



Гибкая разработка программ на Java и C++

принципы, паттерны и методики

Роберт С. Мартин
при участии Джеймса Ньюкирка и Роберта Косса

Agile Software Development

Principles, Patterns, and Practices

Robert Cecil Martin
with contributions by
James W. Newkirk and Robert S. Koss



Pearson Education, Inc.
Upper Saddle River, New Jersey 07458

Гибкая разработка программ на Java и C++

принципы, паттерны и методики

Роберт С. Мартин
при участии
Джеймса Ньюкирка и Роберта Косса



Компьютерное издательство “Диалектика”
Москва • Санкт-Петербург • Киев
2017

ББК 32.973.26-018.2.75

М29

УДК 681.3.07

Компьютерное издательство “Диалектика”

Зав. редакцией С.Н. Тригуб

Перевод с английского Ю.Н. Артеменко

Под редакцией Ю.Н. Артеменко

По общим вопросам обращайтесь в издательство “Диалектика” по адресу:
info@dialektika.com, <http://www.dialektika.com>

Мартин, Роберт С.

М29 Гибкая разработка программ на Java и C++: принципы, паттерны и методики.

Пер. с англ. — СПб. : ООО “Альфа-книга”, 2017. — 704 с. : ил. — Парал. тит. англ.

ISBN 978-5-9908462-8-9 (рус.)

ББК 32.973.26-018.2.75

Все названия программных продуктов являются зарегистрированными торговыми марками соответствующих фирм.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотокопирование и запись на магнитный носитель, если на это нет письменного разрешения издательства Prentice Hall, Inc.

Authorized translation from the English language edition published by Prentice Hall, Ptr., Copyright © 2003 by Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Russian language edition is published by Dialektika Computer Books Publishing according to the Agreement with R&I Enterprises International, Copyright © 2017.

Книга напечатана согласно договору с ООО “ПРИМСНАБ”.

Научно-популярное издание

Роберт С. Мартин

Гибкая разработка программ на Java и C++: принципы, паттерны и методики

Верстка Т.Н. Артеменко

Художественный редактор В.Г. Павлютин

Подписано в печать 15.11.2016. Формат 70×100/16.

Гарнитура Times.

Усл. печ. л. 56,76. Уч.-изд. л. 36,1.

Тираж 300 экз. Заказ № 8385.

Отпечатано в АО «Первая Образцовая типография»

Филиал «Чеховский Печатный Двор»

142300, Московская область, г. Чехов, ул. Полиграфистов, д.1

Сайт: www.chpd.ru, E-mail: sales@chpd.ru, тел. 8(499)270-73-59

ООО “Альфа-книга”, 195027, Санкт-Петербург, Магнитогорская ул., д. 30

ISBN 978-5-9908462-8-9 (рус.)

© Компьютерное изд-во “Диалектика”, 2017,
перевод, оформление, макетирование

ISBN 0-13-597444-5 (англ.)

© Pearson Education, Inc., 2003

ОГЛАВЛЕНИЕ

Часть I. Гибкая разработка	29
Глава 1. Гибкие методики	30
Глава 2. Обзор экстремального программирования	41
Глава 3. Планирование	53
Глава 4. Тестирование	59
Глава 5. Рефакторинг	71
Глава 6. Случай из программирования	85
Часть II. Гибкое проектирование	133
Глава 7. Введение в гибкое проектирование	135
Глава 8. Принцип единственной обязанности (SRP)	149
Глава 9. Принцип открытости/закрытости (OCP)	155
Глава 10. Принцип подстановки Лисков (LSP)	169
Глава 11. Принцип инверсии зависимостей (DIP)	189
Глава 12. Принцип разделения интерфейса (ISP)	201
Часть III. Учебный пример: система расчета заработной платы	217
Глава 13. Команда и Активный объект	223
Глава 14. Шаблонный метод и Стратегия: наследование или делегирование	235
Глава 15. Фасад и Посредник	251
Глава 16. Одиночка и Моносостояние	257
Глава 17. Null-объект	271
Глава 18. Система расчета заработной платы: первая итерация	275
Глава 19. Система расчета заработной платы: реализация	291
Часть IV. Пакетирование системы расчета заработной платы	347
Глава 20. Принципы проектирования с использованием пакетов	348
Глава 21. Фабрика	371
Глава 22. Система расчета заработной платы (часть 2)	379
Часть V. Учебный пример: метеостанция	401
Глава 23. Компоновщик	402
Глава 24. Наблюдатель — подведение под паттерн	407
Глава 25. Абстрактный сервер, Адаптер и Мост	431
Глава 26. Заместитель и Лестница на небеса: управление API-интерфейсами от независимых поставщиков	443
Глава 27. Учебный пример: метеостанция	477
Часть VI. Учебный пример: ETS	517
Глава 28. Посетитель	518
Глава 29. Состояние	555
Глава 30. Инфраструктура ETS	585
Приложение А. Система обозначений UML: пример CGI	618
Приложение Б. Система обозначений UML: статистический мультиплексор	648
Приложение В. Сатирический рассказ о двух компаниях	670
Приложение Г. Исходный код и есть проект	687
Предметный указатель	700

