0. (подготовка) Возможная структура репозитория для задания:

|  |  |  |  |
| --- | --- | --- | --- |
| Имя ветки | Родитель | Доп-но | Назначение |
| master | - | сделать минимум 2 коммита\* до ответвления | сюда попадает только код, который уйдет в продакшен |
| develop | master | сделать минимум 2 коммита, например:  - добавить новый файл с кодом  - изменить этот файл. | здесь ведется разработка между релизами |
| develop-feature1 | develop | min 2 more commit, one in file from develop | ветка для отдельной крупной фичи |
| develop-feature2 | develop | min 2 more commit, one conflicting with develop-feature1 in file from develop | ветка для отдельной крупной фичи |

\* коммит-сообщение должно быть в формате:

%номер таска%: %название таска% (%дополнительные пояснения%)

например:

1234: Исключение при добавлении пользователя (исправлен sql на сервере)

Для практики, желательно попробовать:

1. Слить изменения из develop-feature1 в develop

**git checkout develop** -переключаемся в ветку develop

**git merge develop-feature1** - сливаеи ветку develop-feature1 в текущую

1. Слить изменения develop в master и пометить полученную ревизию master тегом release1

**git checkout master** -переключаемся в ветку master

**git rebase develop -** ребазируем в текущую сюда ветку develop

**git tag -a release1 -m 'release1' -** отмечаем полученную ревизию release1

1. Удалить develop-feature2 локально и с сервера

**git push origin :develop-feature2** - удаление с сервера

**git branch -D develop-feature2** - удаление локально

1. ***Работа с историей изменений****.* Написать команды:

1. которая выводит сообщения и автора всех коммитов из ветки develop-feature1, которые сделаны за последние 3 часа и которых нет в родительских ветках (develop, master). Сортировка по дате, первые более новые.

**git checkout develop-feature1** -переключаемся в ветку develop-feature1

**git log --pretty=format:"commit: %s author: %an" --since=3.hours --no-merges --date-order**

- выводим лог с параметрами:

**--pretty=format:"commit: %s author: %an"** - формат вывода результата

**--since=3.hours** - за последние 3 часа

**--no-merges** - там где не было слияний

**--date-order** - сортировать по дате

1. выводит для каждого коммита из веток master, develop, в сообщении которых есть упоминание заданного номера задачи (допустим задачи 231), его сообщение, автора и дату. Сортировка по дате, первые более новые.

**git log --branches=master,develop --grep=231 --pretty=format:"commit: %s author: %an date: %cd" --date-order**

- выводим лог с параметрами:

**--branches=master,develop** - из веток master и develop

**--grep=231** - коммит содержит 231

**--pretty=format:"commit: %s author: %an date: %cd"**  - формат вывода результата

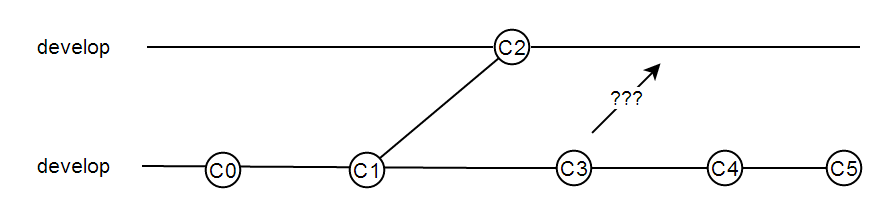
**--date-order** - сортировать по дате

2. ***“Избирательное слияние”***.

Дано: После слияния ветки develop-feature1 в develop, вы продолжили работать в ветке develop-feature1, а develop отдали на тестирование, потом исправили ошибку и закоммитили исправление в develop-feature1, сделали push. Через какое-то время тим лид говорит, что это критический фикс, и надо включить его в develop. Вы делаете pull в ветке develop-feature1 и оказывается, что ваш коллега сделал еще 3 коммита после вашего, и они не стабильные, поэтому не должны быть включены в develop.   
Задание: как добавить в develop только ваш коммит, но чтобы 3 коммита после него в develop не попали?

**git checkout develop** -переключаемся в ветку develop

**git cherry-pick develop-feature1~3** - переносим коммит в develop, игнорируя 3 последние коммита



3. ***“Исправление ошибок”*.**

Дано: Вы начали работу над новой фичей и создали от develop дочернюю ветку develop-feature3.

Сделали там несколько коммитов, и синхронизировались с сервером - все ваши изменения теперь в develop-feature3 на сервере. И тут вы вспомнили, что тим лид строго настрого сказал, что сообщение каждого коммита должно быть форматировано определенным образом (см. примечание п.0) и обязательно содержать номер задачи. А вы об этом забыли и теперь на сервере лежит ветка develop-feature1с “плохими” сообщениями. Задание: Предложите вариант, как наиболее просто исправить ситуацию, чтобы тим лид не увидел ваш промах.

**git checkout develop-feature3** -переключаемся в ветку develop-feature3

**git rebase -i HEAD~2**

получаем в редакторе:

**pick 2b50960 not good commit**

**pick c1aba12 one more not good commit**

В редакторе меняем pick на **edit**, сохраняем, закрываем.

в консоле пишем

**git commit –amend**

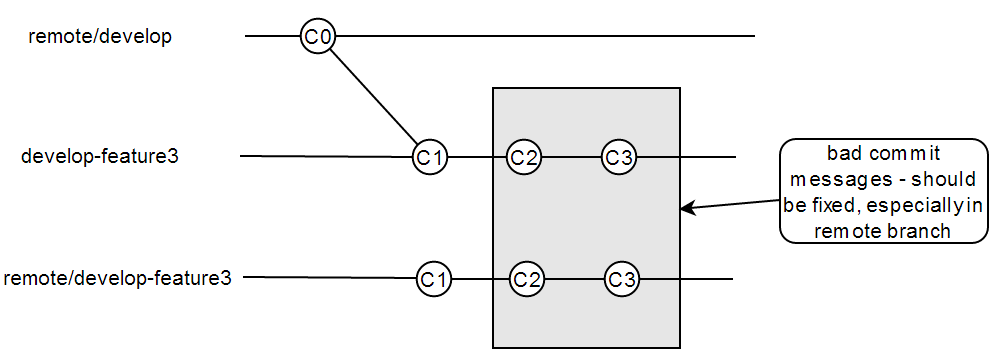
изменяем в редакторе коммит на **1951: now good(not good commit)**

, сохраняем, закрываем

**$ git rebase –continue**

Для второго коммита то же самое

В конце получаем: **Successfully rebased and updated refs/heads/develop-feature3.**



При решении задачи описывайте общую идею, а также приводите shell-комманды с поясняющими комментариями, в том числе и об использованных параметрах.   
Например:

$git log --oneline HEAD --no-merges

вывод в одну строку (сообщения и сокращенного хеша) информации о коммитах из текущей ветки. за исключением коммитов слияния.