

*Посвящается Энн Мэйн (Ann Marie), Анжеле (Angela), Мики (Micah), Джине (Gina), Джастину (Justin), Анжелике (Angelique), Мэтту (Matt) и Алексис (Alexis).*

## Введение



— Но Боб, вы же сказали, что книга будет готова еще в прошлом году.

---

Клаудия Фрес, UML World, 1999

Быстрая разработка предусматривает способность к скоростному созданию программ в условиях непрерывно изменяющихся требований. В связи с этим приходится использовать методы, гарантирующие необходимую организацию работы и обратную связь с разработчиком. При этом следует применять методы проектирования, которые обеспечивают гибкость и простоту дальнейшей эксплуатации ПО. Весьма желательно, чтобы разработчики имели представление о шаблонах проектирования, позволяющих использовать принципы быстрой разработки ПО в целях решения определенных проблем. В данной книге предпринята попытка (довольно успешная) связать все упомянутые концепции в единое целое.

Внимательный читатель найдет здесь описание принципов, шаблонов и методов быстрой разработки ПО, а также примеры их применения на практике. И что более важно, конкретные примеры представлены не как выполненные задачи, а скорее в виде разрабатываемых проектов. Благодаря подобному подходу вы наглядно увидите, как ошибаются разработчики проектов, каким образом происходит процесс идентификации ошибок и их дальнейшее исправление. Также

наглядно демонстрируется процесс решения сложных задач и процедуры анализа возникших противоречий и компромиссов. При этом вы сможете наблюдать сам процесс разработки и выполнения программных проектов “живьем”.

## Демонология: весь смысл — в деталях

Эта книга включает *большой объем* кода на языках Java и C++. Я надеюсь, что вы внимательно прочтете этот код, поскольку в значительной степени именно он является *ключевым моментом* в изучении материала книги.

В тексте повторяется использование одного и того же шаблона. Его суть раскрывается в процессе рассмотрения нескольких конкретных примеров, причем соответствующий программный код может быть кратким или достаточно объемным. Рассмотрению каждого конкретного примера предшествует специальный материал, цель которого заключается в том, чтобы подготовить читателя к дальнейшему рассмотрению самого программного кода. Например, рассмотрению программы, реализующей расчет зарплаты, предшествуют главы, описывающие объектно-ориентированные принципы и шаблоны проектирования, которые используются в процессе создания программного кода.



Книга начинается с рассмотрения методов и процессов разработки программ. Данное обсуждение детализируется путем обсуждения небольших конкретных случаев и примеров. Затем в книге рассматривается проектирование и его принципы, некоторые из шаблонов проектирования, а потом — отдельные принципы проектирования, обеспечивающие создание программных пакетов, а также дополнительные шаблоны.

Данная книга носит сугубо технический характер, поэтому детали имеют определяющее значение.

## Немного истории

Более шести лет тому назад я написал книгу *Проектирование объектно-ориентированных приложений на языке C++ с использованием метода Буча (Designing Object-oriented C++ Applications using the Booch Method)*. Для меня эта книга стоила больших усилий, и я остался доволен полученными результатами, а также большими объемами продаж.

Данная книга изначально была задумана как второе издание “*Проектирования...*”, но итогом переработки явилось существенное изменение исходного издания. В частности, в данную книгу попали всего лишь три главы из предыдущей книги, которые были серьезно переработаны. Остались практически без изме-

нений содержание, концепция и большинство практических занятий из прежней книги. Но за прошедшие шесть лет автор существенно пополнил и расширил свои знания о проектировании и разработке ПО. Все это получило адекватное отображение в новой книге.

Да, много воды утекло с момента появления первого издания! *Проектирование*. . . появилось на свет накануне эпохи бурного развития Internet. С того времени количество аббревиатур, сопровождающих нас в повседневной жизни, удвоилось. Мы свободно оперируем терминами Design Patterns (шаблоны проектирования), Java, EJB, RMI, J2EE, XML, XSLT, HTML, ASP, JSP, Servlets (сервлеты), Application Servers (серверы приложений), ZOPE, SOAP, C#, .NET и т.д. Позвольте заметить, что “идти в ногу со временем” в процессе подготовки текста данной книги оказалось весьма непросто!

