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 «Операционные системы»**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Светлакова Софья Андреевна  1 курс, группа ПИЖ-б-о-22-1,  09.03.04 «Программная инженерия», направленность (профиль) «Разработка и сопровождение программного обеспечения», очная форма обучения  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Доцент кафедры инфокоммуникаций Воронкин Роман Александрович  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

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

Ставрополь, 2023 г.

**Тема:** Исследование основных возможностей Git и GitHub.

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

**Порядок выполнения работы**

1. Создала репозиторий.

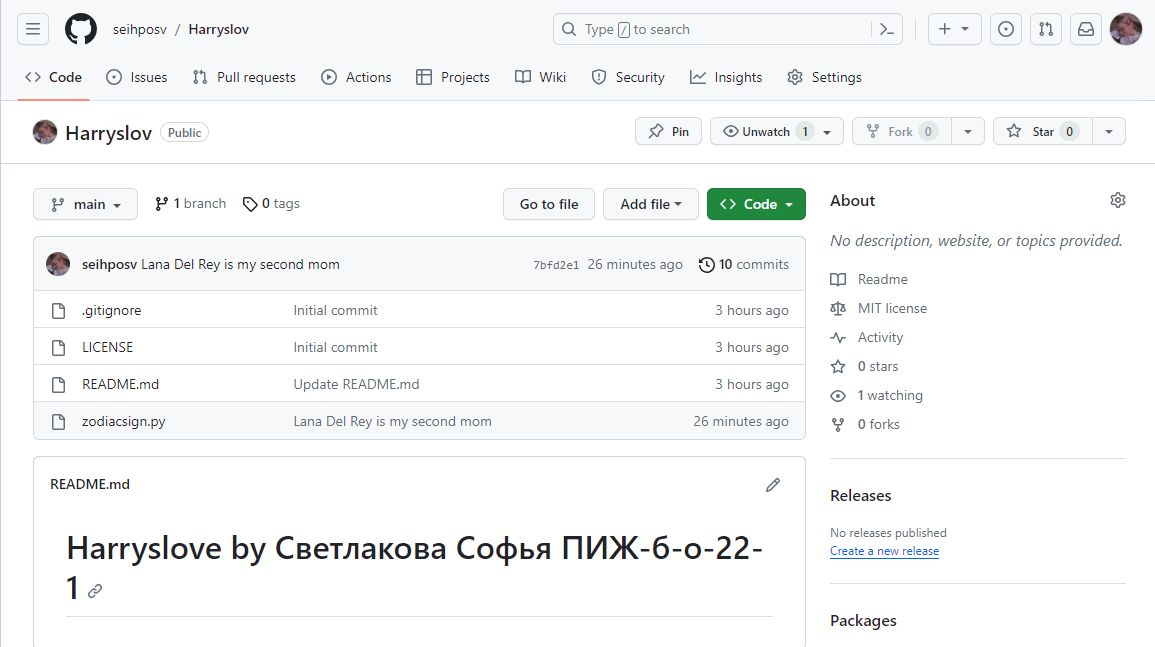


Рисунок 1 - Создание репозитория

1. Добавила необходимые правила в .gitignore.

