**Контрольные вопросы по СКВ**

**Вариант 2**

*Ответить на вопросы, редактируя данный документ*

*Создать репозиторий в папке с данным документом*

*Опубликовать его в удаленном репозитории (ваш собственный public репозиторий на github.com)*

*Ссылку на этот репозиторий прикрепить к заданию в качестве ответа*

1. Объясните, что такое «стадия индексации» (staging area) в Git и как работает команда git add

**Стадия индексации:** Это как “черновик” для вашего коммита. Вы не коммитите изменения напрямую, сначала вы их добавляете сюда.

**git add:** Команда, которая перемещает изменения из вашего рабочего каталога в стадию индексации. Вы можете добавлять отдельные файлы, все измененные файлы или только обновленные, в зависимости от ваших потребностей.

1. Что такое конфликт слияния (merge conflict) в Git и как его можно разрешить?

**Конфликт:** Возникает при попытке объединить ветки, когда Git обнаруживает противоречивые изменения в одних и тех же строках одного и того же файла.

**Разрешение:** Git помечает конфликтные места в файле, вы вручную решаете, какие изменения сохранить, удаляете маркеры и коммитите.

Это ситуация, когда Git не может принять решение за вас.

1. Что такое команда git clone и как она используется?

**Копирование репозитория:** Скачивает весь удаленный репозиторий, включая его историю, к вам на компьютер.

**Начало работы:** Это обычно первая команда, которую вы используете, чтобы начать работать с проектом, размещенным в Git.

Это как “загрузить” исходный код.

1. Что такое git push и как его использовать?

**Отправка изменений:** Загружает ваши локальные коммиты (изменения, зафиксированные у вас) на удаленный репозиторий (например, на GitHub).

**Синхронизация:** Это способ поделиться своими изменениями с другими.

1. Как работает команда git pull и чем она отличается от git fetch?

**git pull:**

**Скачивает** изменения из удаленного репозитория.

**Сразу же сливает** эти изменения в вашу текущую локальную ветку.

Комбинация fetch и merge, более “автоматическая” команда.

**git fetch:**

**Скачивает** изменения из удаленного репозитория.

**Не сливает** их в вашу локальную ветку. Вы можете посмотреть, что изменилось, прежде чем объединять.

Это как “загрузить” изменения, но не применять их автоматически.

Оставляет вам больше контроля над тем, как и когда интегрировать изменения.