СОДЕРЖАНИЕ

Предисловие	18
Вступление	19
Самое сложное кроется в деталях	20
Немного истории	20
Сотрудничество с Бучем	20
Влияние экстремального программирования	21
Сотрудничество с Беком	22
Структура книги	23
Как читать эту книгу	25
Если вы – разработчик...	25
Если вы – менеджер или бизнес-аналитик...	25
Если вы хотите изучить UML...	25
Если вы хотите изучить паттерны...	25
Если вы хотите изучить принципы объектно-ориентированного проектирования...	25
Если вы хотите изучить методы гибкой разработки...	25
Если вы хотите улыбнуться...	25
Ждем ваших отзывов!	26
Благодарности	27
Об авторах	28
Часть I. Гибкая разработка	29
Глава 1. Гибкие методики	30
Организация Agile Alliance	31
Манифест Agile Alliance	32
Принципы	36
Заключение	39
Библиография	40
Глава 2. Обзор экстремального программирования	41
Методики экстремального программирования	42
Заказчик как член команды	42
Пользовательские истории	42
Короткие циклы	43
Приемочные тесты	44
Парное программирование	44
Разработка через тестирование	45
Коллективное владение	46
Непрерывная интеграция	46
Жизнесспособный темп	47
Открытое рабочее пространство	47

Игра в планирование	48
Простое проектное решение	48
Рефакторинг	49
Метафора	50
Заключение	51
Библиография	51
Глава 3. Планирование	53
Первичное обследование	54
Пробные решения, разбиение и скорость	54
Планирование выпуска	55
Планирование итерации	56
Планирование задач	56
Точка посредине пути	57
Выполнение итераций	58
Заключение	58
Библиография	58
Глава 4. Тестирование	59
Разработка через тестирование	60
Пример проектирования с написанием тестов первыми	61
Изоляция тестов	62
Непрогнозируемое уменьшение связанности	64
Приемочные тесты	65
Пример приемочного тестирования	66
Непрогнозируемая архитектура	68
Заключение	68
Библиография	69
Глава 5. Рефакторинг	71
Генерация простых чисел: простой пример рефакторинга	72
Заключительный пересмотр	80
Заключение	84
Библиография	84
Глава 6. Случай из программирования	85
Игра в боулинг	86
Заключение	131
Часть II. Гибкое проектирование	133
Глава 7. Введение в гибкое проектирование	135
Что не так с программным обеспечением?	136
Признаки плохого проекта — особенности разлагающегося программного обеспечения	137
Что побуждает программное обеспечение разлагаться?	140
Гибкие команды не позволяют программному обеспечению разлагаться	140

