

**ОТЧЕТ О ВЫПОЛНЕНИИ**

**практической работы №5**

**по дисциплине / МДК 3.1 «Моделирование и анализ ПО»**

Выполнил: ст.гр. 2-ИСП11-31ВБ

Алирзаева З.Н., Рязанова Ю.Р.

Специальность: 09.02.07 Информационные системы и программирование

Проверил: преп. Боклач Б.И.

Москва

2023

**Тема:** Планирование code-review

**Цель:** Получение практических навыков подготовки среды совместной разработки для проведения инспекции программного года

**Ход работы:**

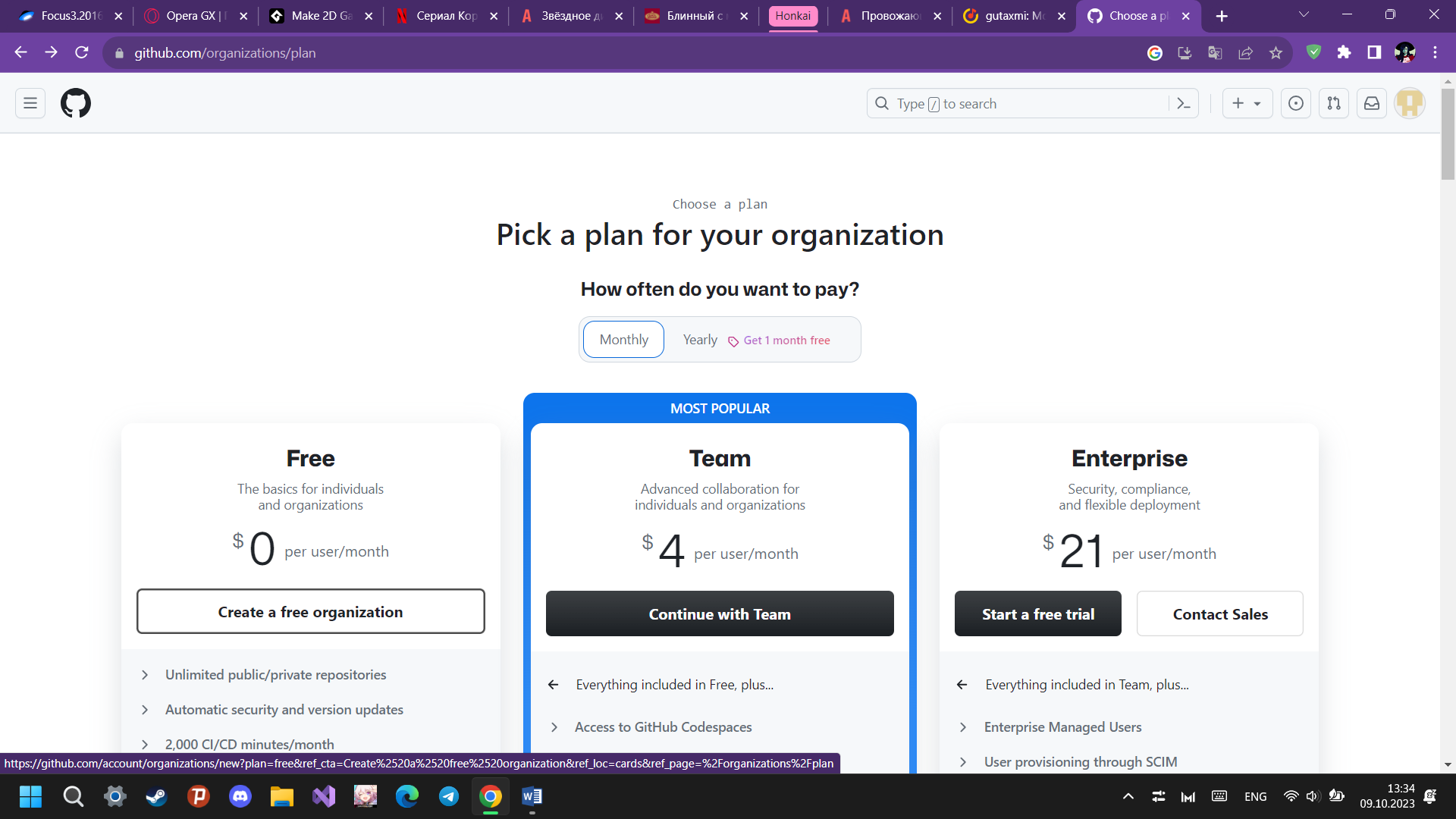


Рис. 1

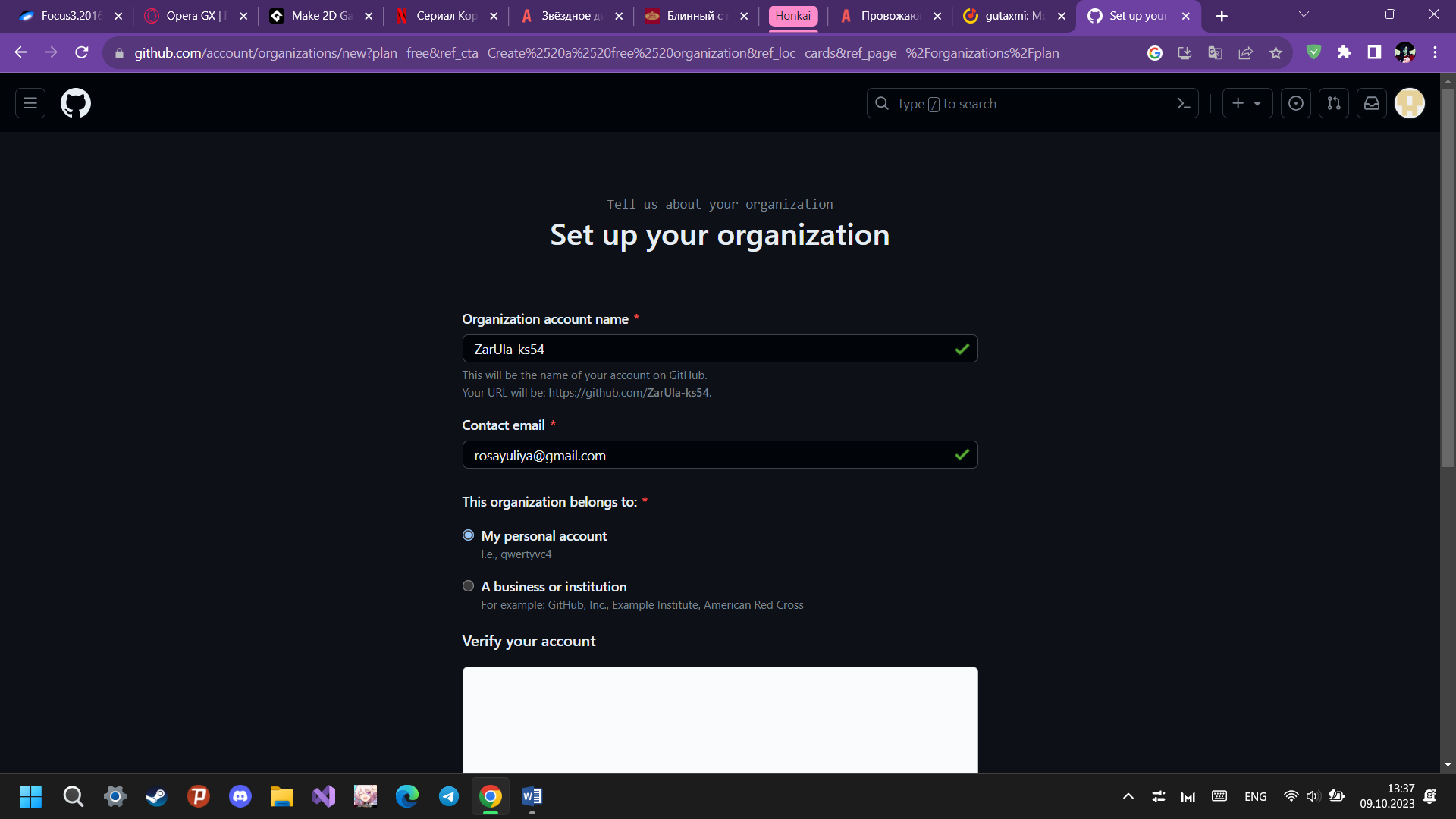


Рис. 2

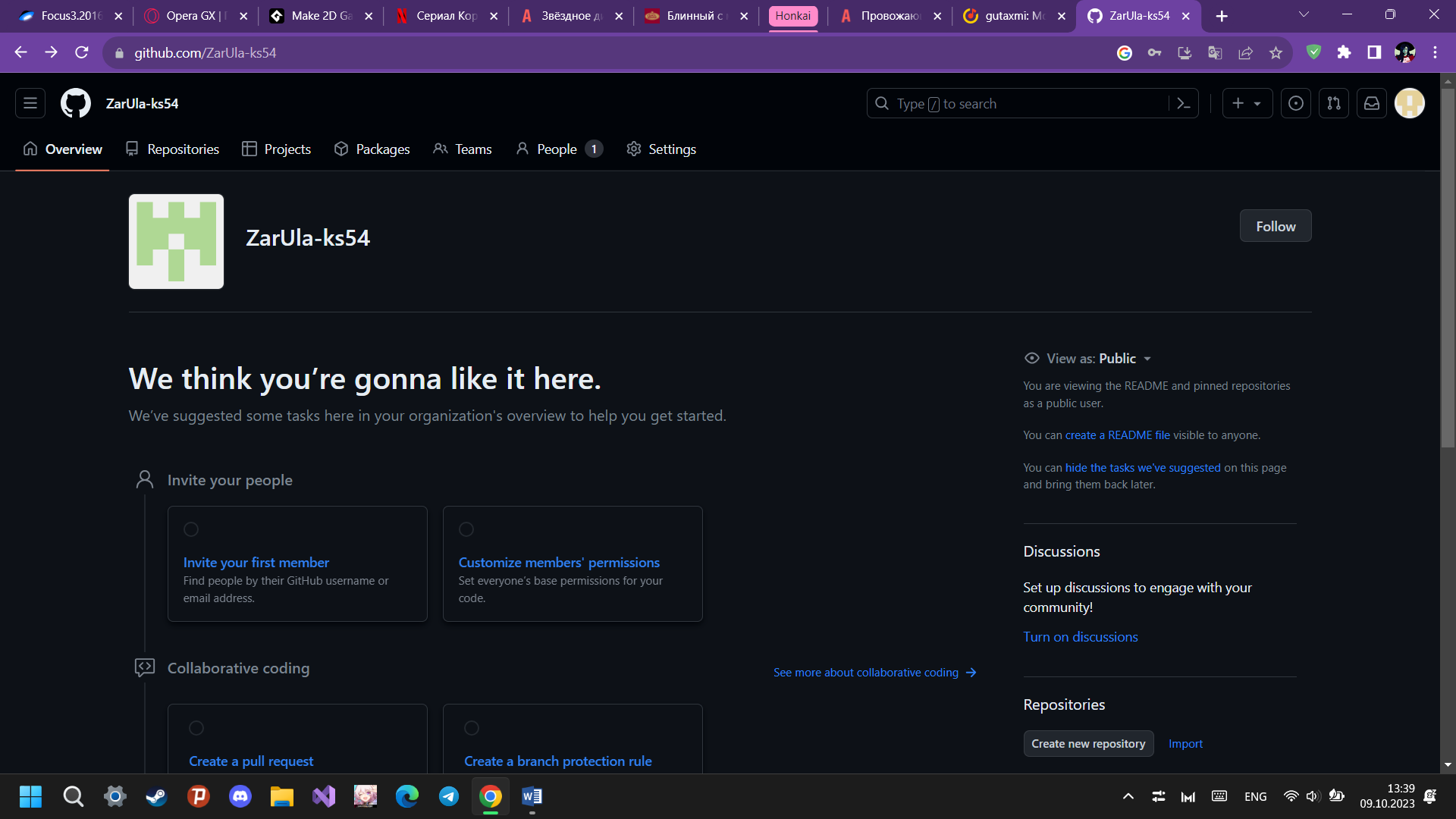


