
























Выполнил(а) Степанов Арсений Алексеевич, № группы P3109, оценка _____

Фамилия И.О. студента

не заполнять

Название статьи/главы книги/видеолекции Анатомия юнит тестирования										
ФИО автора статьи (или e-mail) Дмитрий Абузьяров @dvabuzyarov	Дата публикации (не старше 2019 года) 21.06.2020	Размер статьи (от 400 слов) 1247								
Прямая полная ссылка на источник или сокращённая ссылка (bit.ly, tr.im и т.п.) https://habr.com/ru/post/507594/										
Теги, ключевые слова или словосочетания Информатика, Модульное тестирование										
Перечень фактов, упомянутых в статье 1. При написании тестов стоит чётко соблюдать принципы SOLID, это многократно упростит дальнейшую работу с ними и в целом эффективность тестирования. 2. Наследование может быть вредным для тестов, так как для них нужна максимальная наглядность, которая может быть потеряна в сложной иерархии. 3. Вне зависимости от стека технологий проекта и используемых языков, все тесты должны иметь одинаковый подход при их написании. Это позволит переносить удачные практики тестирования на любые части проекта.										
Позитивные следствия и/или достоинства описанной в статье технологии (минимум три пункта) 1. Модульное тестирование – мощный инструмент, упрощающий разработку приложения, а в дальнейшем также его поддержку и рефакторинг, позволяя сразу отловить ошибки в процессе их возникновения. 2. Использование модульного тестирования оправдывает приложенные к его написанию усилия, тесты позволяют разработчикам тратить меньше времени на отладку приложения, что выгодно с экономической стороны вопроса. 3. Тесты довольно чувствительны к архитектуре проекта, поэтому написание тестов может позволить вскрыть имеющиеся проблемы в дизайне и архитектуре проекта.										
Негативные следствия и/или недостатки описанной в статье технологии (минимум три пункта) 1. Для применения технологии необходим определённый экспериментальный опыт команды разработчиков, иначе на написание тестов уйдёт больше времени и эффективность таких тестов будет ниже. 2. Большие затраты сил для громоздких проектов, для покрытия которых требуется написание многочисленных тестов. 3. При неумелом написании тестов или неполном покрытии, тесты не будут помогать отлавливать ошибки, при этом теряется сам смысл модульного тестирования.										
Ваши замечания, пожелания преподавателю или анекдот о программистах¹ <table border="1" style="margin: auto;"> <tr> <td> java java</td> <td> yes user?</td> </tr> <tr> <td> hogging RAM?</td> <td> no user</td> </tr> <tr> <td> telling lies?</td> <td> no user</td> </tr> <tr> <td> open task manager</td> <td></td> </tr> </table>			 java java	 yes user?	 hogging RAM?	 no user	 telling lies?	 no user	 open task manager	
 java java	 yes user?									
 hogging RAM?	 no user									
 telling lies?	 no user									
 open task manager	