Программа Copy	141
Гибкий проект примера программы Copy	145
Как гибкие разработчики узнали, что делать?	146
Поддержание проекта в таком виде, которое можно лучше видеть	147
Заключение	147
Библиография	147
Глава 8. Принцип единственной обязанности (SRP)	149
Принцип единственной обязанности (Single Responsibility Principle — SRP)	150
Что такое обязанность?	152
Разделение связанных обязанностей	153
Постоянство	153
Заключение	154
Библиография	154
Глава 9. Принцип открытости/закрытости (OCP)	155
Принцип открытости/закрытости (Open-Closed Principle — OCP)	156
Описание	156
Ключом является абстракция	157
Приложение Shape	158
Нарушение принципа OCP	159
Согласование с принципом OCP	161
Ладно, мы соглашаемся	162
Предвидение и “естественная” структура	163
Расстановка “ловушек”	164
Использование абстракции для явного закрытия	165
Использование подхода, управляемого данными, для достижения закрытия	167
Заключение	168
Библиография	168
Глава 10. Принцип подстановки Лисков (LSP)	169
Принцип подстановки Лисков (Liskov Substitution Principle — LSP)	170
Простой пример нарушения принципа LSP	170
Более тонкое нарушение принципа LSP в классах Square и Rectangle	172
Настоящая проблема	175
Достоверность не является внутренне присущей	176
Отношение “является” относится к поведению	176
Проектирование по контракту	177
Указание контрактов в модульных тестах	178
Реальный пример	178
Мотивация	178
Проблема	180
Решение, не удовлетворяющее принципу LSP	182
Решение, удовлетворяющее принципу LSP	182
Вынесение или наследование	183

Эвристические правила и соглашения	187
Вырожденные функции в производных классах	187
Генерация исключений в производных классах	188
Заключение	188
Библиография	188
Глава 11. Принцип инверсии зависимостей (DIP)	189
Принцип инверсии зависимостей (Dependency Inversion Principle — DIP)	190
Разделение на уровне	191
Инверсия владения	191
Зависимость от абстракций	192
Простой пример	193
Поиск лежащей в основе абстракции	195
Пример с котлом	196
Динамический или статический полиморфизм	198
Заключение	199
Библиография	199
Глава 12. Принцип разделения интерфейса (ISP)	201
Загрязнение интерфейсов	201
Разделение клиентов означает разделение интерфейсов	203
Обратное давление, оказываемое клиентами на интерфейсы	203
Принцип разделения интерфейса (Interface Segregation Principle — ISP)	205
Интерфейсы классов или интерфейсы объектов	205
Разделение посредством делегирования	205
Разделение посредством множественного наследования	206
Пример пользовательского интерфейса банкомата	207
Многоместная или одноместная форма	213
Заключение	215
Библиография	215
Часть III. Учебный пример: система расчета заработной платы	217
Глава 13. Команда и Активный объект	223
Простые команды	224
Транзакции	226
Физический и временный разрыв связей	227
Временный разрыв связей	228
Отмена	228
Паттерн Активный объект	229
Заключение	234
Библиография	234
Глава 14. Шаблонный метод и Стратегия: наследование или делегирование	235
Паттерн Шаблонный метод	236
Злоупотребление паттерном	240
Пузырьковая сортировка	240

Паттерн Стратегия	244
Возвращаемся к сортировке	247
Заключение	250
Библиография	250
Глава 15. Фасад и Посредник	251
Паттерн ФАСАД	252
Паттерн ПОСРЕДНИК	253
Заключение	255
Библиография	255
Глава 16. Одиночка и Моносостояние	257
Паттерн Одиночка	258
Преимущества паттерна Одиночка	260
Недостатки паттерна Одиночка	260
Паттерн Одиночка в действии	260
Паттерн Моносостояние	262
Преимущества паттерна Моносостояние	264
Недостатки паттерна Моносостояние	264
Паттерн Моносостояние в действии	264
Заключение	269
Библиография	270
Глава 17. Null-объект	271
Заключение	274
Библиография	274
Глава 18. Система расчета заработной платы: первая итерация	275
Введение	276
Спецификация	276
Анализ по сценариям использования	277
Добавление сотрудников	278
Удаление сотрудников	279
Отправка карточек табельного учета	280
Отправка товарных накладных	280
Отправка счетов за услуги профсоюза	281
Изменение сведений о сотрудниках	282
Расчетный день	284
Обдумывание: чему мы научились?	285
Поиск лежащих в основе абстракций	286
Абстракция графика	286
Методы платежа	288
Членство	288
Заключение	289
Библиография	289

