

Трейси, Саше и Софии

Введение

О чем эта книга

Эта книга о тесном соединении реорганизации кода, или рефакторинга, т.е. процесса улучшения проекта, с шаблонами проектирования — классическими решениями повторяющихся проблем проектирования. *Рефакторинг с использованием шаблонов* предполагает, что шаблоны лучше применять для улучшения уже существующего проекта, а не сразу при его создании. Это справедливо как для кода, существующего долгие годы, так и для кода, существующего минуты. Мы улучшаем проект с помощью шаблонов, применяя последовательность низкоуровневых преобразований проекта, называемую реорганизацией кода.

Каковы цели этой книги

Книга написана, чтобы помочь вам в следующем:

- понять как комбинировать рефакторинг с шаблонами;
- улучшить существующий проект с помощью управляемых шаблонами рефакторингов;
- определить области кода, нуждающиеся в управляемом шаблонами рефакторинге;
- узнать, почему лучше использовать шаблоны для усовершенствования существующего кода, чем непосредственно при создании нового проекта.

Для достижения этих целей в книге представлен следующий материал:

- каталог, содержащий 27 реорганизаций кода;
- примеры, основанные на коде, взятом из реальной жизни;
- описания шаблонов, включая примеры шаблонов из реальной жизни;
- коллекция признаков (т.е. проблем), указывающих на необходимость управляемых шаблонами реорганизаций кода;

- примеры различных способов реализации одного и того же шаблона;
- советы о том, когда следует проводить рефакторинг: добавляя в него шаблоны, изменяя использование шаблонов или отказываясь от шаблонов вовсе.

Чтобы помочь отдельным лицам или группам изучить 27 реорганизаций кода из этой книги, на обратной стороне обложки вы найдете рекомендуемую последовательность изучения.

Кому следует читать эту книгу

Книга предназначена для объектно-ориентированных программистов, занятых или интересующихся улучшением существующего проекта. Одни из них используют шаблоны и/или выполняют рефакторинг, но никогда не реализуют шаблоны в процессе реорганизации; другие мало знакомы с реорганизацией кода и шаблонами и хотели бы узнать об этом больше.

Книга полезна как при разработке “с чистого листа”, когда вы пишете новую систему или функцию с нуля, так и при сопровождении, когда вы в основном поддерживаете унаследованную систему.

Какие знания вам необходимы

В книге предполагается, что вы знакомы с такими концепциями проектирования, как сильное связывание (tight coupling) и слабое связывание (loose coupling), а также с объектно-ориентированными понятиями наследования (inheritance), полиморфизма (polymorphism), инкапсуляции (encapsulation), композиции (composition), интерфейса (interface), абстрактных и конкретных классов (abstract class, concrete class), абстрактных и статических методов (abstract and static methods) и т.п.

Примеры, приведенные в книге, написаны на языке Java. Я считаю, что Java наиболее легко читаем для большинства объектно-ориентированных программистов. Я изо всех сил старался не использовать особенности Java, поэтому независимо от того, пишете ли вы на C++, C#, Visual Basic .NET, Python, Ruby, Smalltalk или другом объектно-ориентированном языке, вы сможете понять Java-код из этой книги.

Настоящая книга тесно связана с классической книгой Мартина Фаулера (Martin Fowler) *Refactoring* [15]. Она содержит ссылки на такие низкоуровневые реорганизации кода:

- *Extract Method*;
- *Extract Interface*;
- *Extract Superclass*;
- *Extract Subclass*;

- *Pull Up Method*;
- *Move Method*;
- *Rename Method*.

Книга *Refactoring* также содержит ссылки на более сложные реорганизации кода, а именно:

- *Replace Inheritance with Delegation*;
- *Replace Conditional with Polymorphism*;
- *Replace Type Code with Subclasses*.

Чтобы понять управляемые шаблонами варианты реорганизации кода из этой книги, вам вовсе не обязательно знать все упомянутые реорганизации кода. Вместо этого вы можете следовать за образцами кода, иллюстрирующими реализации перечисленных реорганизаций. Однако, если вы хотите получить от этой книги как можно больше, рекомендую иметь под рукой книгу *Refactoring*, так как она является бесценным источником реорганизаций кода и немало способствует пониманию книги, которую вы сейчас читаете.

Шаблоны, о которых я пишу, берут свое начало из классической книги *Design Patterns* [12], а также от Кента Бека (Kent Beck), Бобби Вульфа (Bobby Woolf) и меня. В реальных проектах мы с коллегами реорганизовывали код, используя эти шаблоны, применяя их иначе или отказываясь от них. Изучив искусство управляемой шаблонами реорганизации кода, вы поймете, как реорганизовывать код, добавляя, изменяя использование или отказываясь от шаблонов, не упомянутых в этой книге.

