|  |  |
| --- | --- |
| **Do code comments suck?**  A conversation about comments.  https://cdn-images-1.medium.com/max/2000/0*Biqyt6ckfgQiJKFM  “Close-up of a person's hands on the keyboard of a MacBook” by [Glenn Carstens-Peters](https://unsplash.com/@glenncarstenspeters?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com/?utm_source=medium&utm_medium=referral)  **What is a comment?**  A comment is a line in your source code file that can be read by developers but is ignored by compilers and interpreters.  **What’s the point of that?**  Code is often hard to understand. Adding descriptive text allows developers to explain their code to those who will maintain the code afterwards.  Need proof? Take the uncommented code below:  This block of code does several things. The result is that it is difficult for programmers to easily understand what it does.  Now consider the same code with comments.  Doesn’t this seem so much easier to understand than before the comments were added? This is because a comments describe whatthe code is doing. whereas the code is describing *how*it is happening.  As developers we like to think in ‘what’s and not ‘how’s.  **So comments help us clarify our code?**  Yes.  **That sounds like a good thing…**  Yes it is. However there are many ways to clarify our ideas in code. Of these ideas, comments are the worst.  **So what makes comments ‘the worst’?**  Well there are a few things. The first is that there is no guarantee that someone will read them. In fact, most developers just don’t bother. Our tooling is even against us. Take a look at this Webstorm screen cap. The default colour scheme makes comments a light easy-to-ignore grey:  https://cdn-images-1.medium.com/max/1600/1*GP8mnKuopErZNCv_zJWEBw.png  Some code with a hidden comment.  **So why not just write a blog called ‘Please Read Comments’?**  That’s a good question. It’s because you can’t trust the comments you read.  The executable code you write is a source of truth. We know it’s true because we are actually running it. Comments are an alternative source of truth but there is nothing to guarantee the truthfulness of them.  Just like any other system where there are multiple sources of truth it is common for a developer to update one source and forget to update another. Can you honestly say that when you change code you read through all the surrounding comments to ensure that they remain correct?  **Well no, it’s an easy thing to forget.**  Exactly. So always make sure your systems have one source of truth.  **Ok I accept, comments aren’t perfect. Nothing in this world is. Are you saying you have a better plan?**  Yes. You show me your comment, and I’ll show you a better alternative.  **What about a comment that describes what a variable is, are you saying that it isn’t good to clarify what your variables are for?**  Clarifying what variables do is super important. That’s why we give them names. If you give your variable a meaningful name then you have no need for a comment. Better still, every place that the well-named variable is used the maintainer of your code is reminded what it’s for. There’s no need to stop reading and refer back to it’s definition. Look how much nicer the second code is.  **This applies to functions too doesn’t it?**  Yes. Give functions and their parameters names that describe what they do. If you do that, there will be no need for a function header style comment.  **So what about your code from earlier, those comments describe blocks of code that achieve functionality together. That can’t be helped by good naming, can it?**  Yes it can. A few lines of code that do one thing together can be extracted into well named functions. Have a look:  See how the code to create an alien is now in a function that describes what it does? Same story with the stats code. The code above needs no further explanation, it does what it says.  Using a function here also has the advantage that details are hidden. The maintainer doesn’t have to worry about *how*the functions workif it’s not relevant to the particular change that they are making.This makes the code easier to understand.  The maintainer also can’t ignore the function name as they might ignore a comment. The function name *is* the code and thus to read the code they must read the function call and the function name.  **So what about if I have to heavily optimise some code, can I use comments then?**  Yes.  **Wait, what?**  We are Software E*ngineers.*Engineers have tools. Comments are tools, so are functions, so are variable names. No tool fixes every problem. No tool is completely redundant. When the code can’t be improved through structuring it in a way that makes it understandable, comments are the best choice.  **So comments are good?**  No, comments suck! Most comments that I see in the wild are inappropriate. In particular comments are used as an excuse for:   * Bad naming of variables or functions. * Excusing functions for being too long or too complex.   However the rule that comments suck is a rule of thumb not a absolute. There are rare uses for comments too.  https://medium.com/@samuel.fare/do-code-comments-suck-abd99103c45a | **Неужели комментировать код – это плохо?**  Поговорим о комментариях.  https://cdn-images-1.medium.com/max/2000/0*Biqyt6ckfgQiJKFM  “Крупный план рук человека на клавиатуре MacBook”, [Glenn Carstens-Peters](https://unsplash.com/@glenncarstenspeters?utm_source=medium&utm_medium=referral) на [Unsplash](https://unsplash.com/?utm_source=medium&utm_medium=referral)  **Что такое комментарий?