Рис. 3

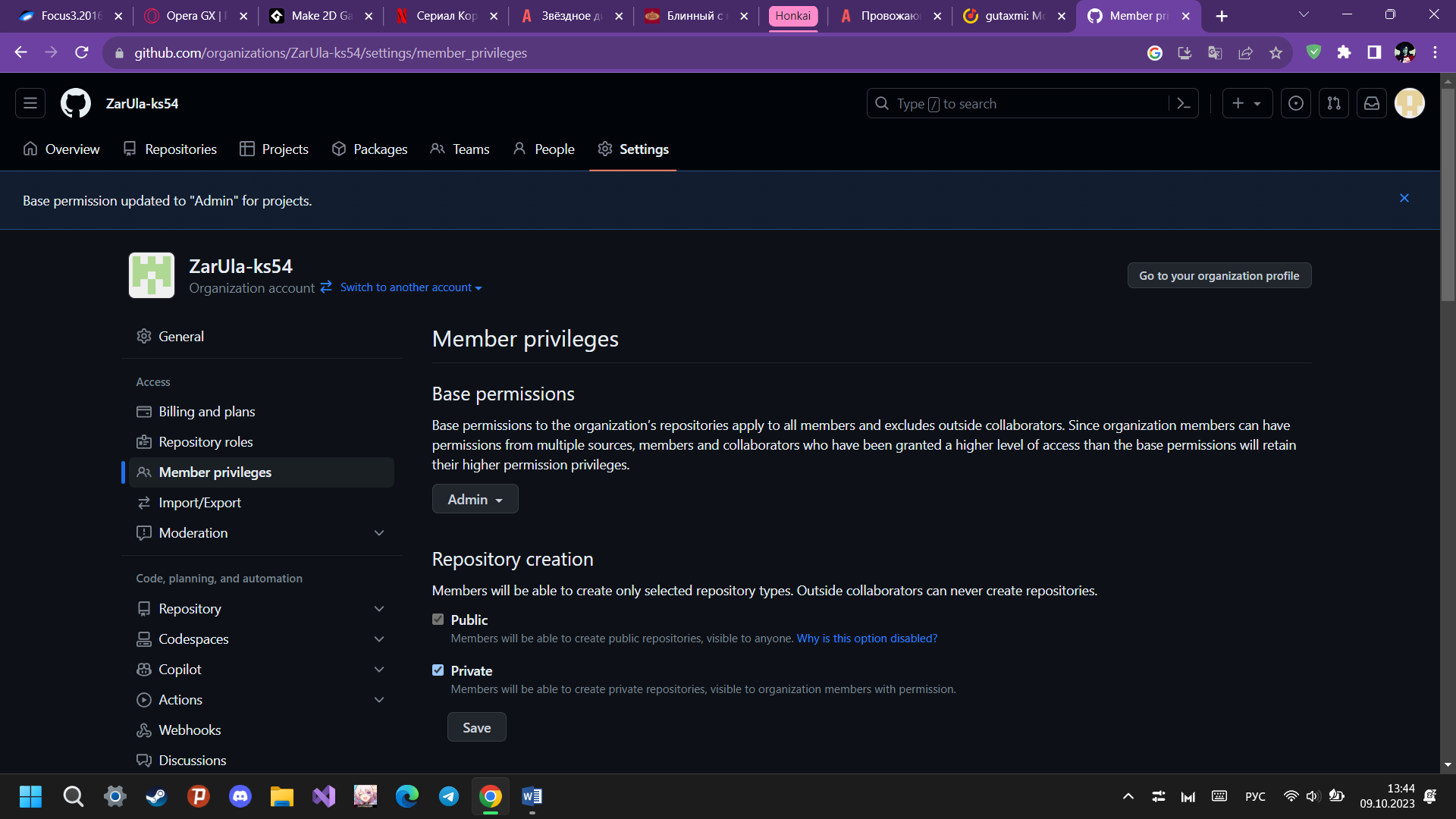


Рис. 4

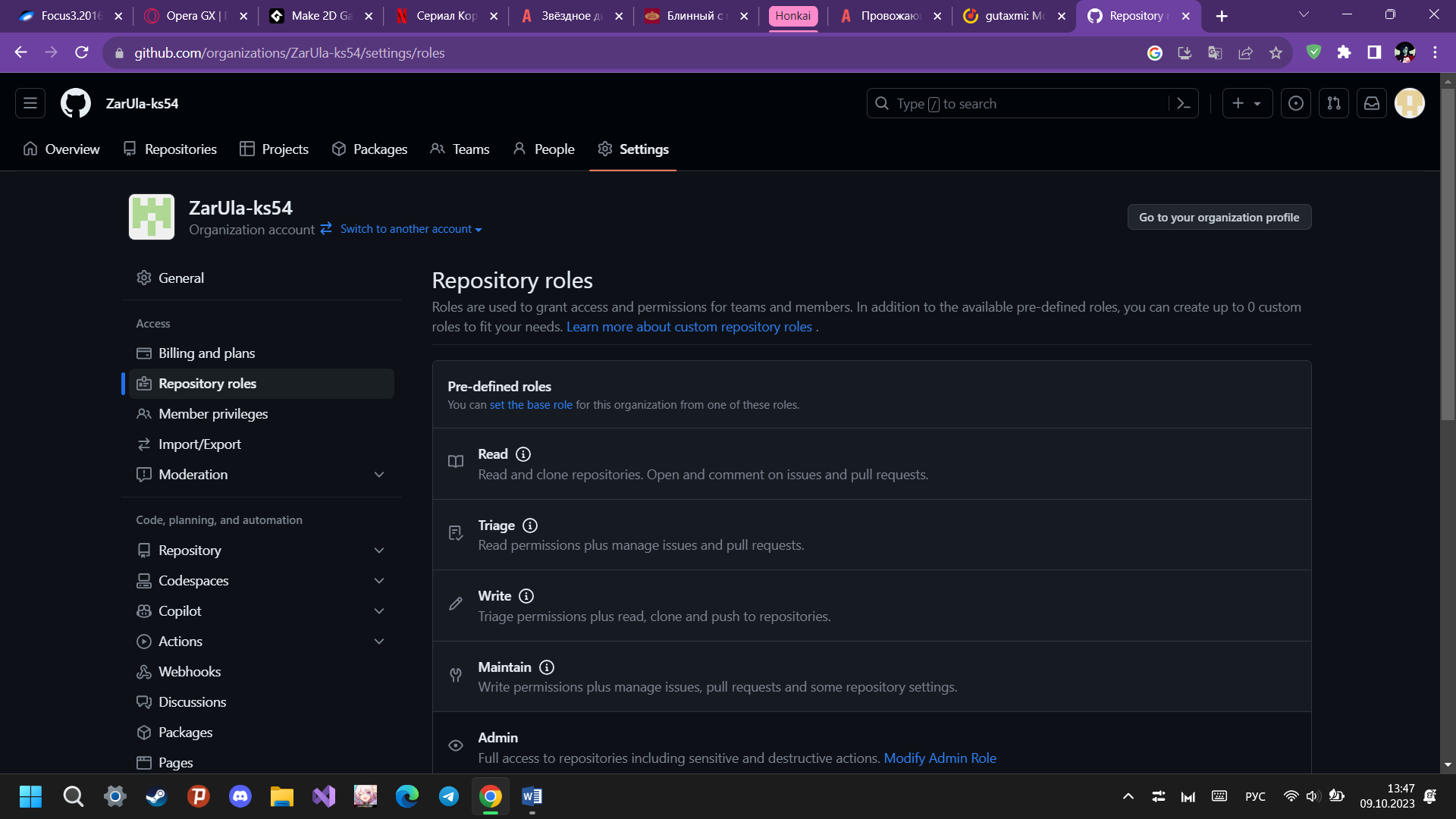


Рис. 5

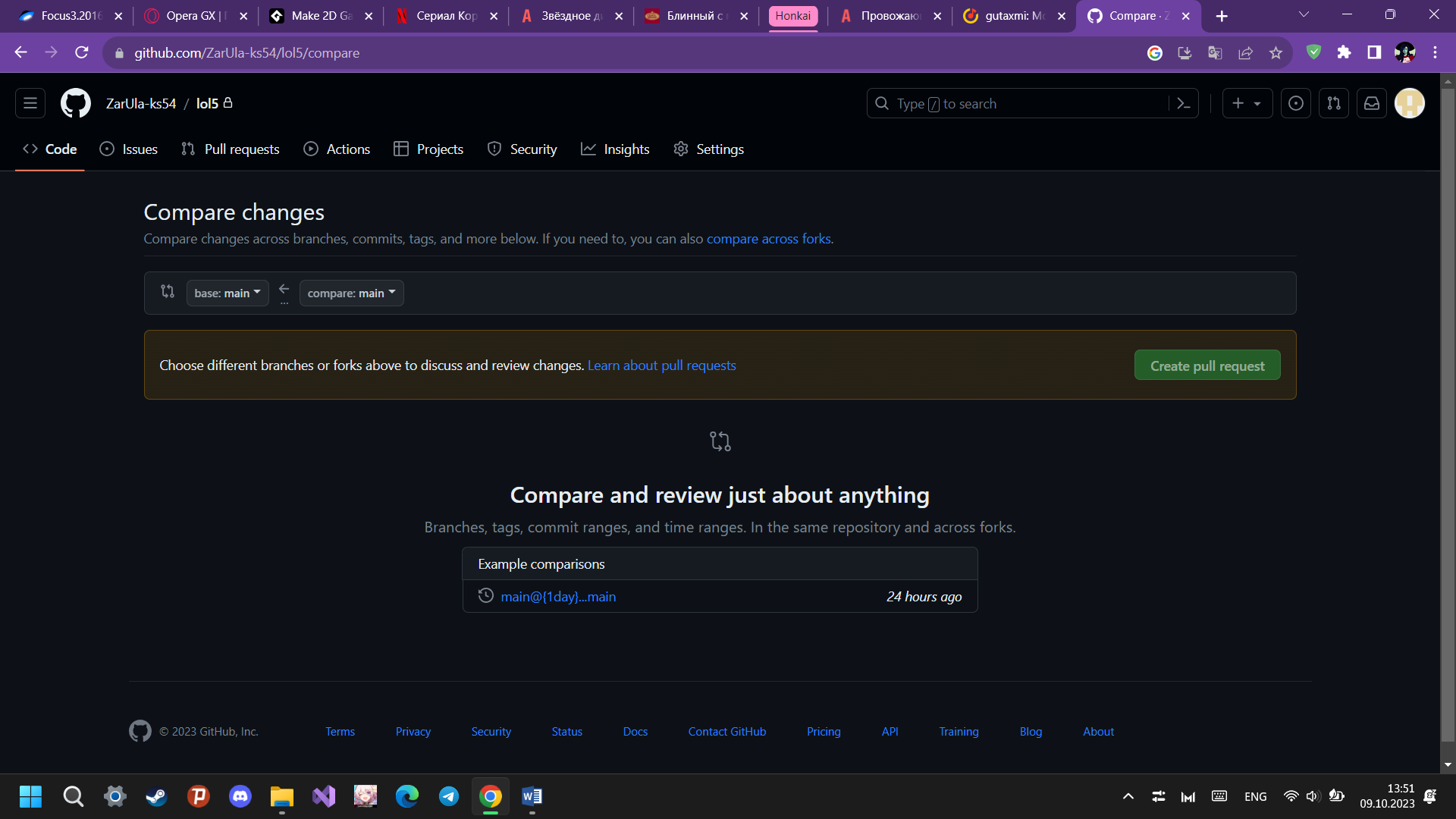


Рис. 6

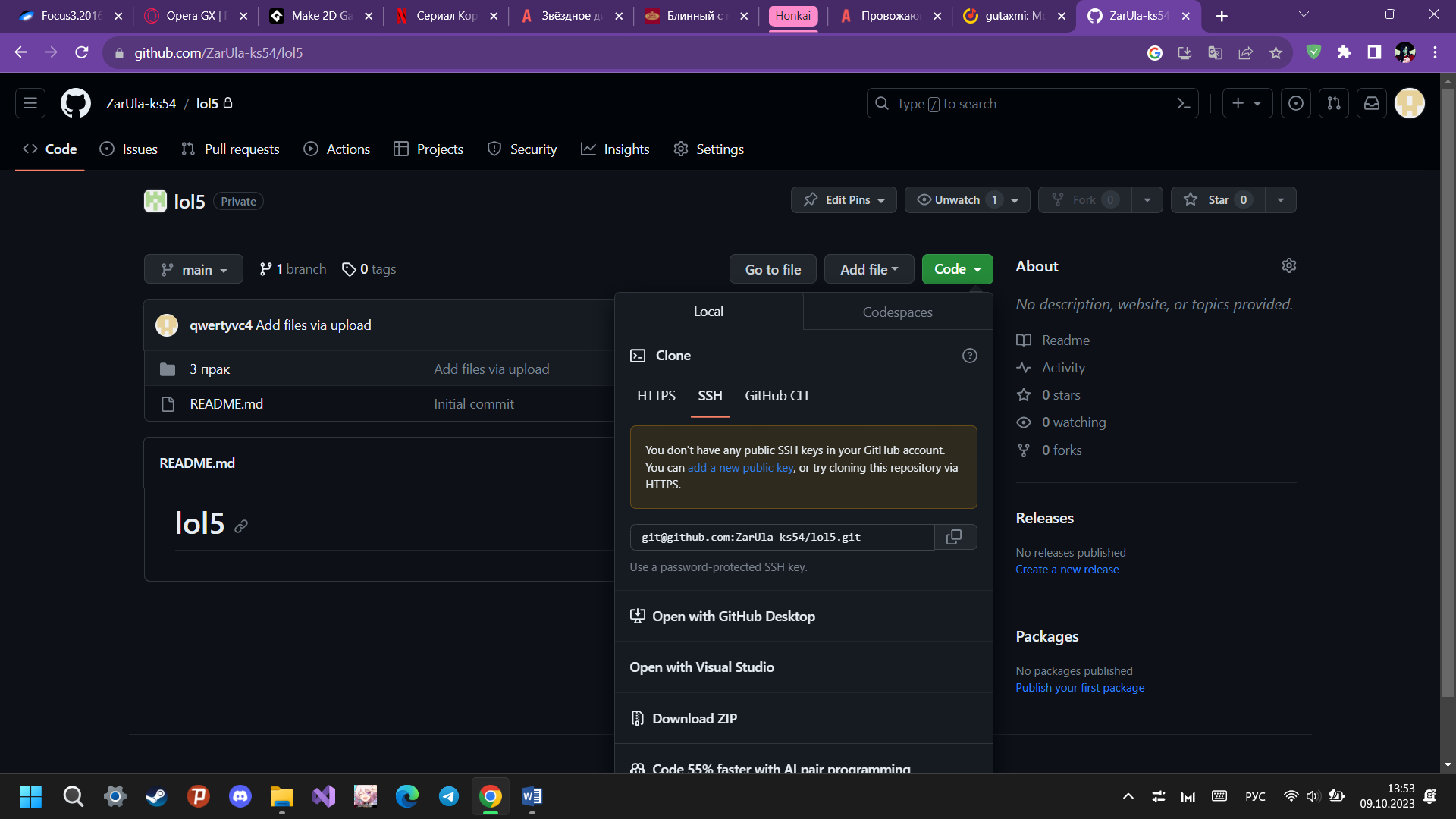


Рис. 7

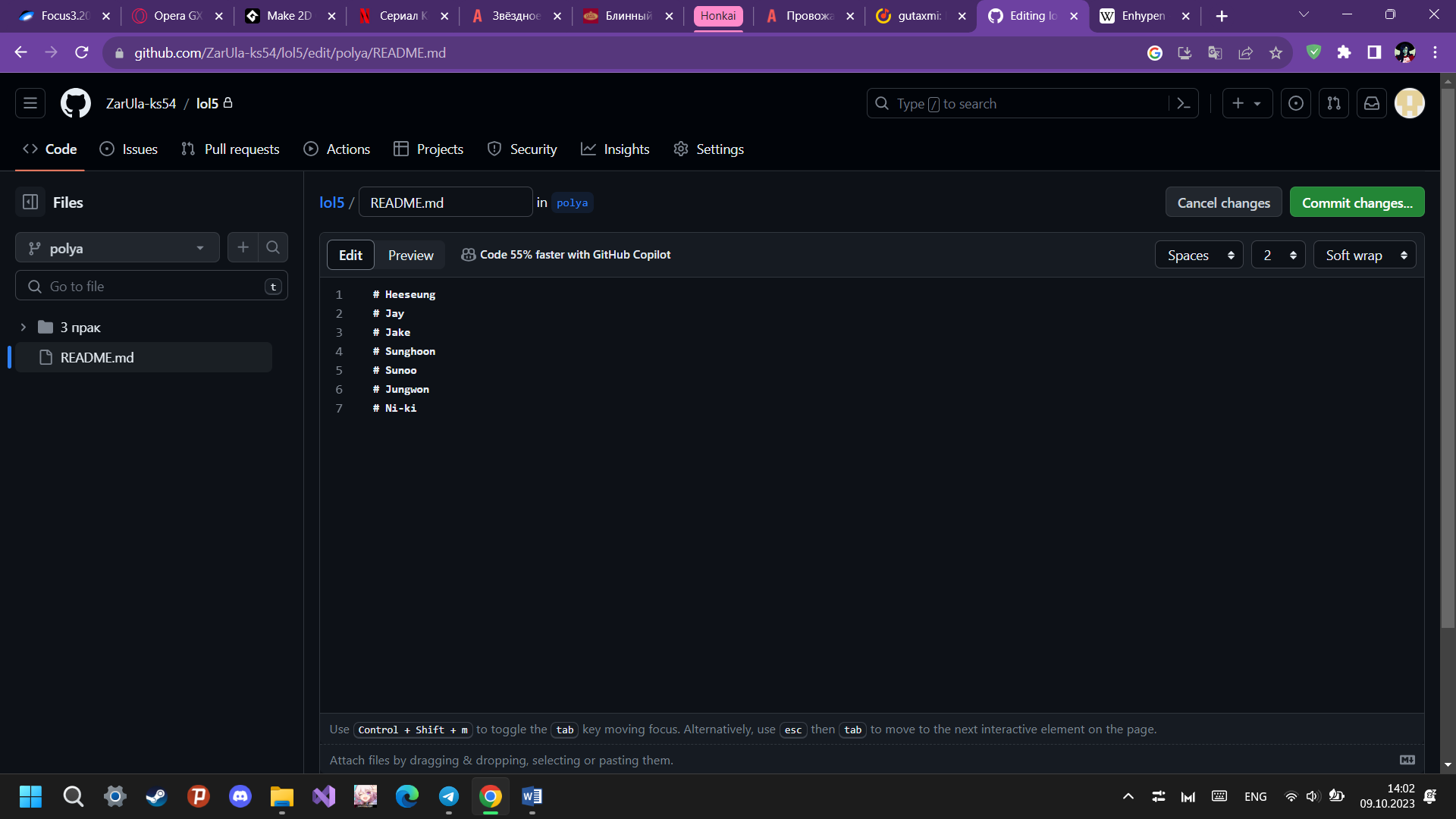


Рис. 8

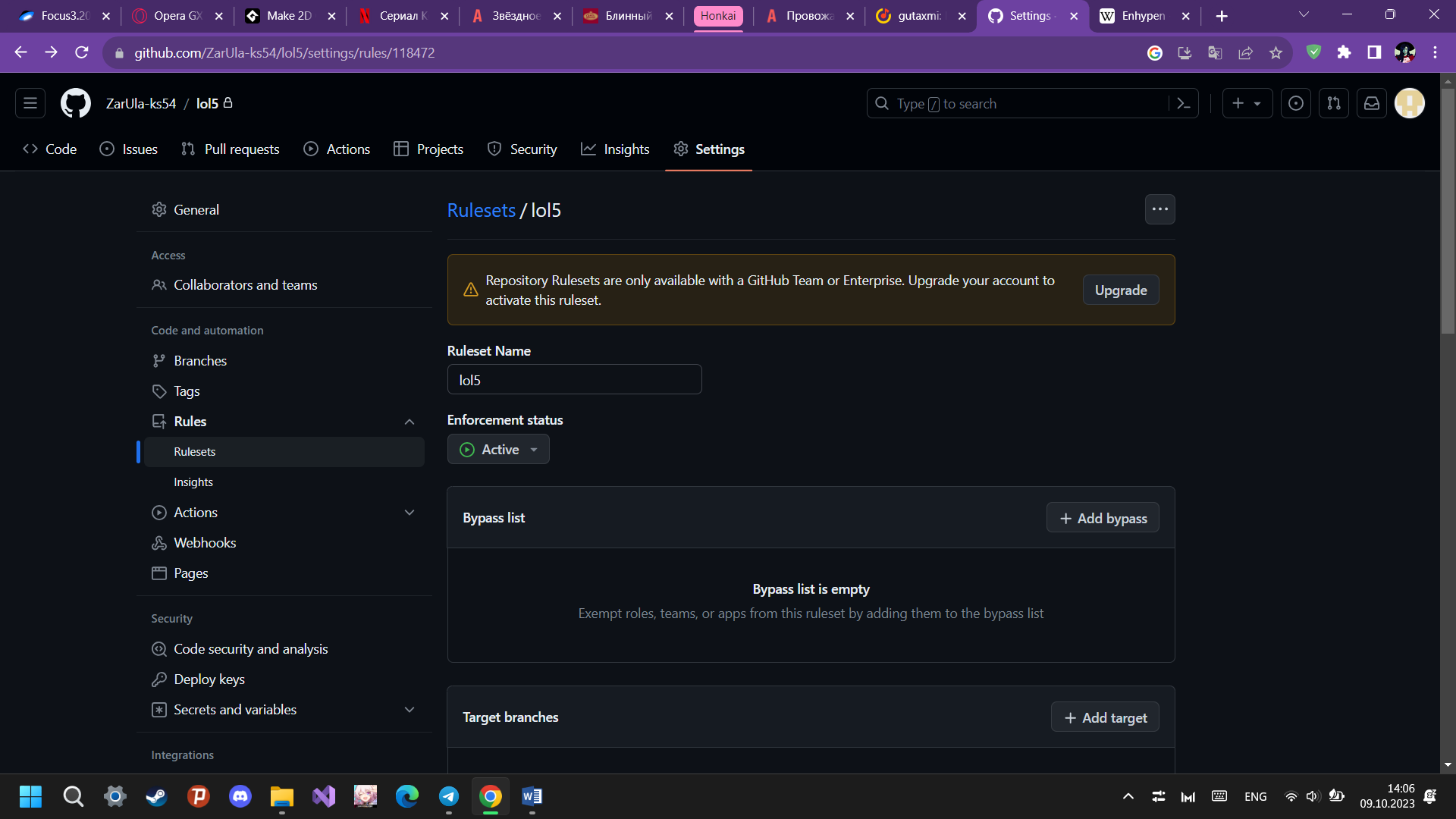


Рис. 9

**Вывод:** Мы получили практические навыки подготовки среды совместной разработки для проведения инспекции программного года.

**Контрольные вопросы:**

1. **Какие действия необходимо выполнить что бы подготовить среду разработки для инспекции кода?** - **-** Установить выбранный инструмент для инспекции кода, например, SonarQube.