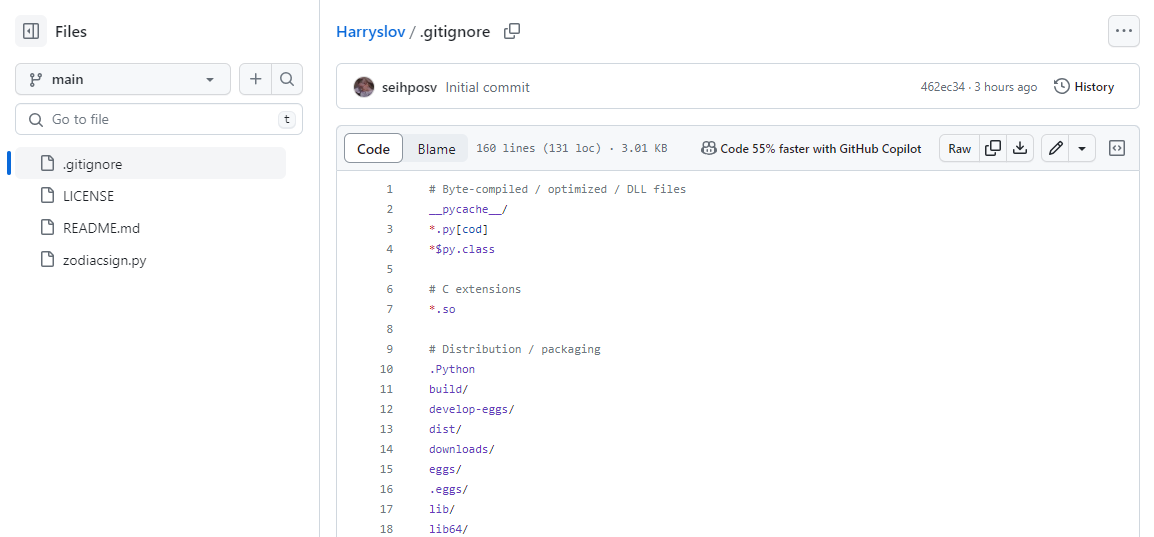


Рисунок - 2 Правила в .gitignore

1. Добавила требуемую информацию в README.md.

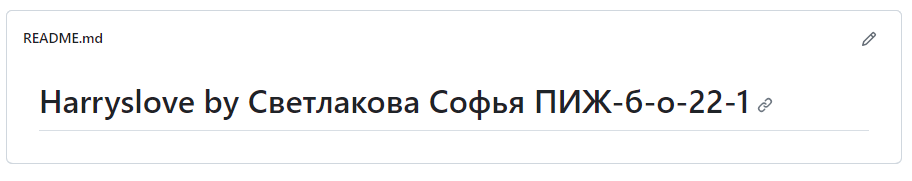


Рисунок - 3 Файл README.md

1. Написала программу. При написании программы фиксировала изменения в локальном репозитории с помощью команды git commit.

