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 «Программирование на Python»
Вариант___

Выполнил:
Иващенко Олег Андреевич
2 курс, группа ИВТ-б-о-22-1,
09.03.02 «Информационные и вычислительные машины»,
направленность (профиль)
«Программное обеспечение средств вычислительной техники и автоматизированных систем»

_____
(подпись)

Руководитель практики:
Воронкин Роман Александрович,
доцент кафедры инфокоммуникаций

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2022 г.

Тема: «Исследование основных возможностей Git и GitHub»

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

Порядок выполнения работы:

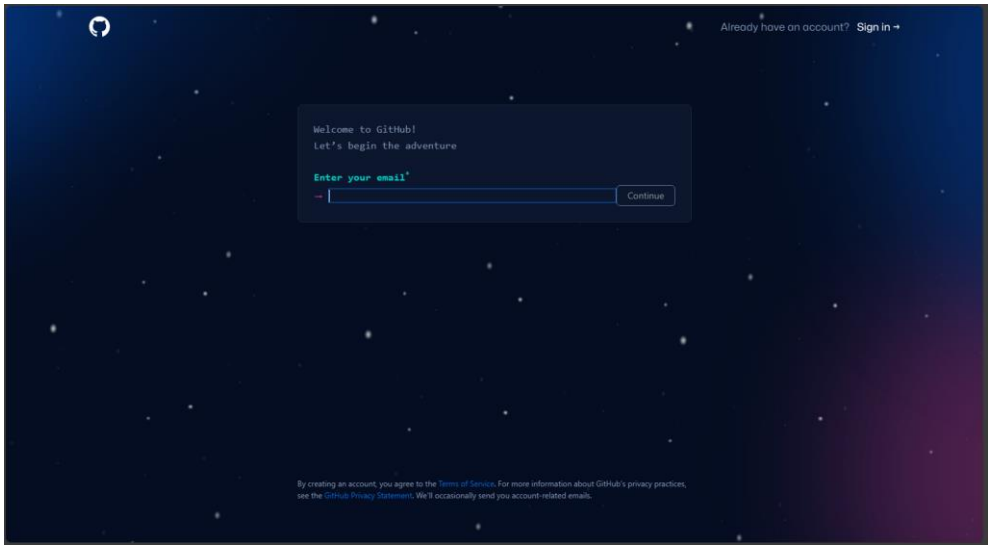

Рисунок 1 – Регистрация на GitHub

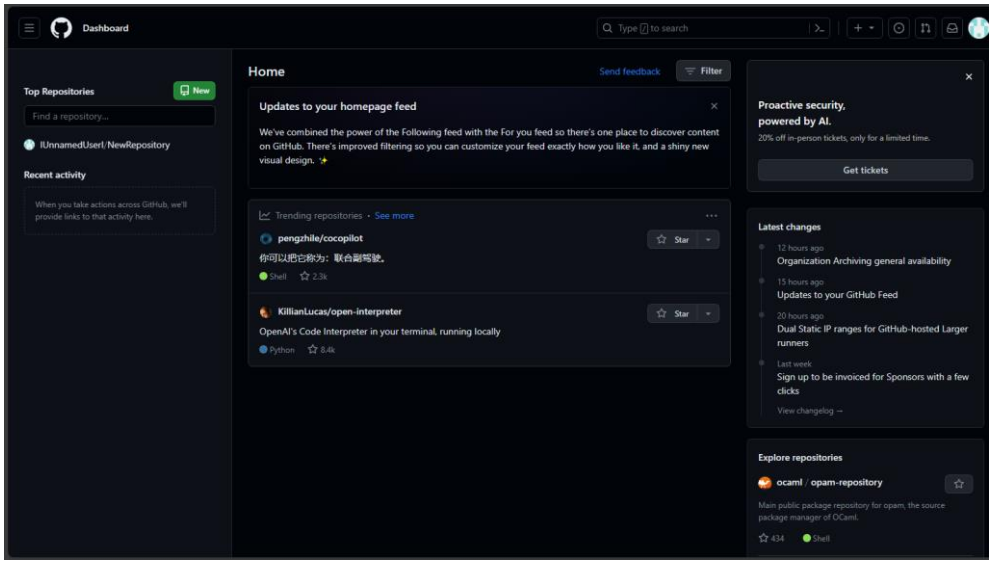

Рисунок 2 – Главная страница GitHub

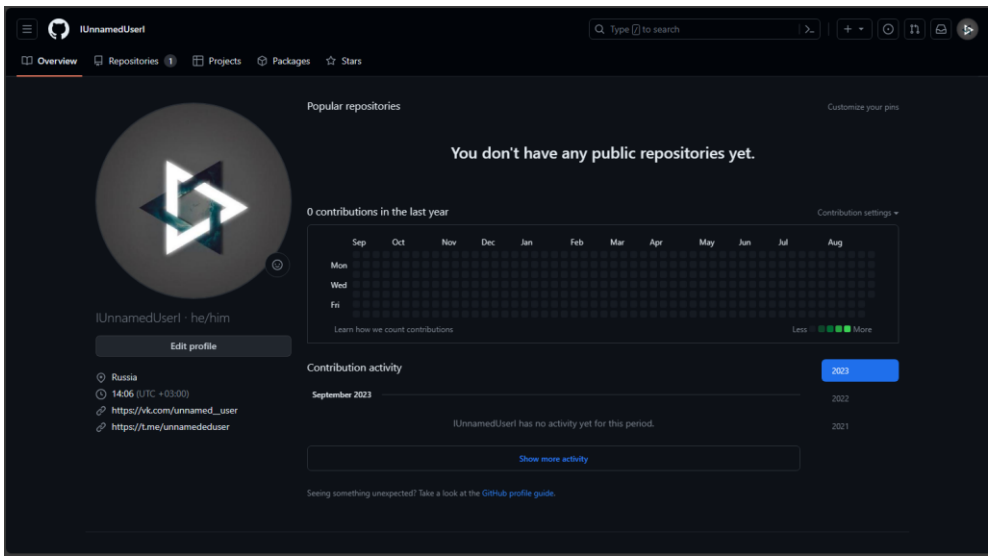

Рисунок 3 – Профиль GitHub

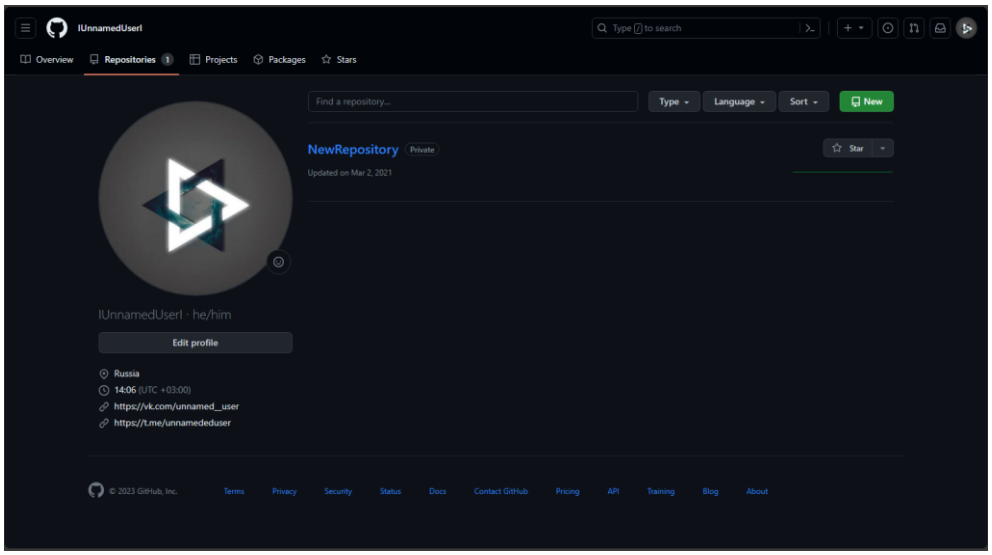

Рисунок 4 – Репозитории

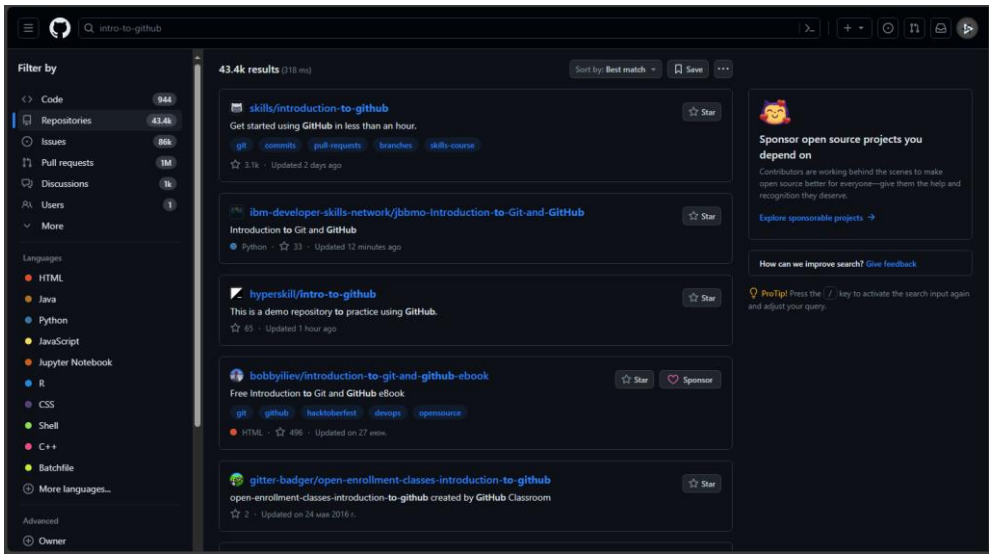

Рисунок 5 – Результат поиска по запросу «intro-to-github»