Глава 19. Система расчета заработной платы: реализация	291
Добавление сотрудников	292
База данных сотрудников	294
Использование паттерна Шаблонный метод для добавления сотрудников	296
Удаление сотрудников	299
Глобальные переменные	301
Карточки табельного учета, товарные накладные и счета за услуги профсоюза	301
Изменение сведений о сотрудниках	310
Изменение классификации	314
Возникшая проблема	320
Начисление заработной платы сотрудникам	325
Хотим ли мы, чтобы разработчики принимали бизнес-решения?	327
Выплата заработной платы сотрудникам на окладе	328
Выплата заработной платы сотрудникам на почасовой оплате	331
Расчетные периоды: проблема проекта	335
Главная программа	343
База данных	344
Итоги проектирования системы расчета заработной платы	345
История	346
Библиография	346
Часть IV. Пакетирование системы расчета заработной платы	347
Глава 20. Принципы проектирования с использованием пакетов	348
Проектирование с применением пакетов?	349
Уровень детализации: принципы сцепления пакетов	350
Принцип эквивалентности повторного использования и выпуска (Reuse-Release Equivalence Principle — REP)	350
Принцип совместного повторного использования (Common-Reuse Principle — CRP)	352
Принцип общей закрытости (Common-Closure Principle — CCP)	353
Сводка по сцеплению пакетов	353
Устойчивость: принципы связанныности пакетов	354
Принцип ацикличности зависимостей (Acyclic-Dependencies Principle — ADP)	354
Еженедельная сборка	355
Устранение циклов в зависимостях	355
Влияние цикла в графе зависимостей между пакетами	357
Разрыв цикла	358
Изменчивость	359
Проектирование сверху вниз	360
Принцип устойчивых зависимостей (Stable-Dependencies Principle — SDP)	360
Устойчивость	361
Метрики устойчивости	362
Не все пакеты должны быть устойчивыми	363
Куда помешать высокоуровневое проектное решение?	365

Принцип устойчивых абстракций (Stable-Abstractions Principle — SAP)	366
Измерение абстракции	366
Главная последовательность	367
Расстояние от главной последовательности	368
Заключение	370
Глава 21. Фабрика	371
Цикл зависимостей	374
Замещаемые фабрики	375
Использование фабрик для тестовых оснасток	376
Насколько важно использовать фабрики?	378
Заключение	378
Библиография	378
Глава 22. Система расчета заработной платы (часть 2)	379
Структура пакетов и система обозначений	380
Применение принципа общей закрытости (CCP)	382
Применение принципа эквивалентности повторного использования и выпуска (REP)	384
Связанность и инкапсуляций	387
Метрики	388
Применение метрик к приложению расчета заработной платы	390
Фабрики объектов	394
Фабрика объектов для пакета TransactionImplementation	394
Инициализация фабрик	395
Переосмысление границ сцепления	396
Финальная структура пакетов	397
Заключение	399
Библиография	399
Часть V. Учебный пример: метеостанция	401
Глава 23. Компоновщик	402
Пример: составные команды	404
Множественность или не множественность	405
Глава 24. Наблюдатель — подведение под паттерн	407
Цифровые часы	408
Использование диаграмм в главе	427
Паттерн Наблюдатель	427
Как паттерн Наблюдатель соблюдает принципы объектно-ориентированного проектирования	429
Заключение	430
Библиография	430
Глава 25. Абстрактный сервер, Адаптер и Мост	431
Паттерн АБСТРАКТНЫЙ СЕРВЕР	432
Кто владеет интерфейсом?	433

