьютерной безопасности

Кафедра компьютерных интеллектуальных технологий проектирования

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

По дисциплине: «Системы хранения и обработки данных»

Тема: «Основы работы с системой управления версиями Git»

Выполнил работу студент группы змиИВТ-241: Москвитина А.Г.

подпись, дата

Принял: Короленко В.В.

подпись, дата

Воронеж 2024

Цель работы: изучить назначение распределённой системы управления версиями, освоить процесс создания репозитория и основы управления версиями файлов.

Основные задачи:

⦁ создание профиля и репозитория на платформе github;

⦁ установка git, запуск;

⦁ создание ветки для отработки своего проекта;

⦁ изучение основных команд git;

⦁ клонирование удаленного репозитория на локальный компьютер.

⦁ создание файла с данными о себе, изменение файла и отправка соответствующих изменений из локального репозитория на удаленный;

⦁ откат изменений до предыдущего коммита.

Ход работы:

Заходим в свой профиль на платформе GitHub (рис. 1).

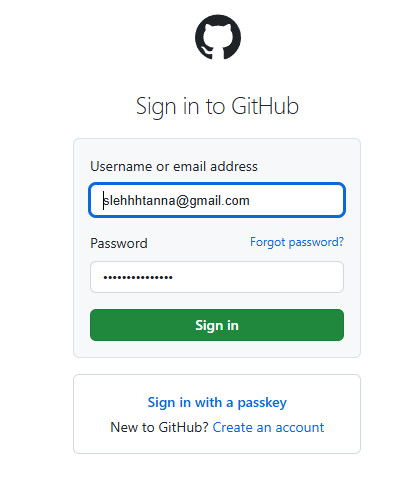


Рисунок 1 – Профиль пользователя

Создаем новый репозиторий (рис. 2).

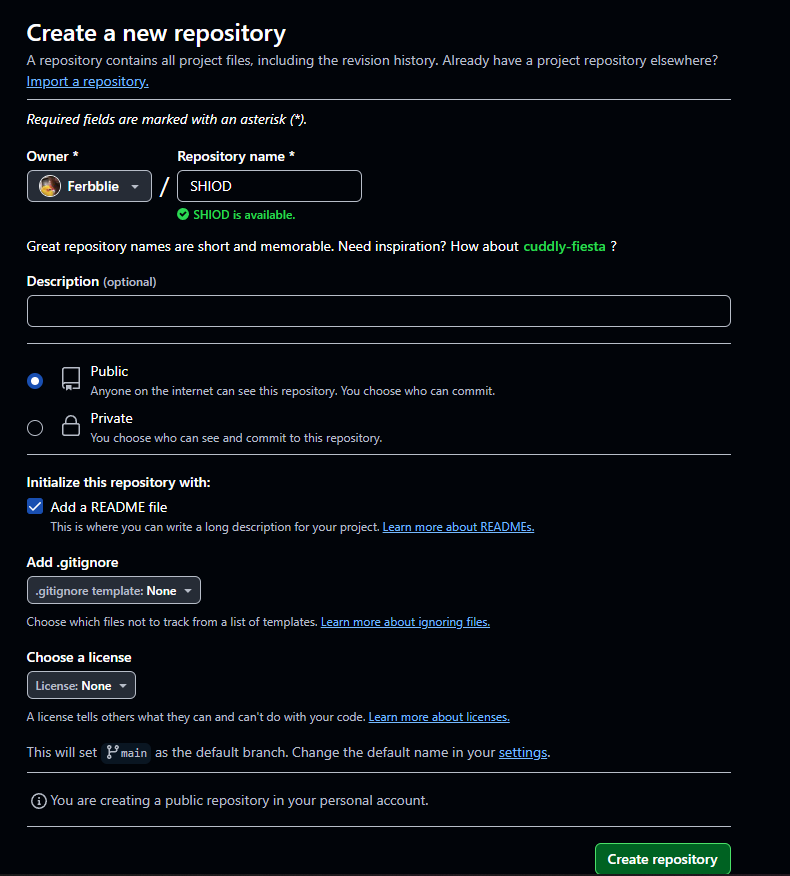


Рисунок 2 – Создание репозитория

Указываем необходимость создания файла Readme.md (рис. 3).