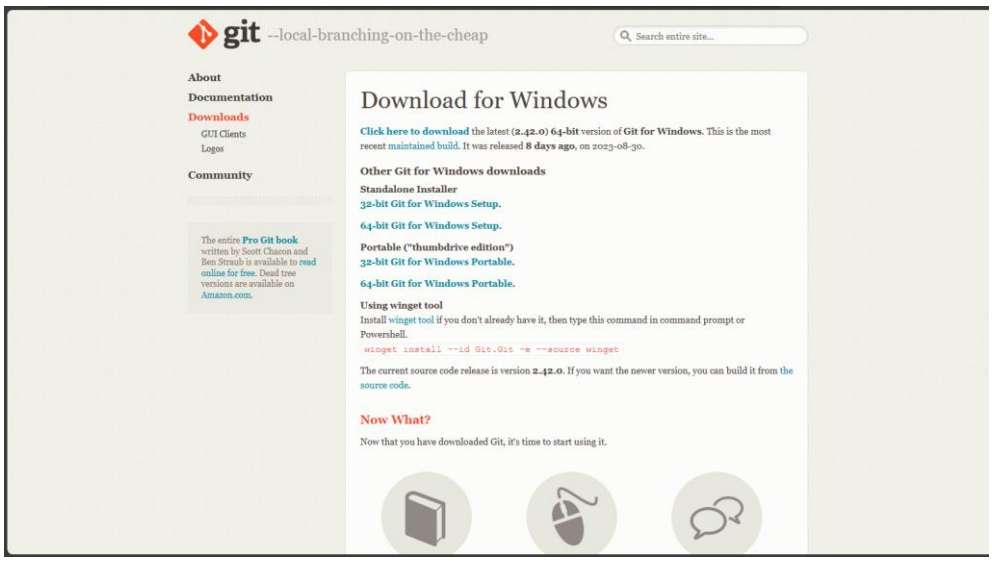

Рисунок 6 – Страница загрузки локального Git

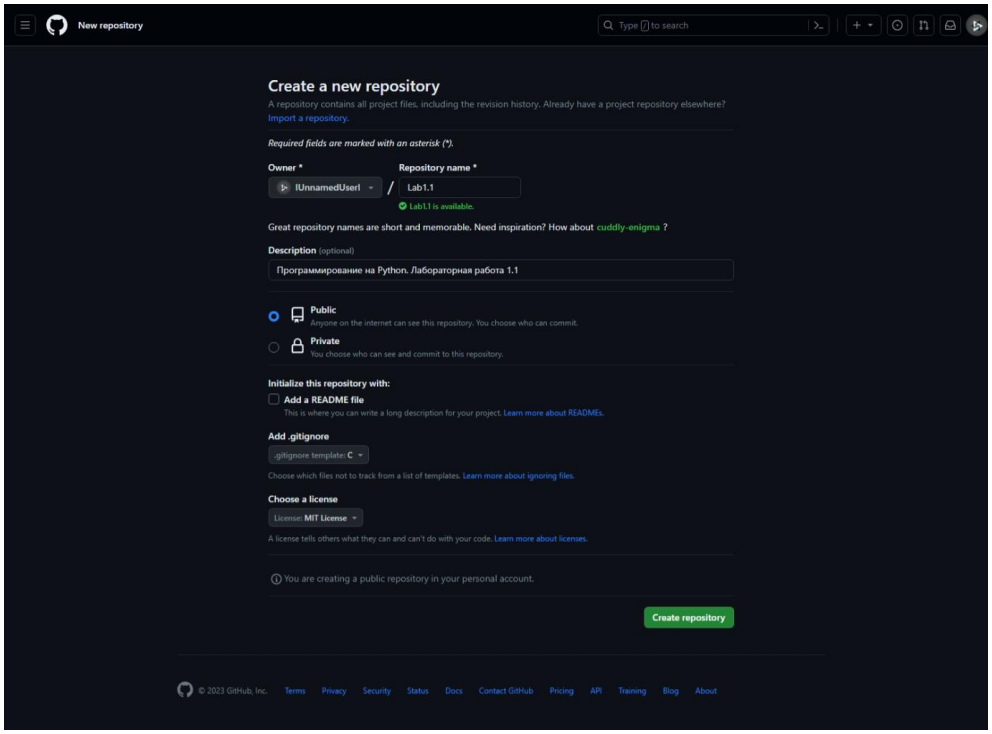

Рисунок 7 – Создание репозитория

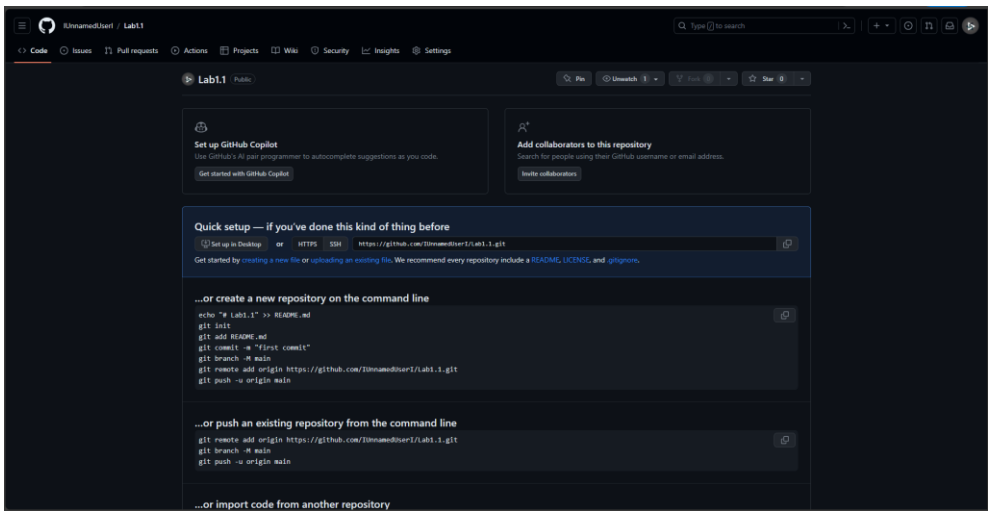

Рисунок 8 – Страница репозитория

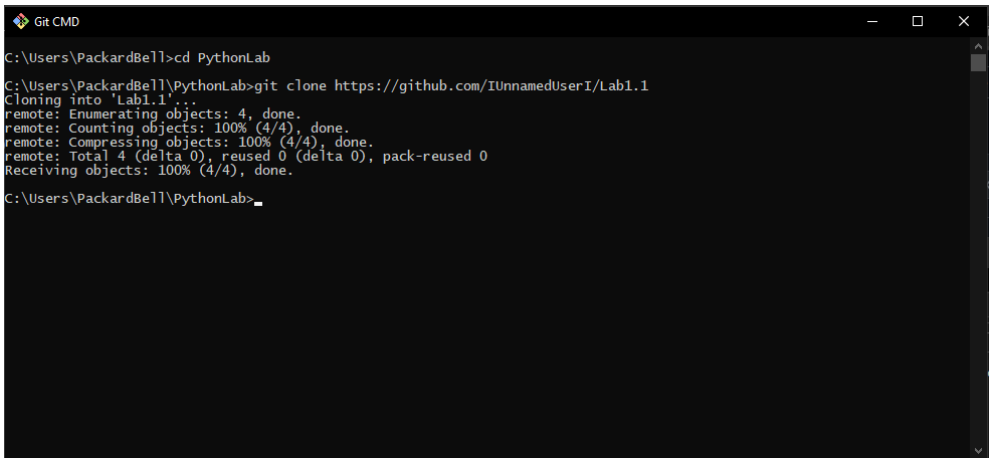

Рисунок 9 – Клонирование репозитория

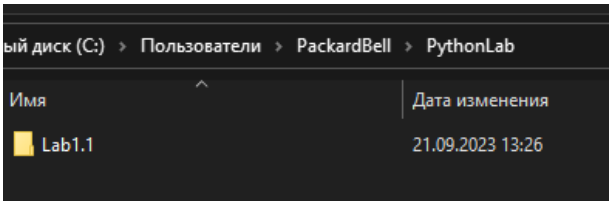

Рисунок 10 – Результат клонирования

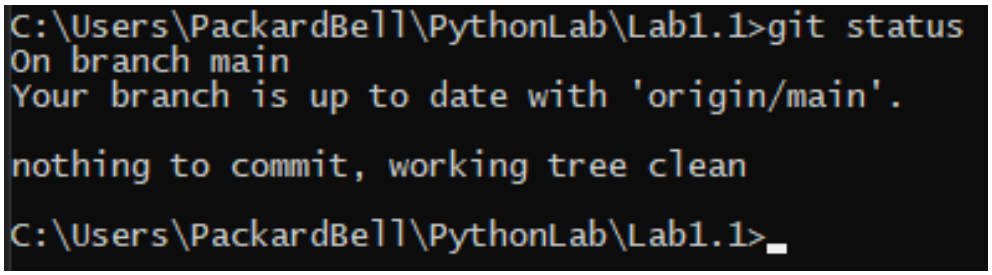

Рисунок 11 – Статус Git

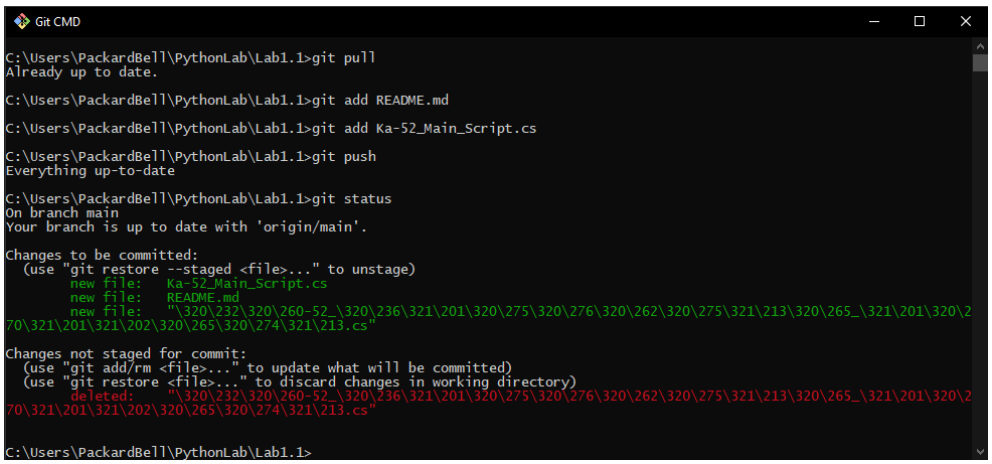

