**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА   
И ГОСУДАРСТВЕННОЙ СЛУЖБЫ**

**ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**НИЖЕГОРОДСКИЙ ИНСТИТУТ УПРАВЛЕНИЯ – филиал РАНХиГС**

Факультет: Управление

Кафедра: информатика и информационные технологии

Направление подготовки / специальность: Прикладная информатика (09.03.03)

*(код, наименование)*

Направленность (профиль) / специализация: Корпоративные информационные системы управления

**ПРАКТИЧЕСКАЯ РАБОТА**

|  |  |
| --- | --- |
| по дисциплине: | информатика и программирование |
| на тему: | Освоение работы в Git, GitHub, GitLab, GitVerse. |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
|  |  | **АВТОР** |
|  |  | Обучающийся 2 курса группы Иб-321 |
|  |  | очной формы обучения |
|  |  |  |
|  |  | Захарова Д.О. |
|  |  | *(подпись) (фамилия, инициалы)* |
|  |  |  |
|  |  | **РУКОВОДИТЕЛЬ** |
|  |  | Окулич Виктор Иванович |
|  |  | *(ученая степень, ученое звание)* |
|  |  | оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  | «\_\_\_\_» \_\_ \_\_\_\_\_\_\_\_\_ 2025\_\_ г. |
|  |  | *(дата защиты)* |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  | *(подпись) (фамилия, инициалы)* |

Нижний Новгород, 2025г.

Оглавление

[Цель работы 3](#_Toc190729777)

[Назначение, возможности, особенности 4](#_Toc190729778)

[Сравнение (Git, GitHub vs GitLab) 5](#_Toc190729779)

[Российская система GitFlic, GitVerse 6](#_Toc190729780)

[Команды системы Git: названия и назначение 7](#_Toc190729781)

[Вывод 8](#_Toc190729782)

# Цель работы

Изучить основы работы с системой контроля версий Git, а также познакомиться с различными платформами для хостинга репозиториев, такими как GitHub, GitLab, GitFlic и GitVerse. В ходе работы будут рассмотрены основные команды Git, их назначение и применение, а также сравнительный анализ различных сервисов для управления репозиториями.

# Назначение, возможности, особенности

**Назначение Git**

Git – это распределённая система управления версиями (VCS – Version Control System), созданная Линусом Торвальдсом в 2005 году. Она предназначена для отслеживания изменений в файлах и координации работы нескольких разработчиков над одним проектом.

**Возможности Git**

1. **Отслеживание изменений** – позволяет сохранять каждую версию проекта и при необходимости возвращаться к предыдущим состояниям.
2. **Работа в распределённой среде** – каждый разработчик имеет локальную копию репозитория, что позволяет работать автономно.
3. **Ветвление (Branching)** – возможность создавать отдельные ветки разработки, тестировать новые функции без риска сломать основной код.
4. **Совместная работа** – позволяет нескольким разработчикам работать над одним проектом параллельно.
5. **Слияние изменений (Merge)** – объединение изменений из разных веток без потери данных.
6. **Поддержка различных рабочих процессов** – Git подходит для разных моделей работы, таких как Git Flow, Forking Workflow и другие.
7. **Безопасность** – данные хранятся в зашифрованном виде, обеспечивая целостность информации.

**Особенности Git**

* **Распределённая архитектура** – в отличие от централизованных VCS (например, SVN), каждый пользователь имеет свою полную копию репозитория.
* **Мгновенные коммиты** – фиксирование изменений в истории происходит локально, что ускоряет работу.
* **Хэширование SHA-1** – для защиты данных используется алгоритм SHA-1, который предотвращает подделку истории изменений.
* **Эффективное хранение данных** – Git использует дельта-кодирование, что позволяет экономить место.
* **Высокая скорость работы** – большинство операций выполняются локально без обращения к серверу.

# Сравнение (Git, GitHub vs GitLab)

| **Характеристика** | **Git** | **GitHub** | **GitLab** |
| --- | --- | --- | --- |
| Тип | Система управления версиями | Хостинг Git-репозиториев | Хостинг и DevOps-платформа |
| Основная функция | Контроль версий | Совместная разработка кода | Полноценное управление DevOps |
| Лицензия | Открытый исходный код (GPL) | Закрытый, есть бесплатный тариф | Открытый исходный код (Community Edition) |
| Поддержка CI/CD | Нет встроенного | Через внешние инструменты | Встроенная CI/CD |
| Локальная установка | Работает на любом компьютере | Только облачная версия | Можно развернуть локально |
| Уровень безопасности | Высокий, но зависит от хостинга | Средний (в бесплатных репозиториях код открытый) | Высокий, есть возможности для приватных проектов |
| Интеграция | С различными сервисами | С GitHub Actions, CodeSpaces | С Kubernetes, Prometheus |

**Вывод:**

* **Git** – основа для всех перечисленных платформ.
* **GitHub** – популярный облачный сервис для размещения Git-репозиториев, ориентирован на совместную разработку.
* **GitLab** – мощная альтернатива GitHub с расширенными возможностями DevOps и CI/CD.

# Российская система GitFlic, GitVerse

**GitFlic**

GitFlic – это российский сервис для размещения и управления Git-репозиториями, разработанный как альтернатива GitHub и GitLab.

**Основные возможности:**

* Хостинг публичных и приватных репозиториев.
* Интеграция с российскими облачными сервисами.
* Инструменты для управления проектами.
* Поддержка CI/CD.
* Возможность развертывания на собственных серверах.

**Отличия от GitHub и GitLab:**

* Полностью российская разработка, соответствует требованиям локального законодательства.
* Оптимизирован для работы внутри страны, обеспечивая высокую скорость доступа.
* Меньшая экосистема по сравнению с международными аналогами.

**GitVerse**

GitVerse – это ещё одна российская альтернатива GitHub, ориентированная на безопасное хранение исходного кода в условиях санкций и ограниченного доступа к зарубежным сервисам.

**Ключевые особенности:**

* Локализованный интерфейс.
* Инструменты для работы с кодом и командного взаимодействия.
* Возможность развертывания на серверах компаний.
* Повышенный уровень безопасности данных.

# Команды системы Git: названия и назначение

| **Команда** | **Назначение** |
| --- | --- |
| git init | Инициализация нового репозитория |
| git clone <url> | Клонирование удалённого репозитория |
| git add <файл> | Добавление файлов в индекс (стейджинг) |
| git commit -m "сообщение" | Фиксация изменений с комментарием |
| git status | Проверка состояния репозитория |
| git log | Просмотр истории коммитов |
| git diff | Просмотр изменений в файлах |
| git checkout <ветка> | Переключение на другую ветку |
| git branch | Просмотр списка веток |
| git branch <имя> | Создание новой ветки |
| git merge <ветка> | Объединение изменений из одной ветки в другую |
| git pull | Получение изменений из удалённого репозитория |
| git push | Отправка локальных изменений в удалённый репозиторий |
| git rebase | Перенос изменений из одной ветки в другую |
| git stash | Временное сохранение изменений без коммита |
| git reset | Отмена изменений (разные уровни) |
| git tag | Создание тега для фиксации определённого состояния кода |
| git remote | Управление удалёнными репозиториями |

# Вывод

Git – мощная система контроля версий, необходимая для управления исходным кодом в современных IT-проектах. GitHub, GitLab, GitFlic и GitVerse предоставляют удобные инструменты для работы с Git, позволяя командам эффективно сотрудничать. Знание команд Git – основа успешной работы в программировании и DevOps.