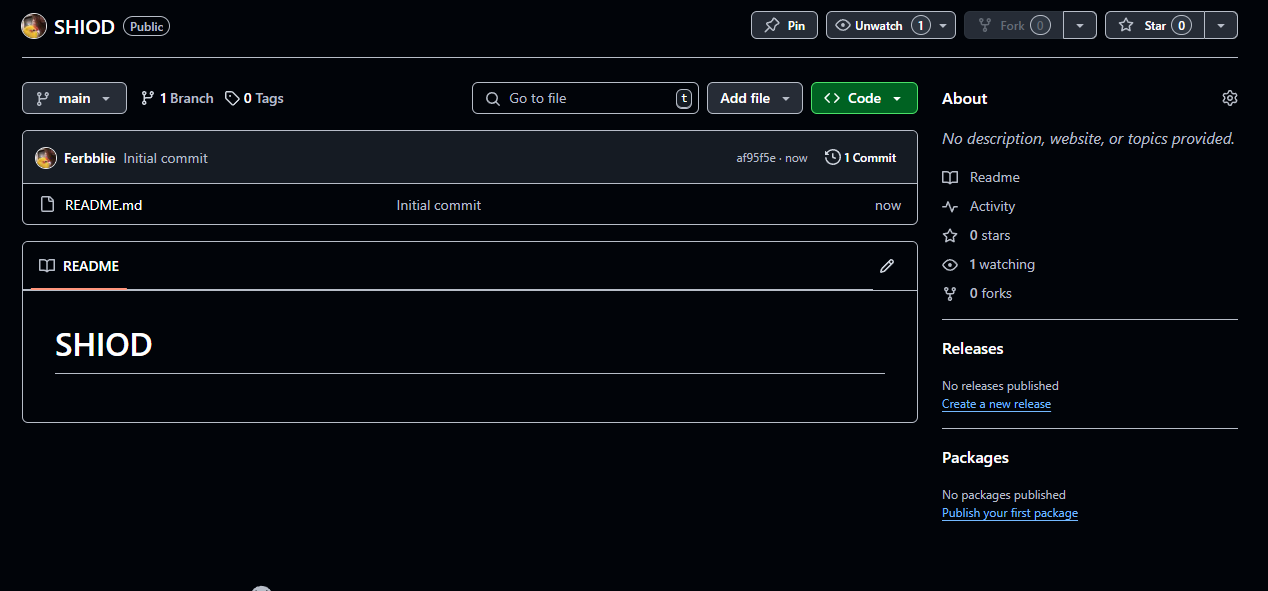


Рисунок 3 – Созданный репозиторий

Добавляем общую информацию в наш файл (рис. 4).

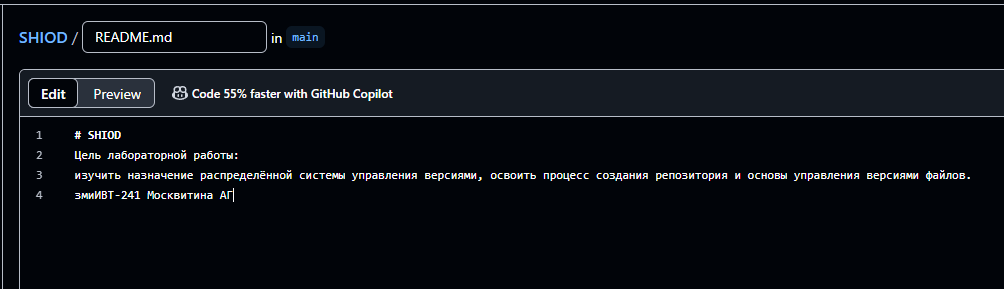


Рисунок 4 – Редактирование файла Readme.md

Общий вид нашего репозитория представлен на рисунке 5.

