

ИСКУССТВО ЧИСТОГО КОДА

КАК ИЗБАВИТЬСЯ ОТ СЛОЖНОСТИ
И УПРОСТИТЬ ЖИЗНЬ

КРИСТИАН МАЙЕР



Краткое содержание

Об авторе	12
О научном редакторе	13
От издательства	14
Предисловие	15
Благодарности	17
Введение	19
Глава 1. Сложность — враг продуктивности	26
Глава 2. Принцип 80/20	53
Глава 3. Создание минимально жизнеспособного продукта ...	81
Глава 4. Написание чистого и простого кода	98
Глава 5. Преждевременная оптимизация — корень всех зол	140
Глава 6. Состояние потока	163
Глава 7. «Делай что-то одно, но делай это хорошо» и другие принципы Unix	174
Глава 8. В дизайне лучше меньше, да лучше	210
Глава 9. Фокус	225
От автора	234

Оглавление

Об авторе	12
О научном редакторе	13
От издательства	14
Предисловие	15
Благодарности	17
Введение	19
Для кого эта книга?	22
Чему вы научитесь?	23
Глава 1. Сложность — враг продуктивности	26
Что такое сложность?	30
Сложность жизненного цикла проекта	31
Планирование	32
Определение требований	33
Проектирование	34
Разработка	35
Тестирование	35
Развертывание	38
Сложность в ПО и алгоритмическая теория	38
Сложность в обучении	45
Сложность в процессах	49

Сложность в повседневной жизни: «смерть от тысячи порезов»	50
Заключение	52
Глава 2. Принцип 80/20	53
Основы принципа 80/20	53
Оптимизация прикладного ПО	55
Продуктивность	57
Метрики успеха	60
Фокус и распределение Парето	62
Значение принципа 80/20 для разработчиков кода	65
Метрика успеха для программиста	66
Распределение Парето в реальном мире	67
Фрактальная структура распределения Парето	72
Практические советы 80/20	75
Источники	78
Глава 3. Создание минимально жизнеспособного продукта . . . 81	81
Проблемный сценарий	82
Потеря мотивации	84
Рассеянность внимания	84
Нарушение сроков	85
Отсутствие обратной связи	85
Ложные предположения	86
Излишняя сложность	87
Создание MVP	89
Четыре кита в создании MVP	93
Преимущества MVP-подхода	95
Скрытое программирование в сравнении с MVP-подходом . . . 96	96
Заключение	97
Глава 4. Написание чистого и простого кода	98
Зачем писать чистый код?	99

Написание чистого кода: основные принципы	101
Принцип 1. Представляйте общую картину	102
Принцип 2. Встаньте на плечи гигантов	104
Принцип 3. Пишите код для людей, а не для машин	105
Принцип 4. Соблюдайте правила именования	107
Принцип 5. Придерживайтесь стандартов и будьте последовательны	109
Принцип 6. Используйте комментарии	111
Принцип 7. Избегайте ненужных комментариев	114
Принцип 8. Принцип наименьшего удивления	116
Принцип 9. Не повторяйтесь	117
Принцип 10. Принцип единой ответственности	119
Принцип 11. Тестируйте	123
Принцип 12. Малое — прекрасно	125
Принцип 13. Закон Деметры	127
Принцип 14. Вам это никогда не понадобится	133
Принцип 15. Не используйте слишком много уровней отступов	134
Принцип 16. Используйте метрики	136
Принцип 17. Правило бойскаутов и рефакторинг	137
Заключение	139

Глава 5. Преждевременная оптимизация — корень всех зол	140
Шесть типов преждевременной оптимизации	141
Оптимизация функций кода	142
Оптимизация функциональности	142
Оптимизация планирования	143
Оптимизация масштабируемости	143
Оптимизация разработки тестов	144
Оптимизация объектно-ориентированной картины мира	144
Преждевременная оптимизация: пример	145

Шесть советов по настройке производительности	150
Сначала измеряйте, потом улучшайте	151
Принцип Парето — всему голова	152
Алгоритмическая оптимизация приносит успех	155
Да здравствует кэш!	156
Лучше меньше, да лучше	159
Знай меру	161
Заключение	161
Глава 6. Состояние потока	163
Что такое состояние потока?	164
Как достичь состояния потока	166
Четко поставленные цели	166
Механизм обратной связи	167
Баланс между потенциалом и возможностями	168
Советы по достижению состояния потока для программистов	169
Заключение	172
Источники	173
Глава 7. «Делай что-то одно, но делай это хорошо» и другие принципы Unix	174
Расцвет Unix	175
Введение в философию Unix	176
15 полезных принципов Unix	178
1. Пусть каждая функция делает хорошо что-то одно	178
2. Простое лучше сложного	182
3. Малое — прекрасно	183
4. Создавайте прототип как можно быстрее	185
5. Предпочитайте портируемость эффективности	186
6. Храните данные в плоских текстовых файлах	189
7. Используйте эффект рычага в своих интересах	191
8. Отделяйте UI от функциональности	193

9. Делайте каждую программу фильтром	198
10. Чем хуже, тем лучше	200
11. Чистый код лучше умного	201
12. Создавайте программы так, чтобы они могли взаимодействовать с другим ПО	202
13. Делайте свой код робастным	203
14. Исправляйте то, что можете, но лучше, если сбой случится раньше и громко	205
15. Избегайте написания кода вручную: если можете, пишите программы для создания программ	207
Заключение	208
Источники	209
Глава 8. В дизайне лучше меньше, да лучше	210
Минимализм в эволюции мобильных телефонов	211
Минимализм в поисковых системах	212
Стиль Material Design	214
Как достичь минимализма в дизайне	216
Свободное пространство	216
Сокращайте количество элементов дизайна	218
Удаляйте функции	221
Снижайте количество шрифтов и цветов	222
Будьте последовательны	223
Заключение	224
Источники	224
Глава 9. Фокус	225
Оружие против сложности	225
Обобщим все принципы	229
Заключение	232
От автора	234