Федеральное государственное бюджетное образовательное учреждение

высшего образования

РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА и ГОСУДАРСТВЕННОЙ СЛУЖБЫ

при ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ

НИЖЕГОРОДСКИЙ ИНСТИТУТ УПРАВЛЕНИЯ

Факультет управления  
Кафедра информатики и информационных технологий  
Направление (профиль) подготовки: 09.03.03 Прикладная информатика

Отчёт по проделанной лабораторной работе

по предмету «Программная инженерия» № 1

Выполнил студент группы Иб-321

Дубков Максим Олегович

Нижний Новгород

2024

**«Освоение работы в Git (Git GUI), GitHab, GitLab**»

# ***Назначение, возможности, особенности***

**Git** - это система контроля версий, которая используется для управления изменениями в исходном коде программного обеспечения. Она позволяет разработчикам отслеживать изменения в коде, вносить изменения параллельно, объединять их и откатываться к предыдущим версиям. Git также поддерживает распределенную систему контроля версий, что позволяет разработчикам работать локально и синхронизировать изменения с удаленными репозиториями.

**Git GUI** (графический интерфейс пользователя) - это графическая оболочка для Git, облегчающая использование Git для тех, кто предпочитает визуальные инструменты. Некоторые из них включают в себя SourceTree, GitKraken и другие.

**GitHub[[1]](#footnote-1)** - это веб-платформа для хостинга проектов, использующих Git. Разработчики могут загружать свои репозитории на GitHub, делиться кодом, участвовать в проектах с открытым исходным кодом, открывать запросы на слияние (pull requests), откатываться к предыдущим версиям, отслеживать задачи и использовать другие инструменты совместной работы.

**GitLab[[2]](#footnote-2)** - это альтернативная веб-платформа для управления репозиториями Git, аналогичная GitHub. Она предоставляет схожий функционал, включая хостинг репозиториев, систему отслеживания задач, возможность создания запросов на слияние, автоматизацию CI/CD (непрерывной интеграции и развертывания) и другие инструменты для командной разработки.

**Основные особенности Git, GitHub и GitLab включают в себя**:

* Ветвление и слияние (Branching and Merging): Возможность создавать ветки для параллельной разработки и последующего объединения изменений.
* История изменений (Commit History): Возможность отслеживать все изменения в коде и просматривать их историю.
* Откат изменений (Rollback): Возможность откатываться к предыдущим версиям кода.
* Совместная работа (Collaboration): Возможность совместной работы нескольких разработчиков над одним проектом.
* Инструменты совместной разработки (Collaboration Tools): Возможность использования задач, запросов на слияние, комментариев и других инструментов для эффективной командной разработки.
* Непрерывная интеграция и развертывание (CI/CD): Интеграция средств для автоматической проверки кода, сборки проекта и развертывания при необходимости.
* Управление доступом (Access Control): Возможность управления правами доступа к репозиториям и проектам.
* Интеграция с другими инструментами (Integration): Возможность интеграции с различными инструментами разработки и совместной работы.

# ***Сравнение (Git, GitHub vs GitLab)***

**GitHub:**

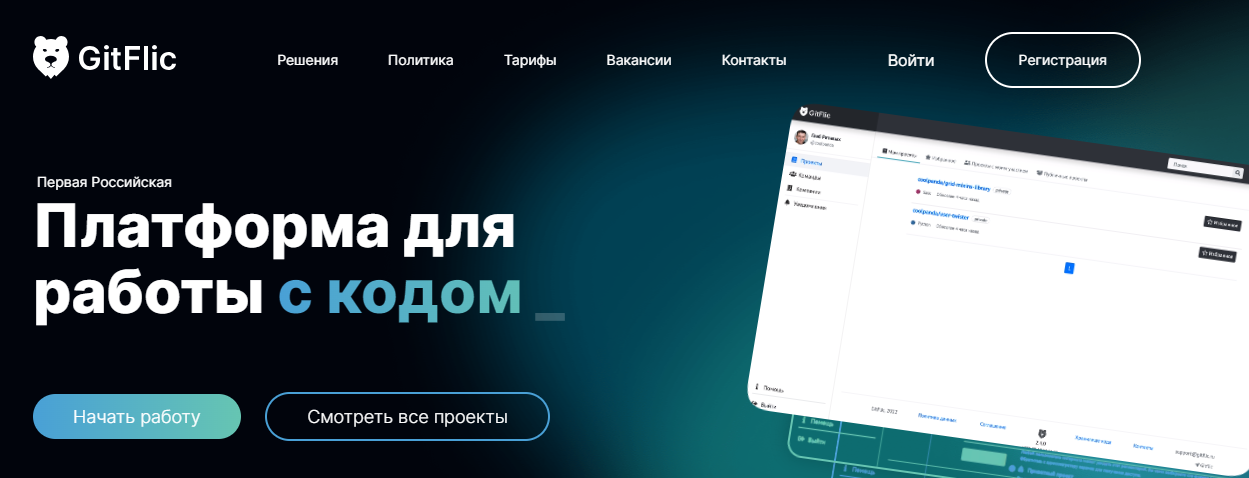
* Назначение: Веб-платформа для хостинга и совместной работы над проектами, использующими Git.
* Возможности:
  + Хостинг репозиториев Git.
  + Задачи и проекты (Issues и Projects).
  + Pull Requests для слияния изменений.
  + Встроенные инструменты совместной разработки.
* Особенности:
  + Крупнейшая платформа для хостинга открытых и закрытых проектов.
  + Интеграция с множеством инструментов и сервисов.
  + Легкость в использовании.