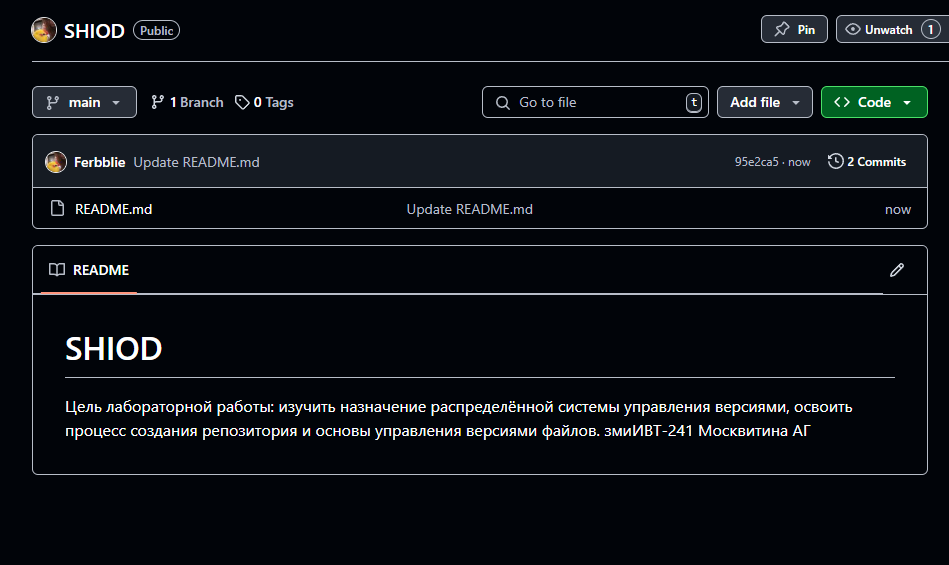


Рисунок 5 – Общий вид

Для отработки и внесения изменений в проект, создаем новую ветку «LR4-MAG». В этой ветке мы планируем проводить основную работу, внесение изменений и тестирование, сохраняя "main" в стабильном состоянии (рис. 6-7).

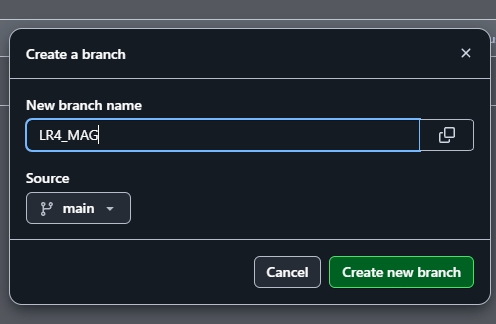


Рисунок 6 – Создание ветки

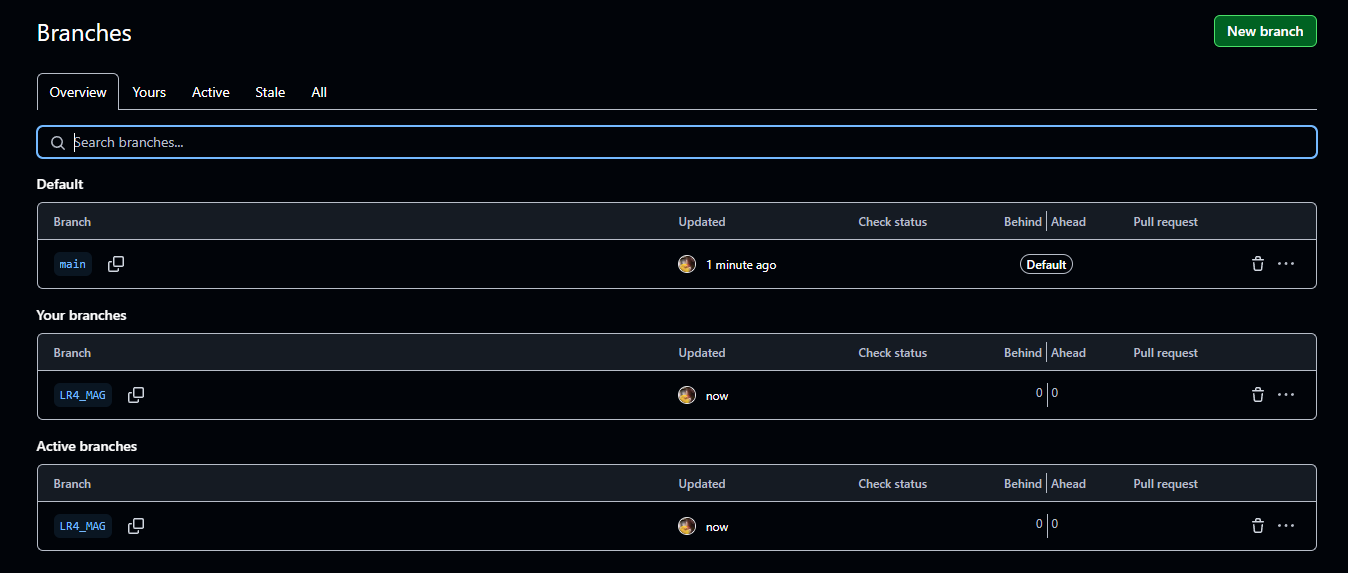


Рисунок 7 – Новая ветка проекта

Устанавливаем MINGW для дальнейшего взаимодействия (рис. 8).

