**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра «Математическая Кибернетика и Информационные технологии»

Лабораторная работа №10.

Практическая работа с Git.

Выполнил: Студент группы

БВТ2402

Стадник Андрей

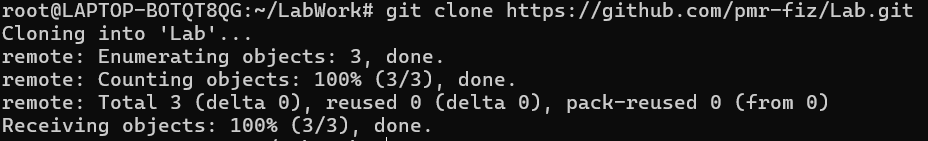
Москва

2025

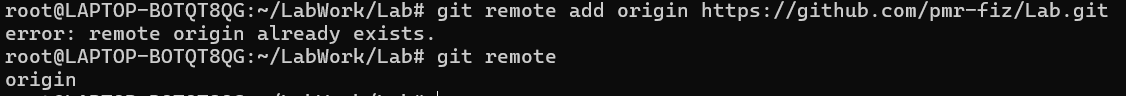
**Цель работы**:

Познакомить студентов с основными операциями работы с удаленными репозиториями в Git, включая клонирование, добавление удаленных репозиториев, отправку изменений и работу с ветками на удаленном репозитории.

**Задание 1: Клонировать удаленный репозиторий**

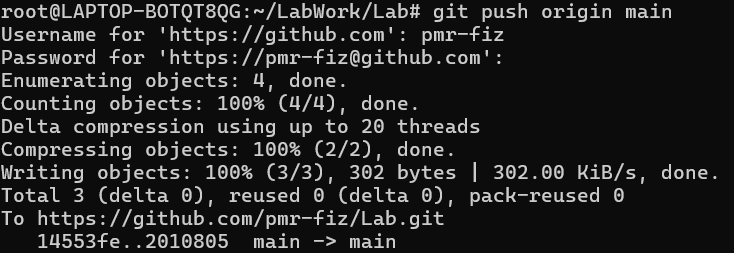
1. Найти удаленный репозиторий на GitHub, GitLab или другой платформе.
2. На локальном компьютере выполнить команду git clone <URL удаленного репозитория>.
3. Убедиться, что репозиторий успешно склонирован на локальный компьютер.

**Задание 2: Добавить удаленный репозиторий**

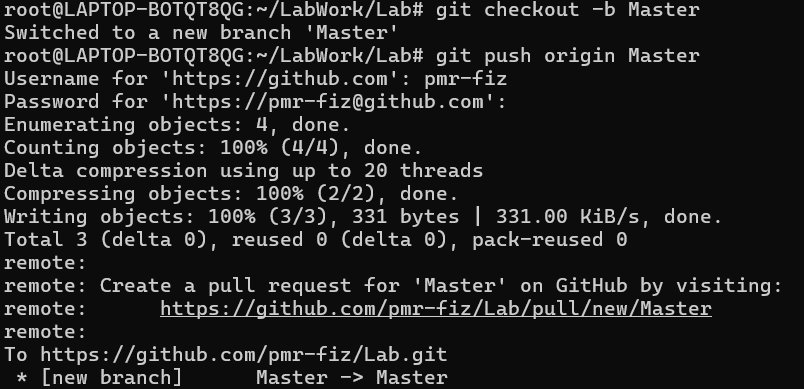
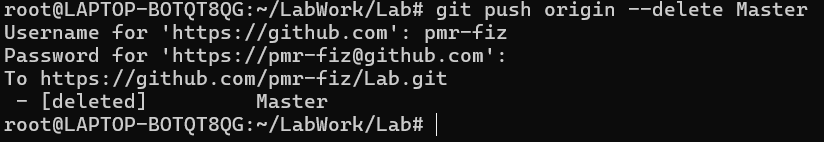
1. В командной строке выполнить команду git remote add origin <URL удаленного репозитория> для добавления удаленного репозитория в локальный.

**Задание 3: Отправить изменения на удаленный репозиторий**

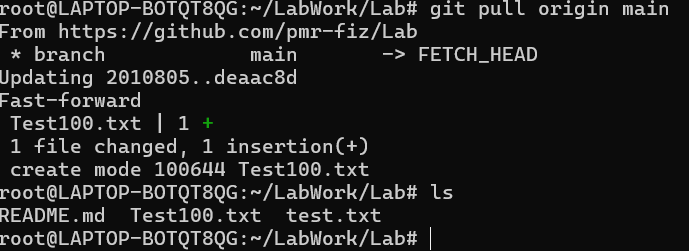
1. Выполнить команду git push origin <branch\_name> для отправки изменений из локального репозитория на удаленный.
2. Проверить, что изменения успешно отправлены на удаленный репозиторий.



