
Шаблоны реализации корпоративных приложений

Implementation Patterns

Kent Beck



ADDISON-WESLEY

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Capetown • Sydney • Tokio • Singapore • Mexico City

Шаблоны реализации корпоративных приложений

Кент Бек



Москва • Санкт-Петербург • Киев
2008

ББК 32.973.26-018.2.75
Б42
УДК 681.3.07

Издательский дом “Вильямс”

Зав. редакцией *С.Н. Тригуб*

Перевод с английского и редакция *А.В. Чеботарева*

По общим вопросам обращайтесь в Издательский дом “Вильямс” по адресу:
info@williamspublishing.com, <http://www.williamspublishing.com>

Бек, Кент.

Б42 Шаблоны реализации корпоративных приложений. : Пер. с англ. — М. : ООО “И.Д. Вильямс”, 2008. — 176 с. : ил. — Парал. тит. англ.

ISBN 978-5-8459-1406-4 (рус.)

Кент Бек, один из самых креативных и признанных лидеров в индустрии программного обеспечения, собрал 77 шаблонов, предназначенных для обслуживания задач ежедневного программирования и написания более читаемого кода. Эта новая коллекция шаблонов предназначена для реализации многих аспектов разработки, включая классы, состояние, поведение, методы, коллекции, инфраструктуры и т.д. Автор использует диаграммы, истории, примеры и эссе для того, чтобы увлечь читателя по ходу описания шаблонов. Вы обнаружите проверенные решения для управления всем, от именованых переменных до проверки исключений.

Эта книга предназначена для программистов всех уровней подготовки, особенно для тех, кто применяет в своей практике шаблоны проектирования и методы быстрой разработки. Книга также окажется неоценимым ресурсом для команд разработчиков, ищущих более эффективные методы совместной работы и построения более управляемого ПО.

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 Addison-Wesley Publishing Company, Inc.

Authorized translation from the English language edition published by Addison-Wesley Publishing Company, Inc. Copyright © 2008

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise.

No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

Russian language edition published by Williams Publishing House according to the Agreement with R&I Enterprises International, Copyright © 2008

ISBN 978-5-8459-1406-4 (рус.)
ISBN 0-321-41309-1 (англ.)

© Издательский дом “Вильямс”, 2008
© Pearson Education, Inc., 2008

Оглавление

Предисловие	15
Глава 1. Вступление	19
Глава 2. Шаблоны	23
Глава 3. Теория программирования	25
Глава 4. Мотивация	35
Глава 5. Класс	37
Глава 6. Состояние	59
Глава 7. Поведение	81
Глава 8. Методы	93
Глава 9. Коллекции	117
Глава 10. Развитие инфраструктур	133
Приложение А. Замеры производительности	147
Библиография	159
Предметный указатель	163

Содержание

Отзывы о книге	11
Предисловие	15
Благодарности	17
От издательства	18
Глава 1. Вступление	19
Краткое содержание	21
Итак... ..	22
Глава 2. Шаблоны	23
Глава 3. Теория программирования	25
Ценности	26
Взаимодействие	26
Простота	27
Гибкость	28
Принципы	28
Локализация последствий	29
Минимизация повторений	29
Объединение логики и данных	30
Симметрия	30
Описательные выражения	31
Частота изменений	32
Выводы	33
Глава 4. Мотивация	35
Глава 5. Класс	37
Класс	38
Простое имя суперкласса	39
Специальное имя subclasses	39
Абстрактный интерфейс	40
Интерфейс	41

Абстрактный класс	42
Интерфейс версий	43
Объект-значение	44
Специализация	46
Субкласс	47
Имплементор	49
Вложенный класс	50
Контекстно-зависимое поведение	51
Условия	51
Делегирование	53
Сменный селектор	55
Анонимный вложенный класс	56
Библиотечный класс	56
Выводы	57
Глава 6. Состояние	59
Состояние	60
Доступ	61
Прямой доступ	62
Косвенный доступ	63
Общее состояние	63
Переменное состояние	64
Внешнее состояние	66
Переменная	66
Локальная переменная	67
Поле	68
Параметры	69
Собирающий параметр	71
Оptionальный параметр	72
Аргументы-переменные	73
Объект-параметр	73
Константа	74
Имя, указывающее на роль	75
Определение типа	76
Инициализация	77
Ранняя инициализация	77
Отложенная инициализация	78
Выводы	79