Рисунок 12 – Добавление и изменение файлов

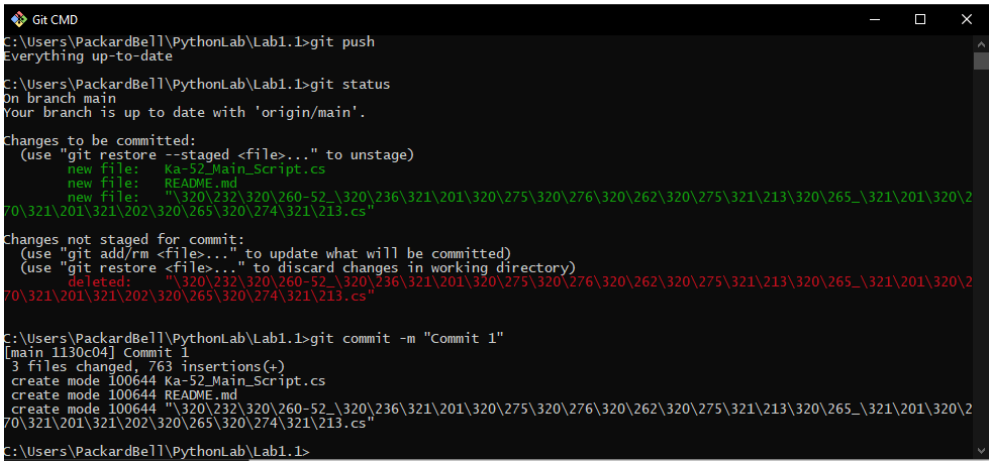

Рисунок 13 – Фиксация изменений

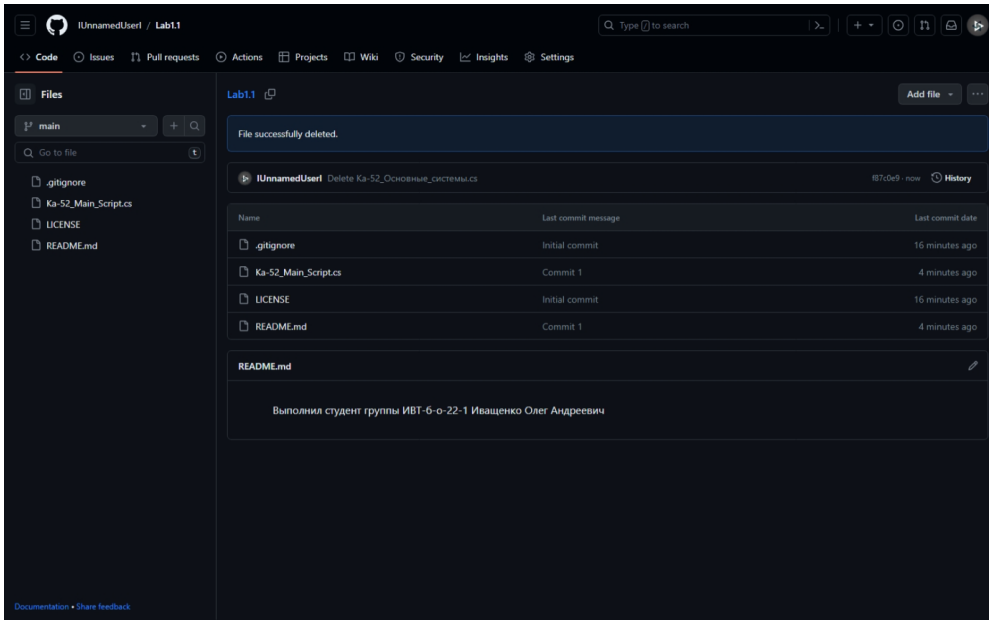

Рисунок 14 – Результат внесённых изменений

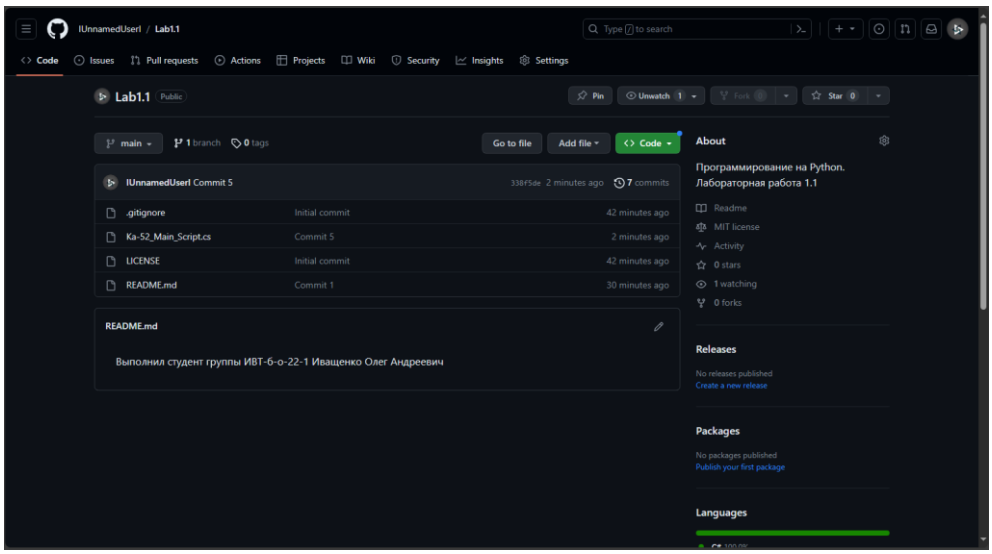

Рисунок 15 – Итоговый репозиторий

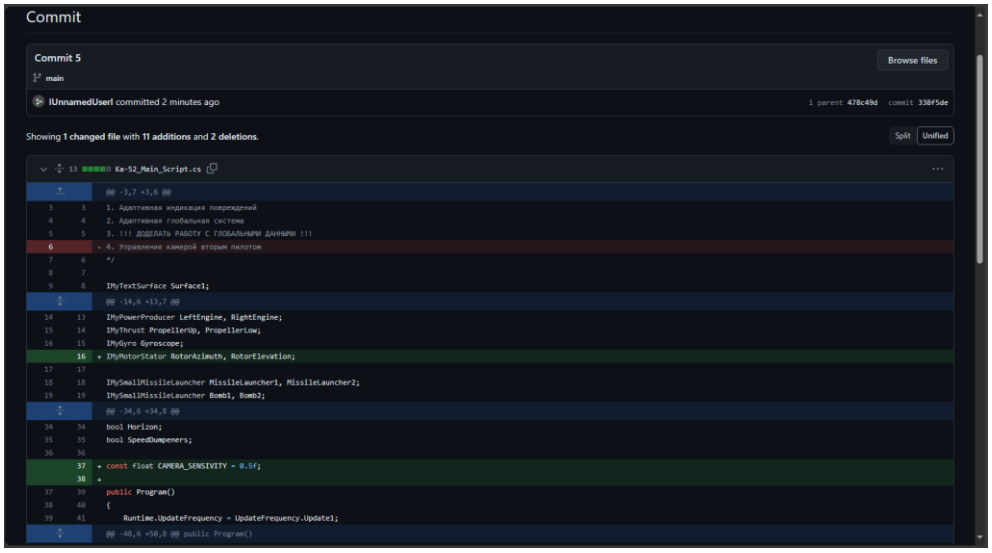

Рисунок 16 – Изменения в файле

Ответы на контрольные вопросы:

1. Что такое СКВ и каково ее назначение?

Система контроля версий (СВК) – 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?