Для чтения книги нет необходимости быть экспертом в этих шаблонах, хотя некоторые знания шаблонов будут полезны. Чтобы помочь вам понять использованные в книге шаблоны, в нее включены короткие резюме о шаблонах, UML-наброски¹ шаблонов и множество примеров реализации шаблонов. Чтобы досконально понять шаблоны, рекомендую изучать эту книгу вместе с упомянутой литературой по шаблонам.

В книге используются диаграммы UML 2.0. Если вы не слишком хорошо знаете UML, то не волнуйтесь — я тоже. Я знаю лишь основы. Во время написания книги я имел под рукой и часто обращался к третьему изданию книги Фаулера (Fowler) *UML Distilled* [17].

¹ Unified Modeling Language — унифицированный язык моделирования. — *Примеч. пер.*

Как пользоваться этой книгой

Для высокоуровневого понимания реорганизаций кода из этой книги можно начать с изучения резюме всех реорганизаций кода (см. главу 5, “Каталог рефакторингов к шаблонам”), а также описаний их преимуществ и недостатков.

Для более глубокого понимания рефакторинга следует изучить каждую часть описания метода реорганизации, за исключением разве что раздела *Механика*; это раздел дополнительный, его назначение — помочь вам реализовать реорганизацию кода, подсказывая, какой именно низкоуровневой реорганизации кода стоит следовать. Для понимания методов рефакторинга в этой книге данный раздел читать не обязательно. Вероятно, вы будете его использовать только в качестве справочной информации, когда займетесь реорганизацией реального кода.

Признаки проблем, описанные в этой книге и в *Refactoring* [15], обеспечивают практичный способ определения проблемы проектирования и поиска метода реорганизации кода для ее решения. Можно также просмотреть алфавитный перечень рефакторингов, чтобы найти метод реорганизации, способный улучшить проект.

Эта книга документирует методы реорганизации кода, которые ведут проект как к использованию (или изменению использования) шаблонов, так и к отказу от них. Чтобы помочь понять, в каком направлении следует двигаться, вы можете обратиться к соответствующему разделу главы 3, “Шаблоны”.

История книги

Я начал писать эту книгу примерно в 1999 году. Что же подвигло меня написать книгу о шаблонах, рефакторинге и экстремальном программировании (extreme programming — XP) [8]?

Во-первых, я был удивлен, что шаблоны не нашли отражения в XP-литературе. Это убедило меня написать работу *Patterns & XP* [20], в которой публично обсуждался предмет и предлагались некоторые советы по объединению этих двух больших сущностей в нашей области знаний.

Во-вторых, я знал, что Мартин Фаулер (Martin Fowler) включил в книгу *Refactoring* [15] только несколько реорганизаций с использованием шаблонов и ясно высказал надежду, что кто-то напишет больше. Это было стоящей целью.

Наконец, я заметил, что людям из The Design Patterns Workshop (семинара по шаблонам проектирования, который вели мои коллеги и я) требуется большая помощь в определении того, когда именно следует применять шаблон к проекту. Одно дело узнать суть шаблона, и совсем другое — действительно понять, когда и как его следует применять. Я считал, что мои слушатели нуждались в изучении примеров из реальной жизни, в которых применение шаблона к проекту было обоснованным, и так я начал сбор материала для каталога таких примеров.

Как только я начал писать книгу, следуя примеру Брюса Эккела, я поместил приблизительное ее содержание в Web, чтобы обеспечить обратную связь. Это было действительно прекрасной идеей: многие коллеги ответили предложениями по содержанию книги, одобрением и признательностью.

Как только мои тексты и идеи приобрели “товарный вид”, я начал представлять их широкой общественности. Это дало новый поток предложений по улучшению книги и множество идей, необходимых программистам для понимания предмета.

Постепенно я пришел к пониманию того, что на шаблоны лучше всего смотреть с точки зрения рефакторинга и что они являются пунктом назначения, до которого проще всего добираться, применяя последовательность низкоуровневых реорганизаций кода.

Мне повезло: когда мой текст начал больше походить на книгу, чем на длинную статью, многие опытные практики просмотрели мою работу и дали советы по ее улучшению. Все эти люди упоминаются в разделе благодарностей.

Стоя на плечах гигантов

Летом 1995 года, рассматривая в книжном магазине литературные новинки, я впервые наткнулся на книгу *Design Patterns* [12] и влюбился в шаблоны. Я хочу поблагодарить ее авторов — Эриха Гамму (Erich Gamma), Ричарда Хелма (Richard Helm), с которым я до сих пор не встретился, Ральфа Джонсона (Ralph Johnson) и Джона Влиссайдза (John Vlissides) — за написание этого прекрасного произведения. Знания, которыми вы поделились в своей книге, помогли мне, как разработчику программного обеспечения, существенно повысить свой профессиональный уровень.