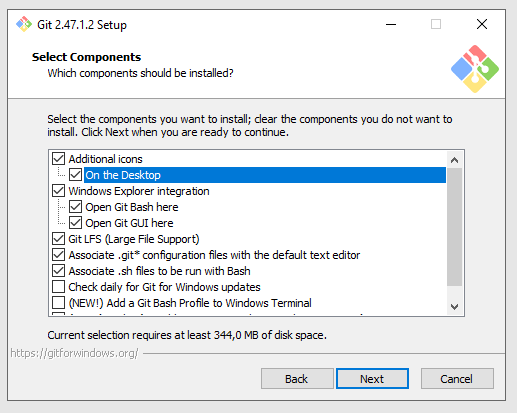


Рисунок 8 – Установка MINGW

Далее был клонирован удаленный репозиторий на локальный компьютер (рис. 9).

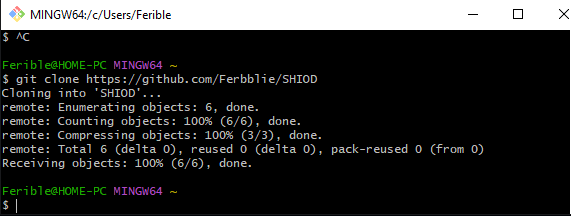


Рисунок 9 – Клонирование

Результатом клонирования становится папка (рис. 10).

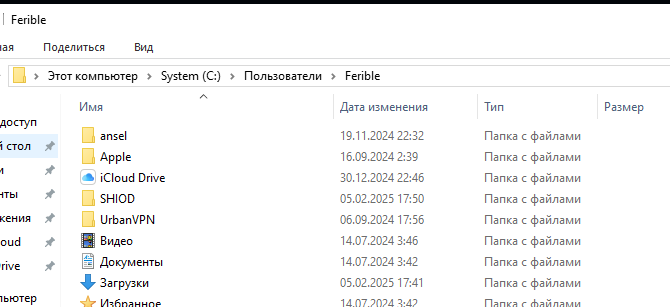


Рисунок 10 – Результат клонирования

Cоздадим файл “MAG.txt” в локальный репозиторий с помощью команды: git add. После все изменения были отправлены на удаленный репозиторий с помощью команд: git commit –m “Add txt file” и git push. Результат работы представлен на рисунке 11.

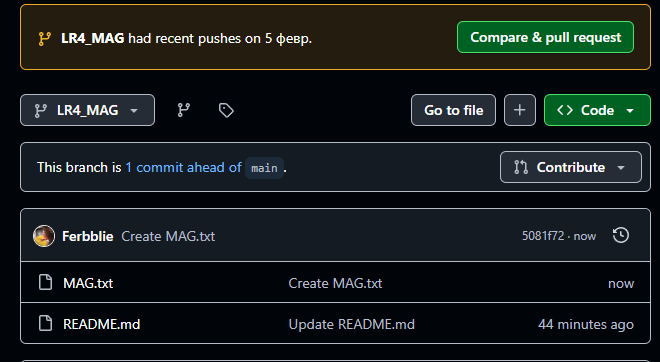


Рисунок 11 – Добавление файла

Далее внесем в текстовые файлы информацию о среднем балле студента, а после отправим на удаленный репозиторий с помощью git commit –m “Добавлена информация о среднем балле” и git push. Результат операции показан на рисунке 12.

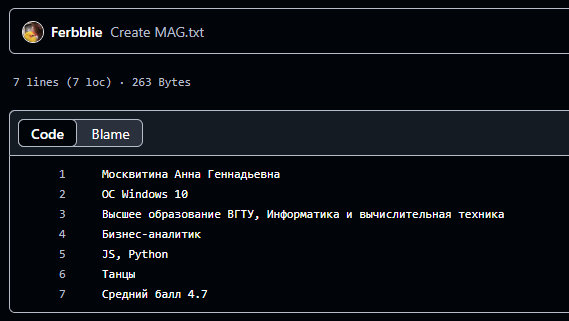


Рисунок 12 – Изменения в файле MAG.txt

Далее добавим информацию о месте рождения и отправим на удаленные репозитории git commit –m “Добавлена информация о месте рождения”. Результат на рисунке 13.