Единая точка отказа, возможность замены или копирования не тех файлов, ограниченность доступа к данными только для участников проекта,

сложность организации работы над проектом, риск потери данных при сбоях оборудования или ПО.

3. К какой СКВ относится Git?

Git поддерживает несколько режимов работы, включая локальную СКВ (LocalSVN), глобальную СКВ (GlobalSVN) и гибридную СКВ (HybridSVN).

4. В чем концептуальное отличие Git от других СКВ?

Git отличается от других СКВ следующими концептуальными отличиями:

- Управление изменениями – Git использует механизм фиксации изменений, называемый коммитом, который гарантирует, что изменения будут сохранены в истории изменений. Это отличает Git от других СКВ, которые используют инкрементальную систему контроля версий, где каждый новый файл создаётся отдельно и не имеет связи с предыдущими файлами;
- Работа в команде – Git предоставляет возможность совместной работы нескольких разработчиков над одним проектом: каждый разработчик может работать над своим репозиторием, а все изменения объединяются в один общий репозиторий;
- Автоматическое управление ветками: Git автоматически создаёт новые ветки на основе изменений в репозитории и переключается между ними по мере необходимости, что позволяет разработчикам быстро отслеживать изменения в проекте и возвращаться к предыдущим версиям;
- Интеграция с другими инструментами разработки: Git интегрируется с различными инструментами разработки, такими как IDE, менеджеры пакетов и системы сборки.

5. Как обеспечивается целостность хранимых данных в Git?