- Настроить инструмент, указав путь к исходным файлам кода, конфигурации правил инспекции и другие параметры.

- Запустить инструмент на анализ кода и получить отчет об ошибках, нарушениях стилей программирования и других проблемах.

1. **Какие методы существуют для настройки Github для совместной работы нескольких разработчиков? - -** Создание команды или организации на Github, чтобы объединить всех разработчиков вместе.

**-** Использование веток для отдельных задач или функциональностей, чтобы разработчики могли работать независимо и объединять свои изменения в основную ветку.

**-** Использование запросов на слияние (pull requests), которые позволяют разработчикам обсуждать изменения, проводить код-ревью и вносить комментарии перед объединением изменений.

1. **Что такое организация разработки в сервисе Github? -** способ организации проектов и команд разработчиков на Github. Организация может включать в себя несколько репозиториев, членов команды с разными уровнями доступа и другие настройки, которые облегчают совместную работу и управление проектами.
2. **Какие уровни доступа могут быть заданы для организации в Github? - -** Владелец (Owner): имеет полный доступ к организации и ее репозиториям, включая добавление и удаление членов команды, изменение настроек и т.д.

**-** Администратор (Administrator): имеет полный доступ к организации и ее репозиториям, но без возможности изменения настроек организации.

**-** Наблюдатель (Member): имеет доступ к организации и репозиториям, но без возможности внесения изменений.

**-** Гость (Guest): имеет только чтение доступа к организации и репозиториям.