С Мартином Фаулером (Martin Fowler) я встретился на конференции по шаблонам приблизительно в 1996 году, еще до того, как он стал известен. Это стало началом долгой дружбы. Я так бы и сомневался, должен ли я писать эту книгу, если бы Мартин (и его коллеги Кент Бек (Kent Beck), Уильям Опдайк (William Opdyke), Джон Брант (John Brant) и Дон Робертс (Don Roberts)) не написали классическую книгу *Refactoring* [15], кардинально изменившую мой подход к проектированию программного обеспечения.

То, что я написал эту книгу, стало возможным только благодаря тяжелому труду авторов *Design Patterns* и *Refactoring*. Я бесконечно благодарен им за их великие книги.

Благодарности

Я неизмеримо счастлив от того, что моя жена с любовью поддерживала меня во время работы над книгой. Трейси, ты самая замечательная! Я с радостью предвкушаю многие десятилетия, которые мы проведем вместе.

Пока я писал книгу, родились две наши дочери, Саша и София. Я благодарен им за то, что они терпеливо выносили те долгие часы, когда их папа писал, писал и писал.

В 1970-х годах Брюс Кериевски (Bruce Kerievsky), мой отец, взял нас с братом к себе на работу, где мы могли рисовать гигантские компьютеры в комнатах с кондиционерами. Он приносил нам громадные зеленые и белые компьютерные распечатки, на которых огромными буквами были напечатаны наши имена. Его работа помогла мне заинтересоваться этой обширной областью. Спасибо ему.

Теперь, когда я высказал все сентиментальные благодарности своей семье, я могу сосредоточиться на специальных благодарностях. А их немало. Наибольший вклад в эту книгу внес Джон Брант (John Brant). Он и его коллега Дон Робертс (Don Roberts) — одни из наиболее осведомленных людей в мире в области реорганизации кода. Джон, просмотревший все четыре рабочие версии рукописи, внес много важных идей и помог избавиться от идей не столь важных. Дон, который был занят в других проектах, не мог дать так много, хотя он просматривал замечания Джона, и я благодарен ему за это. Также я благодарен им обоим за написание послесловия к этой книге. Мартин Фаулер (Martin Fowler) помог мне многочисленными рецензиями и советами, включая то, как упростить общие описания и сделать понятными технические обсуждения. Он помог мне исправить некоторые из моих неверных UML-диаграмм и обновил их до уровня UML 2.0. Мартин удостоил меня чести, включив эту книгу в свою именную серию. Я благодарен ему за предисловие к книге, которое он написал.

Свен Горц (Sven Gorts) просматривал практически все рабочие версии рукописи и высылал мне потоки содержательных замечаний. Он предоставил много полезных идей, улучшивших текст, диаграммы и код в этой книге.

Сомик Раха (Somik Raha) помог во многом улучшить содержание этой книги. Его открытый проект HTML-анализатора, который он начал сразу после тщательного изучения шаблонов, оказался золотым дном кода, нуждающегося в реорганизации с введением в него шаблонов. Мы с Сомиком много работали вместе, и я безгранично благодарен ему за поддержку, поощрение и советы.

Эрик Эванс (Eric Evans), автор книги *Domain-Driven Design*, предоставил отзывы на ранние версии рукописи. В то время, когда мы оба работали над книгами, мы часто встречались в кафе в Сан-Франциско и его окрестностях. Там мы могли писать, обмениваться ноутбуками и комментировать тексты друг друга. Благодарю Эрика за отзывы и компанию.

Крис Лопес (Chris Lopez), член Silicon Valley Patterns Group (SVPG), предоставил множество высокодетализированных и полезных замечаний по поводу моего

текста, диаграмм и кода. Хотя другие члены SVPG также заслуживают благодарности, Крис вышел далеко за рамки своих обязанностей в отзывах об этой книге.

Рус Руфер (Russ Rufer), Трейси Бялик (Tracy Bialik) и другие программисты из SVPG (включая Теда Янга (Ted Young), Фила Гудвина (Phil Goodwin), Алана Харримана (Alan Harriman), Чарли Толанда (Charlie Toland), Боба Эванса (Bob Evans), Джона Брювера (John Brewer), Джеффа Миллера (Jeff Miller), Дэвида Видра (David Vydra), Дэвида У. Смита (David W. Smith), Патрика Мэниона (Patrick Manion), Джона Ву (John Wu), Деби Атли (Debbie Utley), Кэрол Систлетвейт (Carol Thistlethwaite), Кена Скот-Хлебeka (Ken Scott-Hlebek), Саммера Мишержи (Summer Misherghi) и Сиквинг Женг (Siqing Zhang)) множество раз просматривали как ранние, так и более зрелые версии этой книги. Они помогли мне осознать, что необходимо сделать более понятным, что расширить, а что удалить. Отдельное спасибо Русу за такое большое количество встреч для обсуждения книги и Джеффу за запись мыслей, высказанных группой.

