

Выполнил(а) Батманов Даниил Евгеньевич, № группы P3107, оценка _____
Фамилия И.О. студента не заполнять

Название статьи/главы книги/видеолекции Лучшие практики написания комментариев к коду		
ФИО автора статьи (или e-mail) Блок компании VK @AloneCoder https://habr.com/ru/users/AloneCoder/	Дата публикации (не старше 2019 года) 20.07.2021	Размер статьи (от 400 слов) 1601
Прямая полная ссылка на источник или сокращённая ссылка (bit.ly, tr.im и т.п.) https://habr.com/ru/company/vk/blog/568610/		
Теги, ключевые слова или словосочетания Комментарии, код, правило, toDo, проектирование, люди, качество		
Перечень фактов, упомянутых в статье <ol style="list-style-type: none"> 1. У каждой программы две основные аудитории – люди и машины. Обе категории должны правильно понимать написанный код. 2. Машины с лёгкостью понимают синтаксически корректный код, а вот у людей с этим часто случаются сложности, чтобы их избежать – необходимо повышать качество кода, например, одним из способов повышения качества кода являются грамотно написанные комментарии. 3. Чтобы комментарии шли на пользу, стоит придерживаться некоторых правил, таких как: 1) комментарий должен пояснять концепцию, а не дублировать код, так как комментарии, дублирующие код, вносят лишь беспорядок в программу и отнимают время на их написание; 2) не всегда написание комментария может исправить непонятный код, лучше, наоборот, переписать код так, чтобы он стал понятным и без излишних комментариев; 3) грамотно написанный комментарий зачастую должен отвечать на вопрос «зачем?», а не «как?»; 4) пишите комментарии так, чтобы они не вносили путаницу, а наоборот, избавляли от неё; 5) если вы заимствуете часть кода с какого-либо ресурса, тогда оставьте на него ссылку: это очень поможет в обслуживании кода не только другому человеку, но и в вам, в случае недопонимания процессов; 6) не забывайте комментировать, когда исправляете баги в коде, это очень помогает в тестировании; 7) комментарии могут послужить отличным «toDo» для вашего незаконченного участка кода, не забывайте про это. 4. Комментарии в долгосрочной перспективе помогают хорошему коду обслуживаться и улучшаться гораздо быстрее, а если код изначально написан некачественно, то и комментарии его не спасут. 		
Позитивные следствия и/или достоинства описанной в статье технологии (минимум три пункта) <ol style="list-style-type: none"> 1. Повсеместное использование этих правил значительно упростит понимание кода и его обслуживание программистами. 2. Грамотное использование комментариев поможет вам писать код быстрее. 3. Комментарии удобны для планирования общей картины вашей программы. 		
Негативные следствия и/или недостатки описанной в статье технологии (минимум три пункта) <ol style="list-style-type: none"> 1. Написание комментариев, особенно качественных, требует времени и сноровки. 2. Если вы пишете качественные комментарии, то при встрече с некачественными комментариями вам будет сложнее работать. 3. Качественный комментарий является гарантией того, что машина выполняет именно то, что предписывает код, если это не так, то комментарий сбивает с толку. 		
Ваши замечания, пожелания преподавателю или анекдот о программистах¹ <pre>// create a for loop // <-- comment for // start for loop (// round bracket // newline int // type for declaration i // name for declaration = // assignment operator for declaration 0 // start value for i</pre>		

¹ Наличие этой графы не влияет на оценку