**  Комментарий – это строка исходного кода, которую могу прочесть разработчики, но не замечают компиляторы и интерпретаторы.  **Какой в нем смысл?**  Как правило, «прочесть» код достаточно трудно. А поясняющий текст помогает разработчикам «объяснить» написанное тому, кто займется поддержкой кода в дальнейшем.  Нужны доказательства? Возьмем вот этот код без комментариев:  Этот блок кода выполняет несколько функций. Результат: программисты не могут быстро понять, что же именно делает данный код.  А теперь снабдим тот же код комментариями.  Разве читабельность кода не возросла? Все потому, что комментарии описывают то, что делает код, в то время, как сам код показывает, *как* это делается.  Мы же, разработчики, привыкли мыслить критериями «что», забывая про «как».  **Так, получается, что комментарии объясняют наш код?**  Да.  **Звучит не плохо …**  Так и есть. Но пояснять код можно разными способами. И комментирование является самым худшим из них.  **Почему же комментирование стало «самым худшим»?**  Причин для этого сразу несколько. Для начала, нет никакой гарантии того, что кто-то прочтет ваши комментарии. На практике, мало кто из разработчиков обращает на них внимание. Посмотрите на этот скрин Webstorm. В изначальной цветовой гамме комментарии выделяются серым и совершенно непримечательным цветом:  https://cdn-images-1.medium.com/max/1600/1*GP8mnKuopErZNCv_zJWEBw.png  Отрывок кода со скрытым комментарием.  **Так почему бы не завести блог с названием «Пожалуйста, читайте комментарии»?**  Хороший вопрос! Потому что нельзя верить комментариям, которые вы читаете.  Вашим источником истины будет служить исполняемый код, который вы сами пишите. Мы знаем, что код истинен, т.к. действительно выполняем его. Комментарии – это альтернативный источник истины, но гарантия истинности как таковой у нас отсутствует.  Как и в любой другой системе с несколькими источниками истины, разработчик может обновить один код, но забыть о другом. Можете ли вы, положа руку на сердце, сказать, что при корректировке кода вы действительно вчитываетесь во все комментарии и следите за их корректностью?  **Конечно же, нет, это часто забывается.**  Вот именно. Так что делайте так, чтобы в ваших системах присутствовал лишь один источник истины.  **Хорошо, признаю, комментарии – это не самое правильное решение. Но ничто не идеально в этом мире. Так вы говорите, что есть вариант получше?**  Да. Покажите мне ваш комментарий, и я предложу вариант получше.  **А как насчет комментариев с описанием переменных? Вы говорите, что лучше не прибегать к ним для создания пояснений по переменным?**  Описание того, что делают переменные, – штука крайне важная. Поэтому мы присваиваем переменным имена. Если назначить переменной осмысленное имя, то необходимость в комментариях отпадет. А еще лучше в каждых местах использования переменной пояснять в коде ее назначение. Тогда мейнтейнеру не придется отрываться от чтения кода для поиска конкретного куска с объяснением переменной. Посмотрите, насколько лучше выглядит второй код.  **Такой принцип применим и к функциям, не так ли?**  Да. Присвойте функциям и параметрам имена на основании их назначения. Как только это сделано, необходимость в комментировании стиля заголовка функции отпадет сразу.  **А что насчет предыдущего кода? Там комментарии описывали блоки кода, совместно обеспечивающие его функциональность? Одно именование нам явно не поможет, разве нет?**  Нет, оно как раз нам поможет. Несколько строк кода с совместным выполнением определенного действия можно вывести в правильно поименованную функцию. Взгляните сюда:  Видите, как этот код выведен в функцию, которая описывает свое назначение? То же распространяется и на статичный код. К нему не требуется дальнейшего пояснения; код делает то, что написано.  Еще один плюс от использования функции в данном примере заключается в том, что все лишние детали скрыты. Мейнтайнеру не придется беспокоиться о том, *как* работают функции, не относящиеся к отдельно взятому изменению в коде. И так код становится удобнее для понимания.  Кроме того, мейнтайнер не проигнорирует имя функции, тогда как про комментарий он может и забыть. Имя функции *уже* является составляющей кода. Поэтому при прочтении всего кода прочтется и имя, и вызов функции.  **А вдруг мне придется хорошенько потрудиться над оптимизацией какого-то кода. Можно ли тогда обратиться к комментариям?**  Да.  **Стоп… что?**  Мы – *разработчики* программных продуктов. И у всех разработчиков есть инструменты. Комментарии – это те же самые инструменты, что и функции с именами переменных. Но ни одно средство не решает всех проблем сразу. И одного инструмента всегда мало. Если изменение структуры кода не улучшает его читабельность, то стоит вспомнить про комментарии.  **Так комментарии – это хорошо?**  Нет, комментарии – отстой! Большая часть комментариев, которые я встречал на практике, – никуда не годится. В основном, комментарии выбирают, чтобы извиниться за:   * неправильное именование переменных или функций. * сложность или длину кода.   Утверждение о том, что комментарии – это плохо, не является истиной в последней инстанции. Скорее, это сугубо практическое наблюдение. Но и для комментариев есть редкие случаи подходящего использования.  https://medium.com/@samuel.fare/do-code-comments-suck-abd99103c45a |