???????

- Внесение исправлений

???????? ????????????

Некоторые команды в Git основываются на подходе к рассмотрению коммитов в терминах внесённых ими изменений, т. е. рассматривают историю коммитов как цепочку патчей. Ниже перечислены эти команды.

git cherry-pick

Команда `git cherry-pick` берёт изменения, вносимые одним коммитом, и пытается повторно применить их в виде нового коммита в текущей ветке. Эта возможность полезна в ситуации, когда нужно забрать парочку коммитов из другой ветки, а не сливать ветку целиком со всеми внесёнными в неё изменениями.

Этот процесс описан и показан в разделе Схема с перебазированием и отбором главы 5.

git rebase

`git rebase` — это «автоматизированный» `cherry-pick`. Он выполняет ту же работу, но для цепочки коммитов, тем самым как бы перенося ветку на новое место.

Мы в деталях разобрались с механизмом переноса веток в разделе Перебазирование главы 3, включая рассмотрение потенциальных проблем переноса опубликованных веток при совместной работе.

Мы использовали эту команду на практике для разбиения истории на два репозитория в разделе Замена главы 7, наряду с использованием флага `--onto`.

В разделе Rerere главы 7 мы рассмотрели случай возникновения конфликта во время переноса коммитов.

Также мы познакомились с интерактивным вариантом `git rebase`, включающемся с помощью опции `-i`, в разделе Изменение сообщений нескольких коммитов главы 7.

git revert

Команда `git revert` — полная противоположность `git cherry-pick`. Она создаёт новый коммит, который вносит изменения, противоположные указанному коммиту, по существу отменяя его.

Мы использовали её в разделе Отмена коммита главы 7 чтобы отменить коммит слияния (merge commit).