

Git

8 октября 2020 г.

1 Как сдавать задания

1. Создаете ветку **hw5-dev**
2. Делаете в pull request в **свой** репозиторий, называете его HW5, Ваше имя и фамилия
3. Добавляете меня (sloboegen98) в ревьюеры.

2 Задачи

1. Напишите в файле task1.txt команды которые делают следующее.
 - (a) Создать локально git репозиторий. Имя на Ваш выбор.
 - (b) Добавьте файлов bio.md и main.cpp. Напишите в файле bio.md любую информацию, а в файле main.cpp — просто программу, которая выводит "Hello!". Закоммитьте изменения.
 - (c) Создайте Makefile, который будет просто собирать и запускать main.cpp (можно даже в одной цели). Соберите и запустите с его помощью main.cpp. Закоммитьте изменения, но сделайте так, чтобы бинарные, объектные файлы не попали в коммит.
 - (d) Сделайте еще три рандомных коммита. Любые на Ваш вкус, только не удаляйте файлы. Можно хоть три раза один пробельный символ добавить.
 - (e) Вернитесь на первый коммит. Создайте от него ветку *smake*. В ней вместо Makefile напишите файл CMakeLists.txt, который делает то же самое, что и наш Makefile.
 - (f) Смерджите ветку smake с веткой master в репозитории.
2. Напишите скрипт, который принимает два хэша коммитов, а затем пишет в файл release-notes.md сообщения всех коммитов, которые находятся между ними. Можно полгать, что пользователь всегда передаст корректные данные.

Hint: `git log -pretty=format...`

3. Существует практика начинать название коммита с номера issue. Пример коммита: 'issue-1 add Makefile'. Будем считать, вам необходимо написать git hook, который будет проверять, что сообщение коммита соответствует следующему шаблону:
'issue-{цифры} text'

То есть если номер issue не добавлен в сообщение коммита, git hook завершится с ошибкой и не даст пользователю закоммитить свои изменения.