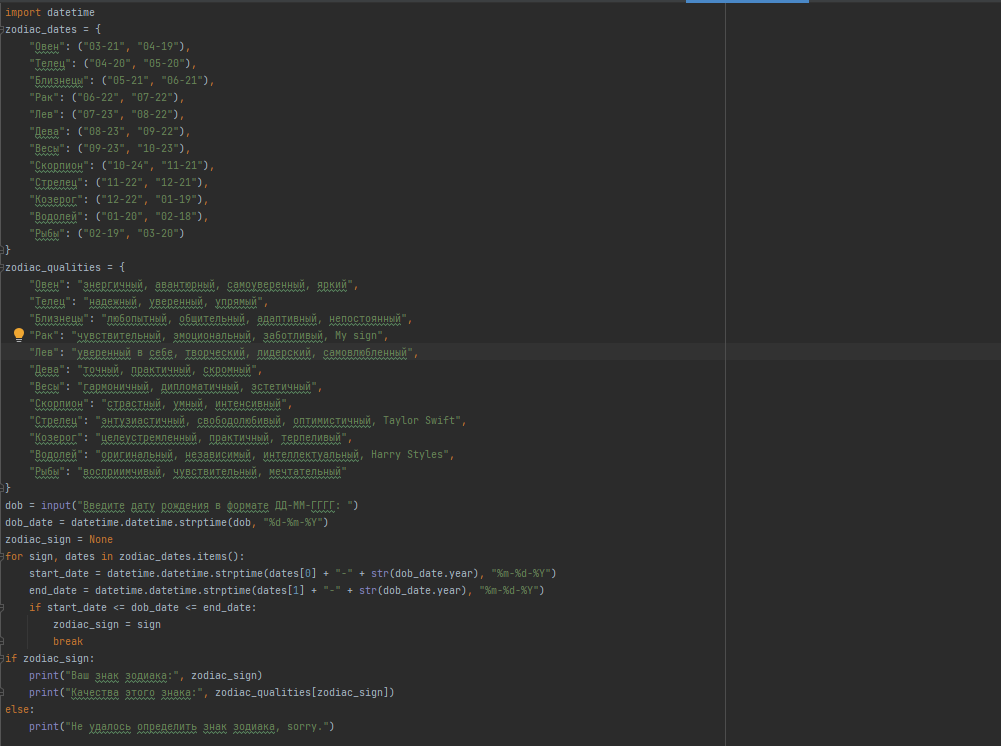


Рисунок 4 – Программа

1. Делаю 7 коммитов.