**Задание 4: Работа с ветками на удаленном репозитории**

1. Создать новую ветку локально с помощью команды git checkout -b <branch\_name>.
2. Запушить новую ветку на удаленный репозиторий с помощью команды git push origin <branch\_name>.
3. Удалить ветку из удаленного репозитория с помощью команды git push origin --delete <branch\_name>.

**Задание 5: Получение изменений из удаленного репозитория**

1. Выполнить команду git pull origin <branch\_name> для получения изменений из удаленного репозитория на локальный.
2. Убедиться, что изменения успешно применены к локальному репозиторию.

# Вопросы для самопроверки

1.Что такое система контроля версий и для чего она используется?

Система контроля версий (СКВ) — **это инструмент, который сохраняет историю изменений в файлах проекта**. Она позволяет разработчикам отслеживать изменения и управлять версиями кода. [1](https://simpleone.ru/glossary/version-control-system)

**Использование СКВ используется для следующих целей**:

* **Управление изменениями**. СКВ помогает избежать потери данных и позволяет вернуться к любой предыдущей версии проекта. [2](https://sky.pro/wiki/javascript/sistemy-kontrolya-versij-zachem-oni-nuzhny-i-kak-ih-ispolzovat/)
* **Совместная работа**. СКВ помогает синхронизировать работу всех участников, предоставляя каждому доступ к актуальной версии проекта и позволяя интегрировать изменения без конфликтов. [2](https://sky.pro/wiki/javascript/sistemy-kontrolya-versij-zachem-oni-nuzhny-i-kak-ih-ispolzovat/)
* **Безопасность и резервное копирование**. СКВ автоматически создаёт резервные копии всех изменений, что повышает безопасность проекта. В случае ошибки или сбоя системы, можно легко восстановить предыдущие версии файлов. [2](https://sky.pro/wiki/javascript/sistemy-kontrolya-versij-zachem-oni-nuzhny-i-kak-ih-ispolzovat/)
* **Отслеживание ошибок**. С помощью СКВ можно легко отслеживать, какие изменения были внесены в проект и кем. Это помогает быстро находить и исправлять ошибки, а также анализировать причины их возникновения.

2.Какие основные преимущества использования Git перед другими системами контроля версий?

Основные преимущества использования Git перед другими системами контроля версий включают:

* **Независимость и гибкость**. Каждый разработчик имеет локальную копию репозитория, что позволяет работать без постоянного подключения к сети. [3](https://telegra.ph/Kakie-preimushchestva-ispolzovaniya-git-pered-drugimi-sistemami-kontrolya-versij-Git-Pochemu-ehto-korol-kontrolya-versij--i-ego--01-01)
* **Параллельная работа**. Git позволяет создавать ветки, что даёт возможность параллельно разрабатывать новые функции или исправлять ошибки, не затрагивая основную кодовую базу. [3](https://telegra.ph/Kakie-preimushchestva-ispolzovaniya-git-pered-drugimi-sistemami-kontrolya-versij-Git-Pochemu-ehto-korol-kontrolya-versij--i-ego--01-01)
* **Безопасность и надёжность**. Git хранит полную историю изменений, что позволяет легко откатиться к любой предыдущей версии. Это защищает проект от ошибок и позволяет восстановить код в случае проблем. [3](https://telegra.ph/Kakie-preimushchestva-ispolzovaniya-git-pered-drugimi-sistemami-kontrolya-versij-Git-Pochemu-ehto-korol-kontrolya-versij--i-ego--01-01)
* **Эффективное сотрудничество**. Git упрощает процесс слияния изменений от разных разработчиков, а также предоставляет инструменты для разрешения конфликтов, если они возникают. [3](https://telegra.ph/Kakie-preimushchestva-ispolzovaniya-git-pered-drugimi-sistemami-kontrolya-versij-Git-Pochemu-ehto-korol-kontrolya-versij--i-ego--01-01)
* **Скорость**. Git работает очень быстро, особенно при работе с большими проектами. Это связано с тем, что он сохраняет данные в виде снимков состояния, а не в виде списка изменений. [3](https://telegra.ph/Kakie-preimushchestva-ispolzovaniya-git-pered-drugimi-sistemami-kontrolya-versij-Git-Pochemu-ehto-korol-kontrolya-versij--i-ego--01-01)
* **Открытый исходный код**. Git является проектом с открытым исходным кодом, что означает его бесплатное использование и постоянное развитие благодаря вкладу сообщества.

3.Что такое команда "git clone" и как она используется?