**GitLab:**

* Назначение: Веб-платформа для управления репозиториями Git и совместной разработки.
* Возможности:
  + Хостинг репозиториев Git.
  + Система отслеживания задач (Issues).
  + Запросы на слияние (Merge Requests).
  + Встроенные инструменты CI/CD.
* Особенности:
  + Возможность развертывания GitLab на собственном сервере (self-hosted).
  + Интегрированная система непрерывной интеграции и развертывания.
  + Управление правами доступа и секретностью.

Выбор между GitHub и GitLab может зависеть от конкретных потребностей проекта, предпочтений команды разработчиков, а также от того, важна ли самостоятельная установка и настройка платформы (в случае GitLab).

# ***Российская система GitFlick (возможности)***

[[3]](#footnote-3)

Возможности платформы - решение для хранения исходного кода и работы с ним. Объединяйтесь в команды или работайте в одиночку.

**Создание**

Использовать все возможности Git-инструмента. Создавать, смотреть и управлять проектами с помощью системы контроля версий. Объединяться в команды и компании, масштабировать их и управлять проектами на всех этапах процесса разработки.

**Тестирование**

Проводитть автоматическое тестирование исходного кода. Настраивайть и запускайть необходимые кастомизированные тесты, условия или вспомогательные действия с помощью технологии непрерывной интеграции (Continious Integration).

**Безопасность**

Анализировать код. Улучшаеть качество кода, применяйть статический анализатор для поиска ошибок и выявлять проблемы безопасности на ранних этапах разработки (SAST).

**Выпуск**

Автоматизировать процесс сборки исходного кода. Проводить автоматическую сборку и подготовку релизов для дальнейшего развертывания и интеграций с помощью технологии непрерывной доставки (Continious delivery). Быстрая и безопасная система доставки артефактов на сервер для оптимизации процессов разработки и развертывания ПО.

**Контроль**

Контролировать процесс на всех этапах разработки для улучшения ключевых показателей. Собирать статистику, снижать риски и проводите анализ пользовательского опыта. Собирайтть обратную связь при помощи инструмента “Проблемы”, чтобы улучшить свои продукты.

**Реестр пакетов**

Cовместное использование пакетов ПО в качестве зависимостей в других проектах. Cоздавайтть как личные, так и общедоступные реестры для различных менеджеров пакетов.

# ***Команды системы Git: названия и назначение***

**Основные команды для работы с Git:**

* git init: Инициализация нового репозитория Git.
* git clone: Клонирование существующего репозитория.
* git add: Добавление изменений в индекс для подготовки к коммиту.
* git commit: Фиксация изменений в репозитории.
* git status: Просмотр текущего состояния рабочего каталога и индекса.
* git log: Просмотр истории коммитов.
* git branch: Просмотр, создание или удаление веток.
* git checkout: Переключение между ветками или восстановление файлов из определенного коммита.
* git merge: Объединение изменений из одной ветки в другую.
* git pull: Получение изменений из удаленного репозитория и объединение их с текущей веткой.
* git push: Загрузка изменений из локального репозитория в удаленный.

