**Контрольные вопросы**

**Системы контроля версий GIT**

1. **Назначение систем CVS** CVS (Concurrent Versions System) — это система контроля версий, предназначенная для отслеживания изменений в исходном коде и других файлах, позволяя нескольким разработчикам одновременно работать над проектом. Основные функции CVS включают управление версиями, возможность отката к предыдущим версиям, совместную работу над проектами и ведение истории изменений.
2. **Виды CVS**
   * **CVSNT:** Расширенная версия, добавляющая поддержку новых платформ и протоколов.
   * **TortoiseCVS:** Графический интерфейс для работы с CVS, упрощающий использование командной строки.
   * **AnkhSVN:** Интеграция SVN и визуальных студий, служащая альтернативой CVS. Каждая из этих систем имеет свои особенности, такие как интерфейсы, расширенные функции или конечные платформы.
3. **Особенности GIT** Git — это распределенная система контроля версий, которая позволяет каждому разработчику иметь полную копию репозитория на своем устройстве. Особенности работы с данными в Git включают:
   * **Локальные репозитории:** Каждый разработчик работает с полной копией репозитория, что уменьшает зависимость от центрального сервера.
   * **Снимки (Snapshot):** Git сохраняет состояние проекта в виде снимков (или коммитов), что облегчает возврат к ранее сохраненным версиям.
   * **Управление ветвлением:** Git упрощает создание и слияние веток, что позволяет разработчикам экспериментировать и параллельно разрабатывать новые функции. Преимущества Git также включают скорость, возможность оффлайн-работы и высокую эффективность при работе с большими проектами.
4. **Установка и настройка GIT на ПК**
   * Скачайте установочный файл Git с официального сайта (git-scm.com).
   * Запустите установщик и следуйте инструкциям: выберите параметры установки, такие как поставщик командной строки и редактор по умолчанию.
   * После завершения установки откройте терминал или командную строку.
   * Настройте имя пользователя и адрес электронной почты с помощью команд:

git config --global user.name "Ваше Имя"

git config --global user.email "ваш\_имейл@example.com"

* + (Опционально) Установите дополнительные параметры, такие как цветовую схему для вывода команд и другие предпочтения.

**GitHub**

1. **Назначение сервиса GitHub** GitHub — это веб-сервис для хостинга Git-репозиториев, предоставляющий возможность совместной работы, управления проектами и хранения кода в облаке. GitHub включает инструменты для отслеживания ошибок, обсуждения проектов, управления задачами и автоматизации рабочего процесса.
2. **Назначение сервиса Github Desktop** GitHub Desktop — это настольное приложение, предоставляющее графический интерфейс для работы с Git и GitHub. Оно упрощает процесс управления репозиториями, позволяя пользователям выполнять такие действия, как коммиты, слияния, назначение пул-реквестов и синхронизация изменений без необходимости использовать командную строку.
3. **Определение терминов**
   1. **Репозиторий:** Место, где хранится весь проект, включая исходный код, историю изменений и метаданные.
   2. **Рабочая область и хранилище:** Рабочая область — это локальная копия файлов проекта, хранилище (или репозиторий) — это их версия в Git.
   3. **Форк:** Копия другого репозитория, созданная для внесения изменений или экспериментов, после чего можно предложить изменения оригинальному репозиторию.
   4. **Ветка:** Отдельная линия разработки в репозитории, позволяющая вносить изменения независимо от основной ветки.
   5. **Мастер:** Основная ветка в репозитории по умолчанию (часто называется "main" или "master").
   6. **Коммит:** Сохранение изменений в репозитории с добавлением сообщения, описывающего сделанные изменения.
   7. **Пул (Pull):** Операция, позволяющая забрать изменения из удалённого репозитория и интегрировать их в локальный.
   8. **Пуш (Push):** Отправка локальных изменений в удалённый репозиторий.
   9. **Пул-реквест (Pull Request):** Запрос на внесение изменений из одной ветки в другую, обычно используется для обсуждения и ревью изменений перед их интеграцией.
   10. **Мердж (Merge):** Интеграция (слияние) изменений из одной ветки в другую.
   11. **Кодревью (Code Review):** Процесс проверки и обсуждения изменений в коде другими разработчиками перед их интеграцией в основной код.