Команда **«git clone»** в Git **создаёт копию существующего репозитория в новый каталог**.

Она позволяет **не создавать и инициализировать локальный репозиторий проекта**, а сразу получить его с сервера.

**Формат команды**:

git clone <ссылка на репозиторий> <название папки>

. Можно указать название новой папки, в которую Git скопирует репозиторий. Если его не указать, папка будет называться так же, как и сам репозиторий.

**Примеры использования команды**:

* **Клонирование с мастер-ветки**:

git clone <ссылка до репозитория>

* **Клонирование проекта из репозитория с определённой его ветки**:

git clone -b <имя ветки> <ссылка до репозитория>

. После клонирования в текущем каталоге появится папка с названием проекта.

4.Какая команда используется для добавления файлов в индекс в Git?

Для добавления файлов в индекс в Git используется команда **git add**.

5.Как создать новую ветку в Git и что означает "ветвление" в контексте системы контроля версий?

Чтобы создать новую ветку в Git, нужно использовать команду git branch. Например, для создания ветки с именем testing нужно выполнить команду

git branch testing

. В результате создаётся новый указатель на текущий коммит.

Ветвление в контексте системы контроля версий означает возможность работать над разными версиями проекта. Используя ветвление, можно отклониться от основной линии разработки и продолжать работу независимо от неё, не вмешиваясь в основную линию.

В Git ветвление облегчённое: операция создания ветки выполняется почти мгновенно, переключение между ветками также происходит быстро. Вместо того чтобы копировать файлы из каталога в каталог, Git хранит ветку в виде ссылки на коммит.

6.Что означает команда "git push" и как она используется для отправки изменений на удаленный репозиторий?

Команда **git push** в системе Git **позволяет отправлять локальную ветку на удалённый репозиторий**. Она помогает разработчикам синхронизироваться в команде, а именно отправляет проделанные изменения.

**Использование команды git push может быть необходимо в следующих случаях**:

* **Сохранение резервной копии кода** на сервере на случай потери данных на локальной машине.
* **Отправка изменений членам команды разработки** для синхронизации кодовой базы.

**Перед отправкой изменений** необходимо связать локальный и удалённый репозитории. Для этого используется команда **git remote add <название удалённого репозитория> link**, где вместо **name** указывается имя удалённого репозитория (чаще всего используется **origin**) и **link** — ссылка на удалённый репозиторий (может быть как HTTPS, так и SSH).

**Для отправки изменений** в терминале используется команда **git push origin <имя ветки>**. Вместо **branch** указывается имя ветки, которую надо отправить. Чаще всего используется **master** или **main**.

7. Как можно получить изменения из удаленного репозитория на локальный с помощью Git?

Чтобы получить изменения из удалённого репозитория на локальный с помощью Git, можно использовать команду git fetch. Она извлекает информацию о последних изменениях, внесенных в удалённый репозиторий, и обновляет ветки удалённого отслеживания. Пример использования:

git fetch origin

Ещё один вариант — команда git pull. Она запускает команду git fetch для загрузки содержимого из указанного удалённого репозитория, а затем выполняет команду git merge, объединяющую ссылки и указатели удалённого содержимого в новый локальный коммит слияния. Пример использования:

git pull REMOTE-NAME BRANCH-NAME

Поскольку git pull приводит к слиянию полученных изменений, перед выполнением команды нужно зафиксировать локальную работу.

8. Что такое merge (слияние) в Git и какое его предназначение?

Merge (слияние) в Git — это перенос изменений из одной ветки в другую. При этом сливаемая ветка остаётся в том же состоянии, что позволяет потом продолжить работу с ней.

Предназначение merge заключается в сохранении хронологического порядка и точной истории всех изменений в проекте. Это важно для совместных проектов, где важно понимать контекст изменений.

9. Какое предназначение у команды "git pull"?

Предназначение команды git pull в Git — синхронизация локальной рабочей копии и всех ссылочных объектов с удалённым репозиторием.  [1](https://smartiqa.ru/courses/git/lesson-6)

Эта команда извлекает изменения из удалённого репозитория и автоматически объединяет их с текущей веткой.

10. Какая команда используется для удаления ветки из удаленного репозитория в Git?

Для удаления ветки из удалённого репозитория в Git используется команда git push origin --delete <название ветки> или git push origin :<название ветки>.

Эта команда отправит удаленному репозиторию сигнал на удаление ветки.

**Вывод:** в процессе выполнения лабораторной работы я ознакомился с основными операциями работы с удаленными репозиториями в Git, включая клонирование, добавление удаленных репозиториев, отправку изменений и работу с ветками на удаленном репозитории.