Глава 7. Поведение	81
Управляющая логика	82
Главный алгоритм	82
Сообщение	82
Выбор сообщения	83
Двойная доставка	84
Декомпозиционное сообщение	85
Обратное сообщение	85
Приглашающее сообщение	86
Поясняющее сообщение	87
Алгоритм исключений	87
Сторожевой пункт	88
Исключения	90
Проверяемые исключения	90
Распространение исключений	91
Выводы	91
Глава 8. Методы	93
Составной метод	95
Имя, передающее замысел	96
Область видимости метода	97
Объект-метод	99
Переопределенный метод	101
Перегруженный метод	101
Возвращаемый тип	102
Комментарий к методу	102
Вспомогательный метод	103
Метод вывода отладки	104
Преобразование	104
Метод преобразования	105
Конструктор преобразования	106
Создание	106
Завершенный конструктор	107
Метод-фабрика	107
Внутренняя фабрика	108
Метод доступа к коллекциям	109
Метод установки булева значения	110
Метод запроса	111
Метод эквивалентности	111

Возвращающий метод	113
Устанавливающий метод	113
Безопасное копирование	115
Выводы	115
Глава 9. Коллекции	117
Метафоры	118
Сущности	119
Интерфейсы	120
Array	121
Iterable	121
Collection	122
List	122
Set	122
SortedSet	123
Map	124
Реализация	124
Collection	125
List	126
Set	126
Map	127
Collections	128
Поиск	128
Сортировка	129
Немодифицируемые коллекции	129
Коллекции из одного элемента	130
Пустые коллекции	131
Расширение коллекций	131
Выводы	132
Глава 10. Развитие инфраструктур	133
Изменение инфраструктур без изменения приложений	133
Несовместимые обновления	134
Осуществление совместимого изменения	136
Библиотечный класс	137
Объекты	137
Стиль использования	138
Абстракция	139
Создание экземпляров	142
Методы	144
Выводы	145

Приложение А. Замеры производительности	147
Пример	147
API	148
Реализация	149
MethodTimer	149
Исключая накладные расходы	152
Тесты	152
Сравнение коллекций	152
Сравнивая ArrayList и LinkedList	155
Сравнивая множества Set	156
Сравнение коллекций Maps	157
Выводы	158
Библиография	159
Общее программирование	159
Философия	161
Java	162
Предметный указатель	163

Отзывы о книге

“Кент — мастер кода и отличный рассказчик, книги которого можно легко и с удовольствием читать и понимать. Каждая глава этой книги содержит прекрасные объяснения и проникает в самую суть маленьких, но важных решений, постоянно принимаемых нами, когда мы создаем код и классы”.

Эрик Гамма (Erich Gamma), ведущий инженер IBM

“У многих команд есть ведущий разработчик, выдающий ряд правильных решений каждый день. Его мысли всегда движутся в верном направлении. Получающийся код легко понять и модифицировать, он безопасен и удобен. Эта книга поможет вам стать ведущим разработчиком команды. Ширина и глубина затронутых тем не оставят равнодушными профессионалов, которые увидят здесь новые приемы и улучшат старые методы, в то время как простота изложения делает книгу доступной и новичкам”.

Русс Руфер (Russ Rufer), Silicon Valley Patterns Group

“Многие люди не понимают, насколько читабельным может быть код и насколько ценна эта читабельность. Кент научил меня очень многому, и я рад, что эта книга дает шанс научиться и остальным”.

Мартин Фоулер (Martin Fowler), ведущий ученый ThoughtWorks

“Код должен заслуживать прочтения не только компилятором, но и людьми. Кент Бек превратил свой опыт в единый набор шаблонов реализации. Его советы сделают ваш код действительно заслуживающим прочтения”.

