

От Антона Архипова

Самая большая моя претензия—это формат книги. Мне кажется, что переключаться между разными языками, русским и английским, довольно утомительно. Гораздо приятней было бы читать данный материал на русском языке, а за неимением полного перевода лучше остановиться на английском варианте.

У читателя может возникнуть ложное чувство, что прочитав вступление на русском языке, он получит достаточно информации. Но это не так. Местами, прочитав только вступление, некоторых читателей может “бомбануть.” Но рано негодовать, если вы не согласны с изложенными идеями, надо обязательно прочитать английский текст, выслушать идеи автора до конца.

Должен сказать, это наиболее практичная книга про ООП, которую мне доводилось встречать. Обычно ООП преподносится как набор шаблонов а-ля Design Patterns от “Gang of Four”, где не говорится абсолютно ничего о том, как мог бы выглядеть хороший код.

Мне очень понравилось, как Егор предлагает нам думать об объектах—уважать их. В этом что-то есть. Я даже задумался, как бы я изменил дизайн многих программ, которые мне довелось писать.

Однако можно найти и изъяны—местами аргументация звучит недостаточно убедительно, чтобы поверить. Например: количество аргументов в конструкторе должно быть не больше четырёх. Почему?? Автор говорит, что больше четырёх аргументов в конструкторе—это слишком много. Для кого много? Для автора? Или для команды, в

которой автор работал? Или для всех людей на Земле цифра 4 обладает каким то магическим свойством? Этот момент слишком субъективен. Например, в книге Clean Code Роберт Мартин говорит, что максимум—это 3. Так почему 4, Егор?

Надо сказать, что чуть позже в книге автор честно признаётся, что у него нет аргументации, почему количество методов у объекта должно быть не более пяти.

Есть и ещё более спорные моменты, за что многие адепты современных тенденций в программировании сожгли бы Егора на костре, если бы были разрешены методы времён инквизиции. Например, сегодня модно считать, что checked исключения—это большая ошибка в Java. Несмотря на это, автор книги вполне бодро объясняет, что эта идея вполне имеет право на жизнь. Почему—прочтите сами!

Несмотря на все спорные моменты, которые по-одиночке могут вызвать у читателя всплеск негодования, надо обратить внимание на то, что эти идеи образуют систему. Систему, в которую автор верит, и убеждён, что следуя принципам, которые изложены здесь, мы сможем писать более поддерживаемый код. Поддерживаемость программ и есть главная цель этой системы. Должен сказать, автор скорее убедил меня в своих идеях, чем нет. Поэтому считаю, что данную книгу можно внести в список рекомендованных к прочтению для специалистов, которые заботятся о своём профессиональном развитии.

@antonarhipov, 3 августа 2016, Эстония