Хранение данных в Git осуществляется в виде отдельных файлов, называемых ветками. Каждая ветка содержит набор изменений, которые были

внесены в репозиторий за определенный период времени. Когда разработчики вносят изменения в репозиторий, эти изменения сохраняются в соответствующей ветке.

Для обеспечения целостности данных в Git используются механизмы фиксации изменений, такие как коммиты. Коммит фиксирует изменение в истории изменений, которое было внесено в репозиторий. При следующем выполнении команды git будет искать изменения, сделанные после последнего зафиксированного коммита, и сравнивать их с текущими изменениями.

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

У Git есть три основных состояния, в которых могут находиться файлы:

- Зафиксированный – файл уже сохранён в локальной базе;
- Изменённый – относятся изменённые файлы, которые не были ещё зафиксированы;
- Подготовленные – это изменённые файлы, отмеченные для включения в следующий коммит.

7. Что такое профиль пользователя в GitHub?

Профиль – это публичная страница на GitHub, аналогичная социальным сетям.

8. Какие бывают репозитории в GitHub?

9. Укажите основные этапы модели работы с GitHub.

Основные этапы модели работы с GitHub включают следующее:

- Создание репозитория;
- Добавление файлов;
- Управление ветками;
- Объединение веток;
- Мониторинг и отслеживание изменений.

10. Как осуществляется первоначальная настройка Git после установки?

После установки Git необходимо выполнить следующие шаги для начальной настройки:

- Создание локальной копии репозитория (git clone <URL>);
- Добавление файлов в локальную копию (git add – добавление в индекс);
- Создание новой ветки (опционально, git checkout -b <branch_name>);
- Отправить изменения в удалённый репозиторий (git push origin <branch_name>);
- Зафиксировать изменения (git commit).

11. Опишите этапы создания репозитория в GitHub.

Шаги по созданию репозитория в GitHub включают:

- Регистрация аккаунта;
- Переход на страницу создания репозитория;
- Указание имени репозитория, установка параметров.

12. Какие типы лицензий поддерживаются GitHub при создании репозитория?

GitHub поддерживает следующие типы лицензии:

- Свободное программное обеспечение (Free Software License);
- Открытое программное обеспечение (Open Source License);
- Совместная собственность (Shareable Content License);
- Авторские права (Copyright License);
- Образовательные учреждения (Education Institutions License);
- Исследовательские организации (Research Organizations License);

- Конфиденциальность и безопасность (Privacy and Security License);
- Обучение и образование (Teaching and Learning License).

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

Клонирование репозитория осуществляется командой git clone <URL>. Клонирование репозитория Git позволяет работать с ним на удаленном сервере или локальной машине.

14. Как проверить состояние локального репозитория Git?

Проверить состояние локального репозитория Git можно командой 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?

После добавления/изменения файла в локальном репозитории изменения будут добавлены в индекс Git, но не будут зафиксированы

После добавления нового/изменённого файла под версионный контроль с помощью команды git add изменения изменят свой индекс.

После фиксации (коммита) изменений с помощью команд git commit и отправки изменений на сервер с помощью команды git push изменения будут зафиксированы в истории коммитов, а также изменения будут отправлены на сервер.

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.

git clone <URL>

git pull <URL>

git push <URL>

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

Существует множество сервисов, которые предоставляют возможность хранения и управления Git:

- Bitbucket;
- GitLab;
- SourceForge;
- GitLog;
- Gitosis.

Сравнительный анализ GitHub и других сервисов заключается в том, что GitHub является наиболее популярным и широко используемым сервисом для хранения и совместной работы над проектами с открытым исходным кодом. Он имеет простой интерфейс, интегрируется с другими инструментами разработки и предоставляет удобный способ обмена сообщениями между разработчиками. Однако, он может быть несколько менее гибким по сравнению с другими сервисами, такими как Bitbucket или SourceForge, которые предлагают более широкий набор функций и возможностей.

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.

Существует несколько программных средств с графическим интерфейсом пользователя для работы с Git:

- Visual Studio Code;
- Sublime Text;
- Atom;
- Brackets;
- Notepad++.

Выводы: В процессе выполнения лабораторной работы исследовал базовые возможности системы контроля версий Git и веб-сервиса для хостинга IT-проектов GitHub.