Грегор Хохне (Gregor Hohpe), автор Enterprise Integration Patterns

“В этой книге Кент Бек показывает, как из приложения простых принципов вытекает написание чистого и читабельного кода. *Шаблоны реализации* помогут разработчикам писать раскрывающий свой замысел код, который одновременно легок для понимания и гибок для дальнейшего расширения. Книгу должны обязательно прочитать все серьезные разработчики”.

Свен Гортс (Sven Gorts)

“Шаблоны реализации заполняют пробел между проектированием и программированием. Бек предлагает новый способ думать о программировании, основываясь на ценностях и принципах”.

*Диомидис Спинеллис (Diomidis Spinellis), автор Code Reading
и Code Quality*

Посвящается Синди. Спасибо тебе за поддержку, настойчивость, питание, ободрение, стимуляцию, редактирование и за чай. Моя преданность этой книге — как муравей рядом со слоном, по сравнению с тем, что ты сделала для меня. Будь здорова.

Предисловие

Эта книга о программировании, а именно о написании понятного кода. Нет никакой магии в создании кода, который могут читать другие люди. Любое писательство предполагает знание целевой аудитории, четкое понимание общей структуры предмета, выражение деталей так, чтобы они составляли единую историю. Язык Java предлагает хорошие средства коммуникации. Изложенные шаблоны реализации — это приемы Java-программиста, приводящие к написанию читабельного кода.

Другой взгляд на шаблоны реализации — это способ мышления: “Что я хочу рассказать читателю об этом коде?” Программисты проводят так много времени в своих собственных размышлениях, что попытка взглянуть на код с чужой точки зрения — значительный прогресс. Полезно задуматься не только над тем, “Что компьютер будет делать с этим кодом?”, но и “Как я могу пояснить свой образ мысли другим людям?” Это здоровый и потенциально прибыльный сдвиг перспективы, так как очень много денег в сфере разработки ПО тратится на понимание существующего кода.

В Америке есть шоу “Джеопарди”, в котором ведущий предлагает ответы, а соперники пытаются отгадать вопросы. “Человек, умеющий работать по дереву. — Столяр? — Правильно!”

Программирование сходно с “Джеопарди”. Язык Java предоставляет ответы в форме базовых конструкций. Программистам обычно нужно выразить вопросы для самих себя, понять, какие проблемы решаются каждой конструкцией языка. Если ответ — это объявление поля `Set`, то вопросом может быть “Как я могу рассказать другим программистам, что набор не содержит копий?” Шаблоны реализации составляют каталог общих проблем программирования и свойств Java, относящихся к этим проблемам.

Контроль объема работ при написании книги так же важен, как и в разработке ПО. Настоящая книга не является руководством по стилю, поскольку она содержит слишком много объяснений и оставляет последнее слово за читателем. Это не книга по проектированию, потому что здесь затрагиваются в основном решения мелкого масштаба, которые типичный программист делает по многу раз в день. Это не сборник шаблонов, так как их формат создан для частного применения. Наконец, это не описание языка, хотя книга и затрагивает многие свойства Java, но предполагается, что читатель уже знаком с этим языком.

Фактически книга основана на достаточно хрупком предположении — значимости хорошего кода. Я видел слишком много плохих программ, принесших большие деньги, чтобы не верить, что качество кода необходимо или достаточно для его коммерческого успеха или повсеместного использования. Однако я все еще верю в значимость качественного кода, даже если он не обеспечивает контроля над будущим.

Программы, способные уверенно развиваться и идти в народ, менять вектор развития в зависимости от возможностей и конкуренции и не терять свой дух перед сложными проблемами и неудачами, будут более успешны, чем программы с убогим, полным ошибок кодом.

Даже если тщательность в программировании не принесет долгосрочных экономических результатов, я все равно буду писать как можно более качественный код. Семьдесят лет жизни состоят из немногим более двух миллиардов секунд. Не хотелось бы их потратить на работу, которой я не горжусь. Хорошее программирование приносит удовольствие само по себе и дает знание, что другие будут способны понять, оценить, использовать и расширить мою работу.