1. **Какие параметры должны быть заданы при настройках отправки и слияния копии репозитория? -** - Протокол отправки (Push Protocol): выбрать протокол, который будет использоваться для отправки изменений, такие как HTTPS или SSH.

- Ветка отправки (Push Branch): выбрать ветку, которую вы хотите отправить в копию репозитория.

- Удаление истории отправки (Push Force): определить, должна ли новая отправка заменить существующую историю ветки.

- Ветка слияния (Merge Branch): выбрать ветку, с которой вы хотите объединить ветку отправки.

- Стратегия слияния (Merge Strategy): определить стратегию слияния, например, использовать обычное слияние или сжатие изменений.

1. **Какие модели применяются при создании копии репозитария? -** - Fork модель: создается копия репозитория в личном аккаунте пользователя, где разработчик может вносить изменения без прямого доступа к исходному репозиторию. Затем через запрос на слияние (pull request) изменения предлагаются к объединению в исходный репозиторий.

- Clone модель: разработчик создает локальную копию репозитория на своем компьютере. Он может внести изменения и отправить их в основной репозиторий, если у него есть достаточные права доступа.

- Mirror модель: создается точная копия репозитория, которая полностью повторяет исходный репозиторий. Изменения, сделанные в зеркальном репозитории, автоматически отражаются в исходном репозитории.

1. **Что такое «обзор кода»? -** Процесс, когда другой разработчик или команда рассматривает и анализирует код, написанный другим разработчиком. Цель обзора кода - обнаружение ошибок, улучшение качества кода, обмен знаниями и проверка соответствия лучшим практикам программирования. Обзор кода обычно включает комментарии и предложения по улучшению, которые помогают разработчику улучшить его работу.