**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 «Информационные технологии»

Лабораторная работа №10

# Практическая работа с Git

Выполнил: Студент группы

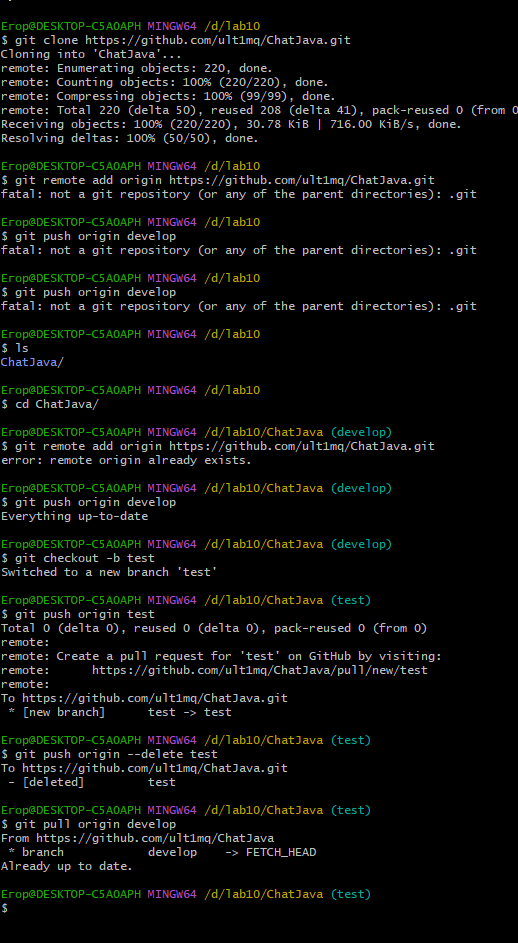
БПИ2401

Сидоренко Егор

Москва

2024

Цель: : познакомить студентов с основными возможностями системы управления версиями Git и научить их выполнять основные операции с помощью этой системы.



Вопросы для самопроверки:

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

Система контроля версий (СКВ) — это инструмент, который помогает отслеживать изменения в файлах и управлять ими. Она используется для хранения истории изменений, совместной работы над проектами, отката к предыдущим версиям и разрешения конфликтов при слиянии изменений от разных разработчиков.

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

- Распределенная архитектура: каждый разработчик имеет полную копию репозитория.

- Высокая производительность и скорость работы.

- Поддержка нелинейной разработки (ветвление и слияние).

- Широкая экосистема инструментов и интеграций.

- Надежность и устойчивость к ошибкам благодаря хешированию данных.

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

Команда `git clone` создает локальную копию удаленного репозитория. Используется так:

```bash

git clone <URL-репозитория>

```

Например:

```bash

git clone https://github.com/user/repo.git

```

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

Команда `git add` добавляет файлы в индекс (staging area). Например:

```bash

git add <имя\_файла>

```

Или для добавления всех изменений:

```bash

git add .

```

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

Новая ветка создается командой:

```bash

git branch <имя\_ветки>

```

Чтобы переключиться на новую ветку:

```bash

git checkout <имя\_ветки>

```

Или одной командой:

```bash

git checkout -b <имя\_ветки>

```

\*\*Ветвление\*\* — это процесс создания изолированной линии разработки, которая позволяет работать над разными задачами или версиями проекта параллельно.

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

Команда `git push` отправляет изменения из локального репозитория на удаленный. Используется так:

```bash

git push <имя\_удаленного\_репозитория> <ветка>

```

Например:

```bash

git push origin main

```

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

Для этого используется команда `git pull`. Она загружает изменения из удаленного репозитория и автоматически сливает их с текущей веткой:

```bash

git pull <имя\_удаленного\_репозитория> <ветка>

```

Например:

```bash

git pull origin main

```

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

\*\*Merge\*\* — это процесс объединения изменений из разных веток. Он используется для интеграции изменений, сделанных в одной ветке, в другую. Например, чтобы добавить изменения из ветки `feature` в `main`, выполните:

```bash

git checkout main

git merge feature

```

9. \*\*Какое предназначение у команды "git pull"?\*\*

Команда `git pull` объединяет две операции: `git fetch` (загрузка изменений из удаленного репозитория) и `git merge` (слияние изменений с текущей веткой). Она используется для обновления локальной ветки до актуального состояния удаленного репозитория.

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

Для удаления ветки из удаленного репозитория используется команда:

```bash

git push <имя\_удаленного\_репозитория> --delete <имя\_ветки>

```

Например:

```bash

git push origin --delete feature-branch

```