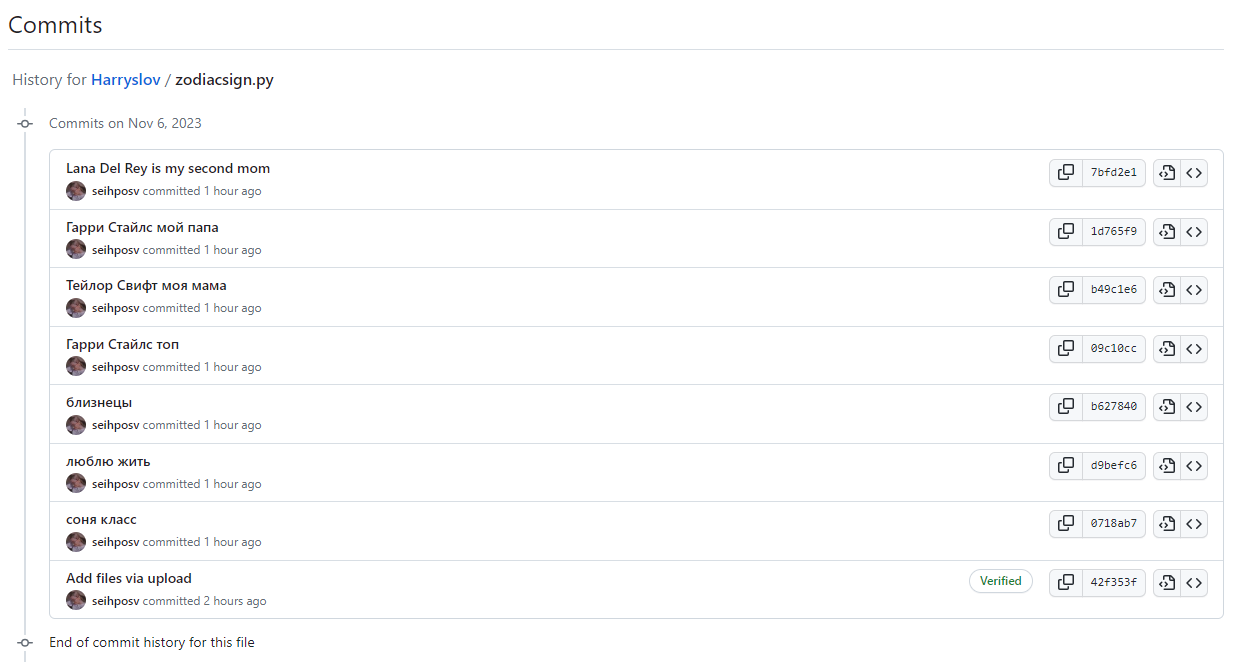


Рисунок 5 – История коммитов

**Ответы на контрольные вопросы**

1. Системы контроля версий — это программные инструменты, помогающие командам разработчиков управлять изменениями в исходном коде с течением времени. В свете усложнения сред разработки они помогают командам разработчиков работать быстрее и эффективнее.
2. Недостатки локальных СКВ: возможность потери данных вследствие возникновения физических поломок оборудования; отсутствие возможности совместной разработки. Недостатки централизированных СКВ: отсутствие доступа к данным при сбое работы сервера; довольно низкая скорость работы (из-за возникновения сетевых задержек).
3. Распределенные СКВ.
4. Рабочий код хранится на нескольких компьютерах, а история всех версий хранится как на удалённом сервере, так и на каждом из этих компьютеров.
5. 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. У Git есть три основных состояния, в которых могут находиться ваши файлы: зафиксированное (committed), изменённое (modified) и подготовленное (staged). Если определённая версия файла есть в Git-директории, эта версия считается зафиксированной. Если версия файла изменена и добавлена в индекс, значит, она подготовлена. И если файл был изменён с момента последнего распаковывания из репозитория, но не был добавлен в индекс, он считается изменённым.
7. На странице профиля отображаются сведения о вашей работе через репозитории, которые вас интересуют, вклад, который вы сделали, и беседы, в которых вы участвовали.
8. Публичный (public) и закрытый (private).
9. Регистрация и настройка, создания репозитория, клонирование репозитория, создание и изменение файлов, добавление и фиксация изменений, отправка изменений на GitHub.
10. Первое, что нужно сделать - указать имя и адрес электронной почты пользователя.
11. Нажать на кнопку создания репозитория, дать имя репозиторию, выбрать видимость репозитория, если необходимо отметить флажки создания файлов README и .gitignore.
12. MIT License, GNU General Public License (GPL), Apache License, Creative Commons.
13. Для клонирования необходимо найти кнопку Clone или Code и щелкнуть по ней, чтобы отобразить адрес репозитория для клонирования. Далее необходимо открыть командную строку или терминал и перейти в каталог, куда необходимо скопировать хранилище. Затем написать git clone и ввести адрес. Клонирование репозитория GitHub позволяет сохранить локальную копию проекта, работать с кодом, получать изменения и сотрудничать с другими разработчиками.
14. Использовать команду git status.
15. После выполнения следующих операций состояние локального репозитория Git изменяется следующим образом:

* Добавление/изменение файла: Когда добавляется или изменяется файл в локальном репозитории Git, Git определяет этот файл как измененный. Однако он еще не отслеживается и не готов для коммита.
* Добавление файла под версионный контроль: Выполнение команды `git add ` добавляет указанный файл в индекс Git. Это означает, что файл теперь отслеживается Git и готов для фиксации. Состояние изменения файла изменяется с "изменен" на "отслеживается".
* Фиксация изменений (коммит): Когда выполняется команда `git commit, Git создает коммит, содержащий фиксацию изменений. Все файлы, находящиеся в индексе, включаются в коммит. После коммита состояние файлов изменяется на зафиксированный (committed).
* Отправка изменений на сервер: Команда git push используется для отправки локальных коммитов на удаленный сервер, например, на GitHub. После успешного выполнения команды Git синхронизирует внешний репозиторий с локальным репозиторием. После отправки изменений состояние удаленного репозитория становится идентичным состоянию локального репозитория.

1. Сначала на первом и втором компьютере необходимо выполнить команду git clone. После этого на обоих компьютерах есть локальные копии репозитория, при этом потребуется регулярно синхронизировать изменения между компьютерами и репозиторием GitHub. Для этого необходимо добавить все изменения в локальном репозитории с помощью команды git add, затем ввести команду git commit для фиксации изменений, после для отправки изменений на GitHub использовать команду git push. Чтобы загрузить последние изменения необходимо использовать git pull.
2. GitLab, CodeBase, SourceForge. GitHub делает упор на высокую доступность и производительность своей инфраструктуры и делегирует другие сложные функции сторонним инструментам. GitLab, наоборот, фокусируется на включении всех функций на одной проверенной и хорошо интегрированной платформе; он обеспечивает все для полного жизненного цикла DevOps под одной крышей. Что касается популярности, GitHub определенно превосходит GitLab.
3. GitHub Desktop обеспечивает удобный способ работы с Git и Github с помощью графического интерфейса. Можно создавать новые репозитории, клонировать существующие, отслеживать изменения, создавать ветки, коммиты и многое другое. Операции Git, такие как создание веток и коммитов, отображаются в интерфейсе приложения, что упрощает их выполнение и позволяет работать с Git без использования командной строки.