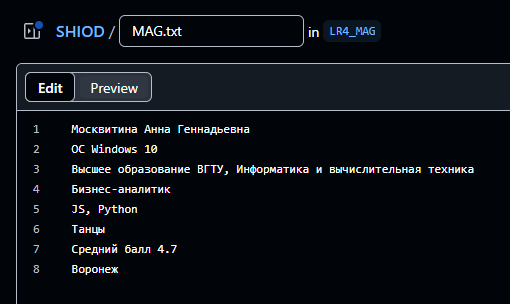


Рисунок 14 – Добавление информации о месте рождения

После выполним откат изменений до предыдущего коммита при помощи команды "git reset --hard <Хеш>". Здесь "Хеш" представляет собой уникальный идентификатор (SHA-1 хеш) коммита, к которому мы хотим вернуться. Этот процесс позволит нам отменить все изменения после указанного коммита и вернуть репозиторий к состоянию, соответствующему выбранному коммиту.

На рисунке 15 мы можем видеть, что команда успешно выполнилась.

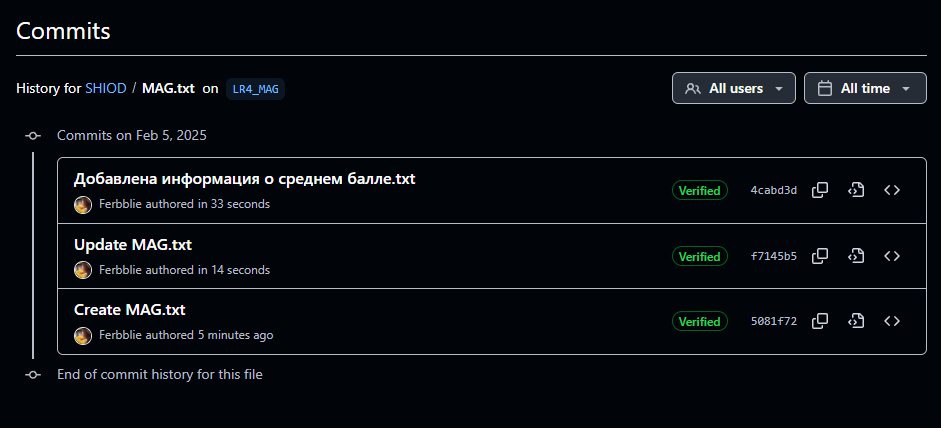


Рисунок 15 – Откат изменений

Вывод: в данной лабораторной работе было изучено назначение распределённой системы управления версиями, освоен процесс создания репозитория и основы управления версиями файлов.

**Контрольные вопросы:**

1. Что такое Git?

Git — это система управления версиями, предназначенная для отслеживания изменений в исходном коде программного обеспечения и координации работы нескольких программистов над одним проектом. Она позволяет отслеживать изменения в коде, возвращаться к предыдущим версиям, объединять изменения от разных разработчиков и многое другое.

1. Для чего команда разработки использует Git?

Команда разработки использует Git для эффективного управления версиями и координации работы над программным обеспечением. Git предоставляет средства отслеживания изменений в коде, позволяет создавать ветви для параллельной разработки, а также управлять и объединять изменения от различных участников проекта. Это обеспечивает надежное и структурированное сотрудничество в команде, позволяет возвращаться к предыдущим версиям кода, а также обеспечивает безопасное внесение и тестирование изменений перед интеграцией в общий кодовую базу. Git также предоставляет возможности работы с удаленными репозиториями, что позволяет разработчикам с легкостью обмениваться своей работой и поддерживать ее в синхронизированном состоянии.

1. Основные команды и операции, необходимые для работы в Git.

⦁ git init - инициализация нового репозитория.

⦁ git clone - клонирование удаленного репозитория на локальную машину.

⦁ git add - добавление изменений в текущей директории в индекс.

⦁ git commit - фиксация изменений в индексе.

⦁ git push - отправка локальных изменений в удаленный репозиторий.

⦁ git pull - получение и объединение изменений из удаленного репозитория в локальный репозиторий.

⦁ git branch - создание, удаление и список веток.

⦁ git merge - объединение другой ветки с текущей веткой.

⦁ git checkout - переключение между ветками и восстановление файлов из репозитория.

⦁ git status - отображение состояния файлов в рабочем каталоге и индексе.

1. Что значит инициализация репозитория, какая команда для этого используется?

Инициализация репозитория в Git означает создание нового репозитория или превращение существующего каталога в Git-репозиторий. Когда вы инициализируете репозиторий, Git создает необходимую структуру данных и файлы для отслеживания изменений в вашем проекте. Это позволяет вам использовать Git для управления версиями кода. Для инициализации нового репозитория в Git используется команда git init. Выполнение этой команды внутри каталога проекта создает скрытую подкаталог .git, который содержит все необходимые файлы и данные для работы с Git.