Наконец, это книга об ответственности. Как программисту вам даны время, талант, деньги и возможности. Что вы сделаете для ответственного использования этих даров? На последующих страницах приведен ответ на этот вопрос для меня: программируй для других, как для самого себя и своего приятеля, центрального процессора.

Благодарности

Сначала, впоследствии и всегда я буду благодарен Синтии Андреес (Cynthia Andres) — партнеру, редактору, группе поддержки и главному “стимулятору”. Мой друг Пол Петралио (Paul Petralia) был вместе со мной во время работы над этим проектом и ободрял меня своими телефонными звонками. Мой редактор Крис Гужиковски (Chris Guzikowski) и я в ходе выполнения проекта наконец-то научились работать вместе. Он обеспечил мне поддержку со стороны редакции в Пирсоне, что позволило закончить книгу. Спасибо производственной бригаде из Пирсона: Джули Нахиль (Julie Nahil), Джону Фуллеру (John Fuller) и Синтии Когут (Cynthia Kogut). Дженнифер Конке (Jennifer Kohnke) изготовила информативные и отличные иллюстрации. Мои критики, Эрик Гамма (Erich Gamma), Стив Метскер (Steve Metsker), Диомидис Спинеллис (Diomidis Spinellis), Том де Марко (Tom deMarco), Михаэль Физерс (Michael Feathers), Дож Леа (Doug Lea), Брэд Абрамс (Brad Abrams), Клифф Клик (Cliff Click), Пекка Абрахамсон (Pekka Abrahamson), Грегор Хохпе (Gregor Hohpe) и Мишель Марчези (Michele Marchesi), обеспечивали четкую и своевременную обратную связь.

Спасибо Дэвиду Саффу (David Saff) за объяснение симметрии между состоянием и поведением. Мои дети, Линкольн, Линдсей, Форрест и Джолли Андре-Бек, оставались дома и поторапливали меня с окончанием.

От издательства

Вы, читатель этой книги, и есть главный ее критик и комментатор. Мы ценим ваше мнение и хотим знать, что было сделано нами правильно, что можно было бы сделать лучше, и что вы бы еще хотели увидеть изданным нами. Нам интересно услышать и любые другие замечания, которые вам хотелось бы высказать в наш адрес.

Мы ждем ваших комментариев и надеемся на них. Вы можете прислать нам бумажное или электронное письмо либо просто посетить наш веб-сервер и оставить свои замечания там. Одним словом, любым удобным для вас способом дайте нам знать, нравится или нет вам эта книга, а также выскажите свое мнение о том, как сделать наши книги более интересными для вас.

Посылая письмо или сообщение, не забудьте указать название книги и ее авторов, а также ваш обратный адрес. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию последующих книг.

Наши электронные адреса:

E-mail: info@williamspublishing.com

WWW: <http://www.williamspublishing.com>

Наши почтовые адреса:

в России: 127055, г. Москва, ул. Лесная, д. 43, стр. 1

в Украине: 03150, Киев, а/я 152

Глава 1

Вступление

И вот мы вместе. Вы взяли в руки мою книгу, теперь она ваша. Вы уже писали код, вероятно, у вас уже есть опыт и выработался собственный стиль.

Цель этой книги — помочь вам передать ваши замыслы в коде. Книга начинается с обзора программирования и шаблонов (главы 2–4). Оставшаяся часть книги (главы 5–8) содержит набор шаблонов и коротких заметок относительно написания читаемого кода на Java. Книга завершается главой о том, как нужно модифицировать приведенные советы в случае, если вы создаете не приложение, а инфраструктуру (framework). В целом книга фокусируется на технологиях программирования, которые расширяют возможности коммуникации.

Существует несколько шагов к началу взаимодействия через код. Для начала я должен был научиться программировать сознательно. Прошло несколько лет, прежде чем я начал воплощать свои мысли в шаблонах. Я был поражен тем фактом, что, хотя решения приходили ко мне легко и быстро, я не мог объяснить своей уверенности в том, что метод должен называться так-то или что данный предмет логически объясним. Первым шагом к началу взаимодействия стало замедление с целью осознать свои мысли и преодолеть мнение, будто я программирую инстинктивно.