Паттерн АДАПТЕР	434
Адаптер в форме класса	435
Задача с модемами, адаптеры и принцип LSP	435
Паттерн МОСТ	440
Заключение	442
Библиография	442
Глава 26. Заместитель и Лестница на небеса:	
управление API-интерфейсами от независимых поставщиков	443
Паттерн Заместитель	444
Применение паттерна Заместитель к корзине для покупок	449
Сводка по паттерну Заместитель	463
Работа с базами данных, промежуточным ПО и другими API-интерфейсами от независимых поставщиков	464
Паттерн Лестница на Небеса	467
Пример применения паттерна Лестница на Небеса	468
Другие паттерны, которые могут использоваться с базами данных	474
Заключение	476
Библиография	476
Глава 27. Учебный пример: метеостанция	477
Компания Cloud Company	478
Программное обеспечение WMS-LC	479
Выбор языка	480
Программный проект Nimbus-LC	481
24-часовая хронология и постоянство	497
Реализация алгоритма определения высших и низших значений	500
Заключение	509
Библиография	509
Обзор требований к Nimbus-LC	509
Эксплуатационные требования	509
24-часовая хронология	510
Настройка со стороны пользователя	510
Административные требования	511
Сценарии использования Nimbus-LC	511
Действующие лица	511
Сценарии использования	511
Хронология измерений	511
Настройка	512
Администрирование	512
План выпусков Nimbus-LC	513
Введение	513
Выпуск I	513
Риски	513
Доставляемые версии	514
Выпуск II	514
Реализованные сценарии использования	514

Риски	515
Доставляемые версии	515
Выпуск III	515
Реализованные сценарии использования	515
Риски	516
Доставляемые версии	516
Часть VI. Учебный пример: ETS	517
Глава 28. Посетитель	518
Семейство паттернов ПОСЕТИТЕЛЬ	519
Паттерн ПОСЕТИТЕЛЬ	519
Паттерн ПОСЕТИТЕЛЬ похож на матрицу	524
Паттерн Ациклический посетитель	524
Паттерн Ациклический посетитель похож на разреженную матрицу	529
Использование паттерна ПОСЕТИТЕЛЬ в генераторах отчетов	529
Другие применения паттерна ПОСЕТИТЕЛЬ	536
Паттерн ДЕКОРАТОР	537
Использование нескольких декораторов	541
Паттерн ОБЪЕКТ РАСШИРЕНИЯ	543
Заключение	553
Библиография	554
Глава 29. Состояние	555
Обзор конечных автоматов	556
Приемы реализации	558
Вложенные операторы switch/case	558
Интерпретация таблиц переходов	563
Паттерн Состояние	564
Компилятор конечных автоматов (SMC)	568
Где должны применяться конечные автоматы?	571
Высокоуровневые политики приложений для графических	
пользовательских интерфейсов	571
Контроллеры взаимодействия с графическим пользовательским интерфейсом	573
Распределенная обработка	574
Листинги	575
Реализация Turnstile.java с использованием интерпретации	
таблицы переходов	575
Реализация Turnstile.java, сгенерированная компилятором SMC,	
и другие поддерживающие файлы	577
Заключение	583
Библиография	583
Глава 30. Инфраструктура ETS	585
Введение	586
Обзор проекта	586
Хронология 1993–1994 годов	588
Инфраструктура?	589

Инфраструктура!	589
Команда 1994 года	589
Срок завершения	590
Стратегия	590
Результаты	591
Проект инфраструктуры	593
Общие требования к приложениям оценивания	593
Проект инфраструктуры оценивания	596
Случай для паттерна Шаблонный метод	600
Писать цикл один раз	601
Общие требования к приложениям выдачи	605
Проект инфраструктуры выдачи	606
Архитектура надзирателя	613
Заключение	617
Библиография	617
Приложение А. Система обозначений UML: пример CGI	618
Система записи на учебные курсы: описание задачи	619
Действующие лица	621
Сценарии использования	621
Модель предметной области	626
Архитектура	631
Абстрактные классы и интерфейсы в диаграмме последовательности действий	644
Заключение	647
Библиография	647
Приложение Б. Система обозначений UML: статистический мультиплексор	648
Определение статистического мультиплексора	648
Программная среда	649
Ограничения реального времени	650
Процедура обслуживания прерывания ввода	650
Процедура обслуживания прерывания вывода	655
Протокол связи	657
Заключение	669
Библиография	669
Приложение В. Сатирический рассказ о двух компаниях	670
Rufus, Inc. Провал проекта	670
Rupert Industries Проект: ~Alpha~	670
Приложение Г. Исходный код и есть проект	687
Предметный указатель	700