

## Организация code review

Code review – просмотр изменений остальными членами команды. Основные цели – поиск багов, ознакомление с изменениями, обсуждение проблем. Цели code review на нашем курсе почти аналогичные, добавляется ещё разнообразие кругозора. Всегда полезно посмотреть на решение той же задачи другим способом.

Немного шуточная статья про code review:

<https://habr.com/ru/post/340550/>

Несколько принципов code review на курсе:

- не рекомендуется делать code review тех домашних работ, которые Вы ещё не выполнили самостоятельно (не контролируется и не наказывает – тут на честность);
- при написании комментариев соблюдайте корректность;
- фидбек должен быть подробным и понятным (не «это неправильно», а «здесь ошибка, потому что ....., предлагаю поправить вот таким способом ....»), желательно избегать двусмысленностей;
- если не знаете, что писать (багов нет, все хорошо), напишите своё мнение о реализации или просто поблагодарите автора за возможность посмотреть на решение задачи под другим углом.

Далее инструкция по подготовке pull-request-ов, через которые и будет проходить code review:

1. Сделайте резервную копию Ваших файлов (на всякий случай).
2. Перейдите в основную ветку репозитория, в которой лежит текущий код.
3. Удалите все файлы, которые нужно показать ревьюеру.
4. Сделайте коммит с удалением файлов (git commit)
5. Выгрузите изменения в удалённый репозиторий (git push).
6. Создайте новую ветку и перейдите в неё (git checkout -b review1)
7. Сделайте revert предыдущего коммита (с удалением файлов):  
git log – нашли хэш последнего коммита  
git revert [хэш] – отменили последний коммит
8. Выгрузите созданную новую ветку в удалённый репозиторий (git push -u origin review1)
9. Зайдите на github под своим логином через браузер.
10. Откройте в своём репозитории раздел "branches" (вкладка "code"). Если видите там новую ветку review1, значит, всё сделали правильно.
11. Справа от новой ветки должна быть кнопка "New pull request". Нажимаете.
12. Убедитесь, что pull request направлен из новой ветки в основную ветку, убедитесь, что в нём фигурируют все нужные изменения, все файлы и прочее.
13. Добавьте ссылку на созданный pull request в общий файл (про файл ниже).

Пример того, что должно получиться в итоге:

<https://github.com/starokurov/otus-cpp/pull/5>

Как организовать:

1. Для начала стоит отметить в общем файле, что будет сигналом к тому, что Вам нужно ревью и Вы готовы сделать ревью в ответ. Ссылка на общий файл:  
[https://docs.google.com/spreadsheets/d/1TpgrmhdVrFvAqxSbNF8wEbnhB5QYxX\\_uR00Wu7qONTc/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1TpgrmhdVrFvAqxSbNF8wEbnhB5QYxX_uR00Wu7qONTc/edit?usp=sharing)
2. Первое, что следует сделать – выбрать лист, соответствующий домашнему заданию, ревью на которое Вы собираетесь сделать (например, ДЗ-02).

3. В колонке «Исполнитель» укажите свои ФИО и добавьте ссылку на репозиторий, который нужно посмотреть.
4. В колонках Ревьюер1 и Ревьюер2 добавьте ссылки на pull request-ы.
5. В Slack, Telegram или любым иным доступным и удобным способом бросьте клич в аудиторию с просьбой выложить код на ревью. Таки образом произойдет взаимовыручка – Вы даете свой код на ревью, кто-то из слушателей выкладывает свой.
6. Взяв на ревью какой-нибудь pull request, в общем документе заполните колонки Ревьюер1 или Ревьюер2, написав в них свои ФИО вместо ???.
7. В «Чат с преподавателем» пришлите ссылки на ревью.