Ральф Джонсон (Ralph Johnson) и его группа Patterns Reading Group из университета Иллинойса предоставили невероятно полезные отзывы на раннюю версию рукописи в виде MP3-файлов. Я провел много времени, слушая их записи и действуя сообразно их советам. Я особенно благодарен Ральфу, Брайану Футу (Brian Foote), Джозефу Йодеру (Joseph Yoder) и Брайану Марику (Brian Marick) за их мысли и советы. Разумеется, в группе были и многие другие; к сожалению, я не знаю их имен, поэтому прошу простить меня и приношу мои благодарности. Отдельное спасибо Ральфу за предисловие к этой книге.

В некоторых случаях чрезвычайно полезные отзывы, включая много подробных заметок на первом черновике рукописи, давал Джон Влиссайдз (John Vlissides). Он постоянно поощрял мою работу, чем очень помог.

Эрих Гамма (Erich Gamma) сделал несколько прекрасных предложений как по поводу вводного материала этой книги, так и по реорганизации кода.

Кент Бек (Kent Beck) просмотрел несколько методов реорганизации кода из этой книги и предложил диаграмму *Inline Singleton*. Я также благодарен ему за совместную работу над шаблоном *State* на конференции XP2002, в Алгхеро, Италия.

Вард Каннингем (Ward Cunningham) внес вклад в диаграмму *Inline Singleton* и в последний момент дал дельный совет по размещению главы во вводном материале. Дирк Баумер (Dirk Baumer), ведущий программист Eclipse, и Дмитрий Ломов (Dmitry Lomov), ведущий программист IntelliJ, также внесли массу предложений ко многим методам реорганизации кода из этой книги.

Кайл Браун (Kyle Brown) просматривал первый черновик рукописи и проявил отменное понимание вопроса. Кен Ширифф (Ken Shirriff) и Джон Тангни (John Tangney) дали исчерпывающие содержательные замечания на версии рукописи.

Кен Томасес (Ken Thomases) указал на критическую ошибку в старых версиях описания механики *Replace Type Code with Class*.

Роберт Хершфилд (Robert Hirshfeld) помог прояснить механику в самых ранних версиях *Move Embellishment to Decorator*.

Рон Джеффрис (Ron Jeffries) помог мне четко выделить основную тему этой книги, будучи вовлеченным в бесчисленные дискуссии на `extremeprogramming@yahooogroups.com`.

Дмитрий Кериевски (Dmitri Kerievsky) выполнил литературную обработку текста введения.

По ходу дела полезные советы мне давали коллеги Гунджан Доши (Gunjan Doshi), Джеф Григ (Jeff Grigg), Каору Хосокава (Kaoru Hosokawa), Дон Хинтон (Don Hinton), Эндрю Сван (Andrew Swan), Эрик Миде (Erik Meade), Крейг Демьянович (Craig Demyanovich), Дэйв Гувер (Dave Hoover), Роб Ми (Rob Mee) и Алекс Чеффи (Alex Chaffee).

Я также благодарен за отзывы коллегам, обсуждавшим методы реорганизации из этой книги на `refactoring@yahooogroups.com`.

Я хотел бы поблагодарить тех слушателей Industrial Logic, семинара по шаблонам проектирования The Design Patterns Workshop и семинара по тестированию и реорганизации кода The Testing & Refactoring Workshop, которые дали отзывы на методы реорганизации кода из этой книги. Многие из вас помогли мне понять, что в книгу необходимо добавить, а что убрать из нее. Особая благодарность моему редактору Полу Петралиа (Paul Petralia) и его команде (Лиза Ярковски (Lisa Iarkowski), Файе Джеммелларо (Faye Gemmellaro), Джон Фуллер (John Fuller), Ким Арни Малкахи (Kim Arney Mulcahy), Криста Мидоубрук (Chrysta Meadowbrooke), Ребекка Райдер (Rebecca Rider) и Ричард Эванс (Richard Evans)). Когда издатели соперничали за публикацию этой книги, Пол отчаянно боролся, чтобы право издать ее досталось Addison-Wesley. Я очень рад, что именно так и произошло. Многие годы я читал великие книги Addison-Wesley, а теперь и я удостоен чести стать членом этой семьи. За время написания книги Пол стал моим другом. Когда он не произносил патетические речи о том, что надо поскорее заканчивать книгу, мы много разговаривали о наших детях, играли в теннис, смеялись и даже иногда выпивали. Спасибо, Пол. Я рад, что ты был моим редактором.