

ПОВНИЙ ЗМІСТ

Передмова	18
Вступ до другого видання	21
Структура книги	22
Що означає ім'я	23
Початковий код та інші ресурси	24
Надсилайте нам свої відгуки	24
Із вступу до першого видання	25
Для кого ця книга	26
Що означає бути програмістом-прагматиком	26
Окремі прагматики та великі команди	28
Це безперервний процес	29
Глава 1. Філософія прагматизму	31
Тема 1. Це ваше життя	32
Тема 2. Кіт з'їв мій початковий код	33
Довіра у команді	34
Взяття на себе відповідальності	34
Тема 3. Програмна ентропія	36
Перш за все — не нашкодити	38
Тема 4. Суп із каменів та варені жаби	39
Тема 5. Достатньо хороше програмне забезпечення	42
Йдіть на компроміси з користувачами	43
Будьте помірковані	44
Тема 6. Ваш вантаж знань	46
Ваш вантаж знань	46
Створення свого вантажу знань	47
Цілі	48

Можливості для навчання	50
Критичне мислення	50
Тема 7. Спілкуйтесь!	53
Знайте, з ким ви спілкуєтесь	53
Знайте, що вам потрібно сказати	54
Обирайте зручний момент	55
Обирайте стиль спілкування	55
Подавайте свої ідеї у привабливій формі	56
Залучайте тих, з ким спілкуєтесь	56
Вчіться слухати	56
Відповідайте людям	57
Документація	57
Короткі підсумки	58
Глава 2. Прагматичний підхід	61
Тема 8. Сутність якісного проектування	62
Принцип ETC — це цінність, а не правило	63
Тема 9. Пороки дублювання	64
Принцип DRY не лише для кодування	66
Дублювання у початковому коді	66
Дублювання у документації	69
Презентаційне дублювання	72
Дублювання серед розробників	73
Тема 10. Ортогональність	74
Переваги ортогональності	76
Проектування	78
Інструментальні засоби та бібліотеки	79
Кодування	80
Тестування	81
Документація	82
Життя з ортогональністю	83
Тема 11. Зворотність	85
Зворотність	86
Гнучка архітектура	87

8 Повний зміст

Тема 12. Трасуючі кулі	89
Код, що сяє у темряві	90
Трасуючі кулі не завжди влучають у ціль	94
Трасуючий код у порівнянні з прототипуванням	94
Тема 13. Прототипи та пам'ятні нотатки	96
Що підлягає прототипуванню	97
Як користуватися прототипами	97
Прототипування архітектури	98
Як <i>не слід</i> користуватися прототипами	99
Тема 14. Предметно-орієнтовані мови	100
Деякі предметно-орієнтовані мови	101
Характеристики предметно-орієнтованих мов	104
Компромiс між внутрішніми та зовнішніми предметно-орієнтованими мовами	104
Внутрішня предметно-орієнтована мова майже задарма	105
Тема 15. Оцінювання	107
Якої точності оцінки достатньо?	108
Звідки беруться оцінки	109
Оцінювання строків виконання проєктів	111
Що відповісти на прохання що-небудь оцінити	114
Глава 3. Основні інструментальні засоби	115
Тема 16. Сила простого тексту	117
Що таке простий текст?	117
В чому сила простого тексту?	118
Найменший спільний знаменник	120
Тема 17. Ігри у шкаралупки	121
Ваша власна оболонка	123
Тема 18. Ефективне редагування	125
Що означає вільне володіння редактором	125
Прагнення до вільного володіння редактором	126
Тема 19. Контроль версій	128
Каталоги, які спільно використовуються, це <i>не</i> система контролю версій	129

Все стартує з початкового коду	129
Розгалуження	130
Контроль версій як центральний вузол проєкту	132
Тема 20. Налаштування	134
Психологія налаштування програм	134
Мисленева установка на налаштування	135
З чого починати налаштування	136
Стратегії налаштування	137
Програміст у чужій країні	137
Бінарний пошук	139
Метод гумового качення	141
Елемент здивування	143
Контрольний список питань із налаштування	144
Тема 21. Маніпулювання текстом	145
Тема 22. Технічні щоденники	148
Глава 4. Прагматична параноя	151
Тема 23. Проєктування за контрактом	152
Принцип проєктування за контрактом	153
Реалізація проєктування за контрактом	157
Проєктування за контрактом та аварійне завершення	158
Семантичні інваріанти	159
Динамічні контракти та агенти	160
Тема 24. Мертві програми не брешуть	162
Принцип “піймав — відпустив” — лише для ловлі риби	163
Аварійне завершення замість відправки на смітник	163
Тема 25. Стверджувальне програмування	165
Твердження та побічні ефекти	166
Залишайте увімкненим режим тверджень	167
Тема 26. Як збалансувати ресурси	169
Вкладене виділення ресурсів	172
Об’єкти та винятки	173
Баланс винятків	174

