<https://github.com/07Rinat07>

**1. Что такое система управления версиями?**

Система управления версиями (VCS — Version Control System) — это инструмент, позволяющий:

* отслеживать изменения в файлах проекта,
* хранить историю изменений,
* возвращаться к предыдущим версиям,
* работать в команде, объединяя изменения нескольких разработчиков.

Она обеспечивает безопасность кода, контроль над изменениями и совместную работу над проектом.

**2. В чем различия между Git и Subversion?**

* **Subversion (SVN):** централизованная система.
  + Вся история хранится на одном сервере.
  + Для работы нужен постоянный доступ к серверу.
  + Один главный репозиторий.
* **Git:** распределённая система.
  + Каждый разработчик имеет полный клон репозитория.
  + Можно работать офлайн, а потом синхронизироваться.
  + Репозиториев может быть несколько (например, форки).

**3. Что такое репозиторий и рабочая копия?**

* **Репозиторий** — хранилище проекта и всех его изменений (истории коммитов, ветвей, тегов). Может быть локальным или удалённым (например, на GitHub).
* **Рабочая копия** — это локальная папка на компьютере, в которой разработчик редактирует файлы.

**4. Для чего используются коммиты?**

**Коммит (commit)** — это фиксация изменений в проекте:

* сохраняет текущее состояние файлов,
* добавляет комментарий к изменению,
* формирует точку в истории проекта, к которой можно вернуться.

**5. Что такое ветвь в Git?**

**Ветвь (branch)** — это отдельная линия разработки проекта.

* Ветка позволяет работать над новой функцией или исправлением, не затрагивая основную версию проекта.
* Основная ветка по умолчанию — master (или main).
* После завершения работы изменения можно объединить (слияние, merge).

**6. Как решаются конфликты при слиянии ветвей?**

**Конфликт возникает**, если в одной и той же строке файла разные ветки содержат разные изменения.

Решение:

1. Git помечает конфликтные строки специальными символами.
2. Разработчик вручную выбирает, какое изменение оставить (или объединяет оба варианта).
3. После исправления файл снова добавляется в коммит.

**7. Как синхронизировать локальный репозиторий с GitHub?**

Для синхронизации используют команды:

* git pull — получить последние изменения с GitHub.
* git push — отправить локальные изменения в GitHub.

В Visual Studio это выполняется через вкладку **Синхронизация**.

**8. Для чего нужен сервис GitHub?**

**GitHub** — это облачный сервис для хранения и совместной работы с Git-репозиториями. Он позволяет:

* хранить код онлайн,
* работать в команде,
* использовать систему задач (Issues),
* проводить код-ревью (Pull Requests),
* использовать CI/CD для автоматизации.

**9. Какие команды Git используются чаще всего?**

* git init — создать новый репозиторий.
* git clone <url> — клонировать репозиторий.
* git add <file> — подготовить файлы к коммиту.
* git commit -m "сообщение" — зафиксировать изменения.
* git status — проверить состояние репозитория.
* git log — история коммитов.
* git branch — список ветвей.
* git checkout <branch> — переключение между ветками.
* git merge <branch> — объединить ветки.
* git pull — получить изменения из удалённого репозитория.
* git push — отправить изменения в удалённый репозиторий.

**10. Как удалить локальный и удаленный репозиторий?**

* **Локальный репозиторий**: достаточно удалить папку проекта (rm -rf project\_name).
* **Удалённый репозиторий (GitHub):**
  1. Зайти в настройки (Settings) репозитория.
  2. В разделе **Danger Zone** выбрать **Delete this repository**.
  3. Подтвердить удаление.