

ГЛАВА 2. ФИЛОСОФИЯ ПРОГРАММИРОВАНИЯ

Ключевые цитаты и их интерпретация

В истории развития языков программирования неоднократно подчеркивалась важность читаемости кода. Как отмечал один из пионеров структурного программирования,

Отладка кода вдвое сложнее, чем его написание. Если вы напишете код настолько умно, насколько сможете, то вы, по определению, недостаточно сообразительны, чтобы его отлаживать. ^[1]

Это высказывание подчеркивает приоритет ясности над изощренностью.

Другой важный аспект — правильная архитектура и избегание излишней сложности. В широко известной книге «The Mythical Man-Month» Фредерик Брукс сформулировал принцип, который часто цитируют в контексте проектирования:

Планируйте выбросить один экземпляр — всё равно придётся. ^[2]

Данная мысль предостерегает от попыток создать идеальную систему с первой попытки и призывает к итеративной разработке.

Особое место занимает вопрос лаконичности. В сообществе Unix бытует известная максима, приписываемая Дугласу Макилрою:

Лучше писать программы, которые помогают писать программы, чем просто программы. ^[3]

Здесь акцент смещается в сторону повторного использования и создания инструментов.

Нельзя обойти стороной и более современные наблюдения. Роберт Мартин (дядя Боб) в своих работах по чистоте кода формулирует правило мальчика-скаута:

Оставляйте код чище, чем вы его застали. ^[4]

Это правило помогает предотвратить энтропию кодовой базы.

Наконец, стоит упомянуть прагматичный подход Эндрю Ханта и Дэвида Томаса, которые ввели понятие «раздражающего запаха» в коде:

Если вы обнаружили ошибку в чужом коде, который используете как библиотеку, — исправьте её и отправьте патч автору. Не ждите, что кто-то другой это сделает. ^[5]

Такая позиция способствует развитию сообщества и качества программногo обеспечения.

Пояснения и библиографические ссылки

- Керниган Б., Пайк Р.** «Практика программирования» — часто цитируемая фраза, отражающая сложность отладки. В оригинале: «Debugging is twice as hard as writing the code in the first place. Therefore, if you write the code as cleverly as possible, you are, by definition, not smart enough to debug it.» (Brian W. Kernighan, Rob Pike).
- Брукс Ф.** «Мифический человеко-месяц» (1975). Глава 11 — «Plan to throw one away; you will, anyhow.». Эта идея лежит в основе создания прототипов и экспериментальных версий.
- Макилрой Д.** (Douglas McIlroy) — один из создателей конвейеров Unix. Цитата из книги «The Art of Unix Programming» (Eric S. Raymond): «Write programs that do one thing and do it well. Write programs to work together. Write programs to handle text streams, because that is a universal interface.» Приведённая фраза — вариация на тему метапрограммирования.
- Мартин Р.** (Uncle Bob) «Clean Code» (2008), глава 1. Правило мальчика-скаута: «Always leave the code you’re editing a little cleaner than you found it.» — мощный приём борьбы с разрастанием технического долга.
- Хант Э., Томас Д.** «Программист-прагматик» (1999). Раздел «Программирование намеренное и ответственное». Авторы призывают не мириться с чужими ошибками: «If you find a bug in someone else’s code that you’re using as a library, fix it and send the patch to the author. Don’t wait for someone else to fix it.» Это развивает инициативность и профессионализм.

** Все цитаты приведены в переводе, сноски соответствуют источникам.*