10 Повний зміст

Коли не можна збалансувати ресурси	175
Перевірка балансу	176
Тема 27. Не випереджайте світло фар свого автомобілю	177
Чорні лебеді	179
Глава 5. Гнучкість чи ламкість	181
Тема 28. Розв’язування	182
Аварії поїздів	184
Вади глобалізації	188
Наслідкування погіршує зв’язування	189
Вся справа у змінах	189
Тема 29. Маніпулювання реальним світом	190
Події	191
Кінцеві автомати	191
Проектний шаблон “Оглядач”	196
Модель “видавець — підписник”	197
Реактивне програмування, потоки даних та події	198
Події всюдисущі	201
Тема 30. Перетворювальне програмування	202
Виявлення перетворювань	204
В чому ж тут користь	209
А як щодо обробки помилок?	210
Перетворення перетворюють програмування	214
Тема 31. Податок на наслідування	215
Трохи передісторії	216
Складнощі застосування наслідування для спільного використання коду	217
Кращі альтернативи	219
Наслідкування рідко є тим, що вам треба	224
Тема 32. Конфігурування	225
Статична конфігурація	226
Конфігурація як служба	226
Не пишіть морально застарілий код	227

Глава 6. Паралельність	229
Все паралельно	229
Тема 33. Розривання часового зв'язування	231
У пошуках паралельності	231
Можливості для досягнення паралельності	233
Можливості для досягнення паралелізму	234
Виявити можливості простіше за все	234
Тема 34. Спільний стан — неправильний стан	236
Неатомарні оновлення	236
Множинні транзакції ресурсів	240
Оновлення без транзакцій	242
Інші види виняткового доступу	243
Докторе, мені боляче	243
Тема 35. Актори та процеси	244
Актори можуть бути лише паралельними	244
Простий актор	245
Відсутність явної паралельності	250
Erlang готує ґрунт	250
Тема 36. Класні дошки	251
Класна дошка у дії	253
Системи обміну повідомленнями можуть бути подібними до класних дошок	254
Але не все так просто...	255
Глава 7. По ходу кодування	257
Тема 37. Дослухайтеся до своїх інстинктів	259
Страх порожньої сторінки	259
Боротьба з собою	260
Як дослухатися до своїх інстинктів	261
Час грати!	261
Не лише <i>свій</i> код	262
Не лише код	263

12 Повний зміст

Тема 38. Програмування за збігом	264
Як програмувати за збігом	264
Як програмувати помірковано	268
Тема 39. Швидкодія алгоритмів	271
Що мається на увазі під оцінюванням алгоритмів	271
Асимптотичне позначення	272
Розумне оцінювання алгоритмів	274
Швидкодія алгоритму на практиці	275
Тема 40. Рефакторинг	278
Коли слід виконувати рефакторинг	280
Порядок виконання рефакторингу	282
Тема 41. Тестувати, щоб кодувати	284
Обмірковування тестів	285
Кодування на основі тестів	286
Застосовуючи розробку на основі тестування, треба знати, куди йти	288
Повернення до коду	290
Модульне тестування	290
Тестування відповідності до контракту	291
Спеціальне тестування	293
Створення тестового вікна	293
Культура тестування	294
Тема 42. Тестування на основі властивостей	296
Контракти, інваріанти та властивості	296
Генерування тестових даних	298
Виявлення невдалих припущень	299
Тести на основі властивостей здатні дивувати	302
Тести на основі властивостей допомагають проєктувати	303
Тема 43. Будьте обережні	304
Інші 90%	304
Основні принципи захисту	305
Здоровий глузд та криптографія	310
Тема 44. Іменування	312
Повага до культури	315

Узгодженість	316
Переименовувати ще складніше	317
Глава 8. До початку проєкту	319
Тема 45. Пастка вимог	320
Міф про вимоги	320
Програмування як терапія	321
Вимоги — це процес	323
Поставте себе на місце клієнта	324
Вимоги та правила	325
Вимоги та реальність	325
Документування вимог	326
Надто докладна специфікація	328
“Ще один м’ятний пластивець...”	328
Ведення словника термінів проєкту	328
Тема 46. Рішення нерозв’язних головоломок	330
Ступені свободи	331
Не закривайте шлях самому собі!	332
Доля сприяє підготовленому розуму	333
Тема 47. Спільна робота	334
Парне програмування	336
Групове програмування	337
Що слід робити?	337
Тема 48. Сутність гнучкості	338
Гнучкий процес взагалі неможливий	340
Що ж тоді робити?	340
І це рухає проєкт	342
Глава 9. Прагматичні проєкти	343
Тема 49. Прагматичні команди	344
Жодних розбитих вікон	345
Зварені жаби	345
Планування поповнення вантажу знань	346

14 Повний зміст

Зовнішнє спілкування команди	347
Не повторюйтесь	348
Трасуючі кулі у команді	349
Автоматизація	350
Знайте, коли зупинитися	350
Тема 50. Кокосами не обійтись	351
Вся справа у контексті	352
Один і той самий підхід годиться не для всіх	353
Головна мета	354
Тема 51. Початковий набір інструментальних засобів програміста-прагматика	356
Ведення проекту шляхом контролю версій	357
Суворе та безперервне тестування	358
Затягування сітки	362
Повна автоматизація	362
Тема 52. Приносьте задоволення своїм користувачам	364
Тема 53. Гордість та упередження	366
Додаток А. Післямова	369
Моральний орієнтир	370
Уявляйте майбутнє таким, яким ви хочете його бачити	371
Додаток Б. Бібліографія	373
Додаток В. Можливі відповіді до вправ	375