# ***Работа с GitHub***

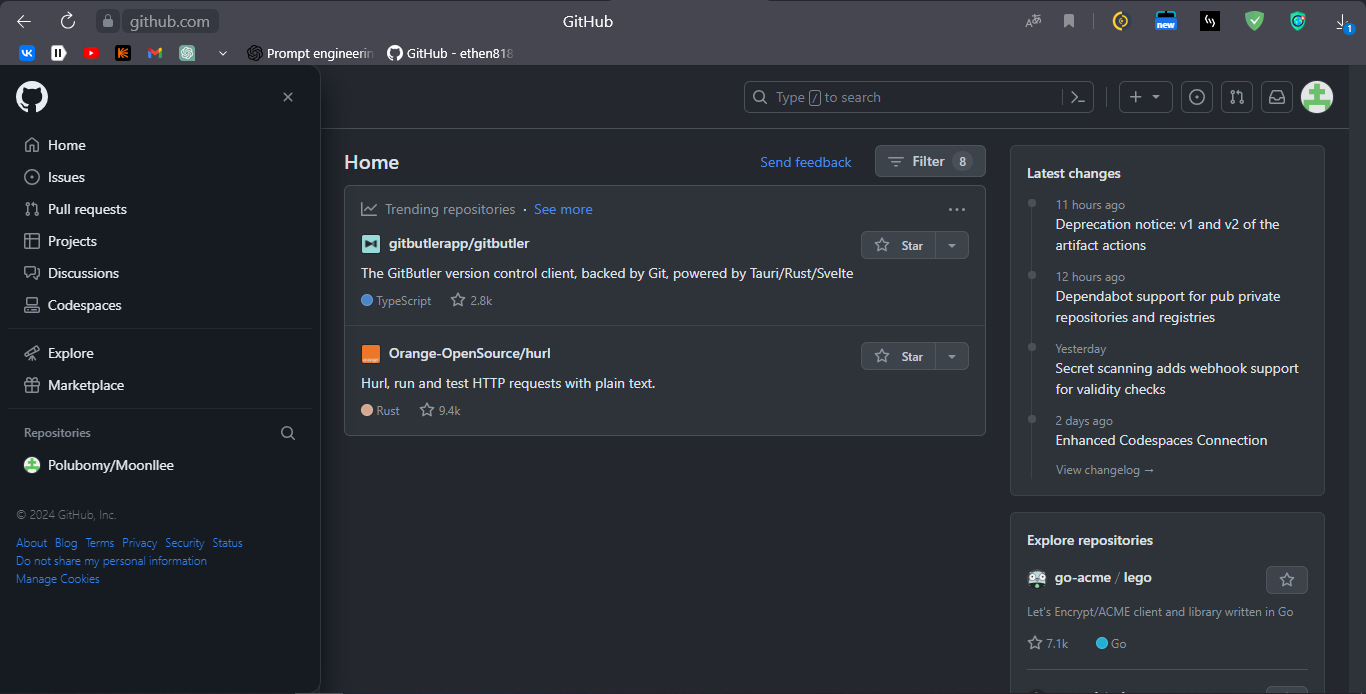


Рисунок 1. Главная страница

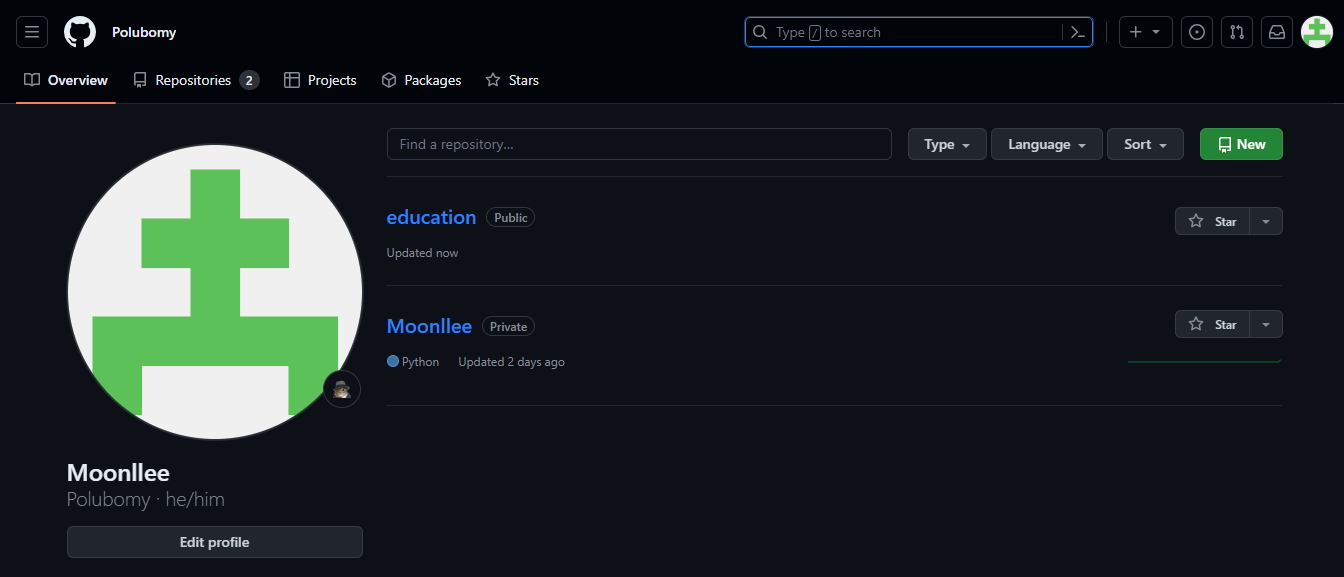


Рисунок 2. Страница репозиториев

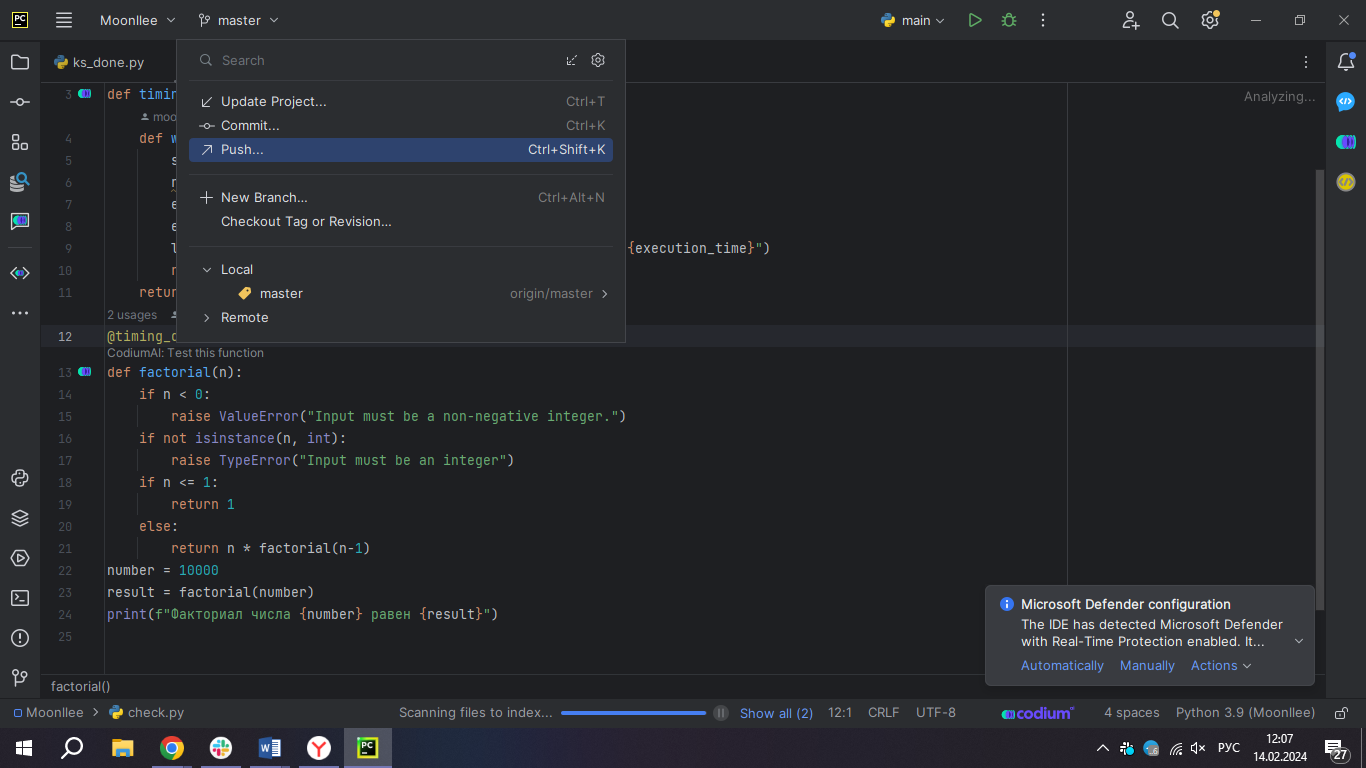


Рисунок 3. Использую плагин для простой отправки проектов в GitHub

1. https://github.com/ [↑](#footnote-ref-1)
2. https://gitlab.com/ [↑](#footnote-ref-2)
3. https://gitflic.ru/ [↑](#footnote-ref-3)