**1. Основы системы контроля версий**

* Что такое система контроля версий?
* Зачем нужен Git?
* Основные концепции: репозиторий, коммиты, ветки.

**2. Установка и настройка Git**

* Установка Git на вашу операционную систему.
* Настройка имени пользователя и электронной почты.
* Генерация SSH-ключей для работы с удаленными репозиториями.

**3. Основные команды Git**

* git init – инициализация нового локального репозитория.
* git clone – клонирование существующего репозитория.
* git add – добавление файлов в индекс.
* git commit – создание коммита.
* git status – проверка статуса изменений в рабочем дереве.
* git log – просмотр истории коммитов.
* git diff – сравнение изменений между разными версиями файла.

**4. Работа с ветками**

* Создание новой ветки (git branch).
* Переключение между ветками (git checkout).
* Слияние веток (git merge).
* Решение конфликтов при слиянии.
* Удаление веток (git branch -d).

**5. Удалённые репозитории**

* Работа с удалёнными репозиториями (например, на GitHub, GitLab).
* Отправка изменений в удалённый репозиторий (git push).
* Получение изменений из удалённого репозитория (git pull, git fetch).
* Взаимодействие с другими разработчиками через pull requests.

**6. Теги и версии**

* Создание тегов (git tag).
* Просмотр информации о тегах.
* Управление версиями проекта с помощью тегов.

**7. Откат изменений**

* Отмена изменений до последнего коммита (git reset).
* Возврат к предыдущим состояниям (git revert).
* Использование git stash для временного сохранения незаконченных изменений.

**8. Реализация сложных сценариев**

* Пересмотр истории коммитов (git rebase).
* Интерактивная правка истории (git rebase -i).
* Перемещение файлов и директорий (git mv).
* Игнорирование определённых файлов (.gitignore).

**9. Совместная работа и управление проектами**

* Разделение обязанностей среди разработчиков.
* Практики ветвления (feature branches, release branches).
* Рабочее взаимодействие через pull/merge requests.
* Code review и обсуждение изменений.

**10. Дополнительные инструменты и интеграции**

* Графические интерфейсы для работы с Git (Sourcetree, GitKraken).
* Интеграция с CI/CD системами (Jenkins, Travis CI).
* Автоматическое деплоймент приложений после изменения кода.