Вторым шагом стало понимание важности остальных людей. Я находил удовлетворение в программировании, но был замкнут на себе. Перед тем как я смог написать взаимодействующий код, я должен был поверить, что остальные люди так же важны, как и я сам. Программирование вряд ли является просто контактом между человеком и машиной. Забота об остальных людях — это сознательное решение, требующее практики.

И таким образом я пришел к третьему шагу. С тех пор как я открыл свои мысли солнечному свету и свежему воздуху и понял, что другие люди имеют столько же прав на существование, как и я, у меня появилась необходимость продемонстрировать открывшиеся перспективы на практике. Я использую эти воплощенные принципы для того, чтобы программировать сознательно, для других так же, как для самого себя.

Вы можете прочитать эту книгу только ради технических деталей — полезных приемов с объяснениями. Однако я считаю нужным уведомить вас, что здесь есть нечто гораздо большее, по крайней мере для меня.

Вы можете найти эти технические части, пролистывая главы с шаблонами. Эффективная стратегия заключается в прочтении материала перед его использованием. Чтобы читать “непосредственно во время”, я предлагаю пропустить все, вплоть

до главы 5, и бегло просмотреть остаток до конца, после чего держать книгу под рукой, когда вы программируете. После того как вы используете некоторые шаблоны, всегда можно вернуться назад к вступительному материалу, чтобы ознакомиться с философской основой использованных идей.

Если же вы заинтересованы в тщательном осмыслении изложенного материала, то можете читать непосредственно с самого начала. В отличие от остальных моих книг, главы в этой книге довольно длинные, поэтому потребуется сконцентрироваться, чтобы прочесть целую часть от начала до конца.

Большая часть материала этой книги построена по принципу шаблонов. Большинство программных решений похожи на применявшиеся ранее. Вы можете определить миллион переменных за время вашей программистской карьеры. У вас не появится совершенно нового подхода к именованию каждой переменной. Общие ограничения к именованию всегда те же: вы должны донести назначение, тип и время жизни переменной до читателей, вы должны выбрать легко читаемое имя, вы должны выбрать имя, которое легко писать и форматировать. Добавьте к этому общие ограничения к специфике отдельной переменной, и вы получите ее рабочее имя. Наименование переменных является примером шаблона: решение и его ограничения повторяются, хотя вы можете создавать каждый раз другое имя.

Я думаю, шаблоны часто нуждаются в различных представлениях. Иногда шаблон лучше объяснить содержательной заметкой, иногда диаграммой, поучительной историей или примером. Вместо того чтобы втискивать каждый шаблон в жесткий формат, я описываю каждый из них наиболее соответствующим образом. Эта книга содержит 77 однозначно именованных шаблонов, каждый из которых затрагивает свой аспект написания читаемого кода. В дополнение здесь есть также много меньших шаблонов или их вариаций, которые я хотел бы упомянуть. Цель данной книги — посоветовать, как выполнять самые обыденные, ежедневные задачи программирования таким образом, чтобы будущие читатели могли понять, для чего нужен данный код.

Эта книга находится где-то посередине между книгой *Шаблоны проектирования* (Design Patterns) и руководством по Java. В книге о *шаблонах проектирования* речь идет о решениях, которые принимаются по несколько раз в день во время разработки приложений, обычно относящихся к регулировке взаимодействий между объектами. Вы применяете *шаблоны реализации* каждые несколько секунд, когда программируете. В то время как руководства по языку хороши для описания того, что вы можете сделать на Java, в них не говорится, почему вам следует использовать определенную конструкцию кода, или о выводах, которые может сделать кто-то, читающий ваш код.

Часть моей философии при написании этой книги заключалась в том, чтобы придерживаться хорошо знакомых мне тем. Результаты параллелизации, например, не входят в эти шаблоны реализации не потому, что многопоточность не является важным результатом, но потому, что это не то, о чем здесь следовало бы сказать. Моей стратегией многопоточности всегда была максимальная изоляция параллельных частей приложения. Пока что это мне в целом удается, так что я не хочу это разъяснять.