## Сотрудничество с Бучем

В 1997 году ко мне обратился Гради Буч (Grady Booch) с просьбой помочь ему подготовить третье издание поразительно успешной книги *Объектно-ориентированный анализ и проектирование приложений (Object-Oriented Analysis and Design with Applications)*. Прежде мне приходилось работать с Гради над несколькими проектами, и я с большим интересом знакомился с его различными работами, в том числе посвященными описанию UML-диаграмм. Поэтому я с радостью принял его предложение, попросив о помощи своего хорошего друга Джеймса Ньюкирка.

За последующие два года Джеймс и я совместно написали несколько глав для книги Буча. Конечно, можно было бы сделать и больше, но “нельзя объять необъятное. . .”. Более того, книга, которую вы сейчас держите в руках, была в то время, по сути, просто вторым изданием *Проектирования*. . . , и мне не слишком хотелось продолжать работу над ней. Всегда хочется сказать нечто новое, носящее революционный характер.

К сожалению, этой книге Буча не суждено было увидеть свет. Трудно выкроить время для написания книги в череде будней. В горячие дни работы над проектом “.com” это было практически невозможно. С течением времени Гради переориентировался на работу с новыми проектами. Поэтому наш совместный проект попал в “долгий ящик”. В конце концов, я попросил у Гради и руководства издательства Addison-Wesley разрешение включить главы, написанные мною и Джеймсом, в эту книгу. Они любезно согласились. Поэтому несколько примеров и глав, посвященных описанию UML-диаграмм, берут начало именно из этого источника.

## Экстремальное программирование и его влияние на практику разработки программных проектов

Идея экстремального программирования (XP, Extreme Programming), подвергшая сомнению наши изначальные представления о разработке ПО, возникла в конце 1998 года. Следует ли разрабатывать множество UML-диаграмм, прежде чем приступить к написанию какого-либо программного кода, или же лучше воздержаться от этого и начинать реальную работу? Нужно ли создавать множество документов с комментариями, описывающими проект, или же имеет смысл попытаться придать выразительность самому *коду*, в результате чего будет устранена необходимость в разработке вспомогательных документов? Следует ли практиковать метод парного программирования? Нужно ли разрабатывать тесты перед созданием программного кода? Как вообще следует поступать в тех или иных ситуациях?

Как по мне, так революция в программировании произошла вовремя. Во второй половине 90-х годов прошлого столетия компания Object Mentor предоставила услуги нескольким компаниям при решении вопросов, возникающих в процессе объектно-ориентированного (ОО) проектирования и управления проектами. Мы же всего лишь помогали компаниям *довести* работу над проектами *до конца*. При этом мы делились с членами команд разработчиков своими взглядами на процесс разработки, а также внедряли собственные методы. К сожалению, эти взгляды и методы не были зафиксированы в письменном виде. Заказчики получали все необходимые сведения в устной форме.

В 1998 году я осознал, что следует в письменном виде сформулировать применяемый процесс и методы разработки, что позволит в более эффективной форме сформулировать их для наших заказчиков. Это побудило меня к написанию большого количества статей, посвященных рассмотрению процесса разработки проектов, которые были опубликованы в журнале *C++ Report*<sup>1</sup>. Но этим статьям не хватало широты взглядов, а также отсутствовала интересная подача материала. Они были информативны и даже занимательны, но вместо того, чтобы систематизировать методы и взгляды, которые применялись при разработке проектов, они выполняли функции “пропаганды” ценностей, которые навязывались мне на протяжении десятилетий. На этот недостаток и указал мне Кен Бек (Kent Beck).

### Участие Бека в проекте

В конце 1998 года, в процессе усердной работы над кодированием процесса Object-Mentor, я случайно наткнулся на работу Кена по экстремальному программированию (XP). Фрагменты статьи были размещены на Web-узле Уорда

---

<sup>1</sup>Эти статьи можно найти в разделе “publications” на Web-узле <http://www.objectmentor.com>. Первые три из них называются “Iterative and Incremental Development” (I. II. III). Последняя статья называется “C.O.D.E Culled Object Development procEss”.

Каннингема (Ward Cunningham) под названием *wiki*<sup>2</sup>, где они были представлены вперемешку с работами других авторов. С некоторым трудом мне удалось понять суть того, о чем говорил Кен. Я был заинтригован, но настроен весьма скептически. Некоторые из аспектов, которые были отображены в идеологии экстремального программирования, в точности совпадали с моей концепцией процесса разработки. Другие аспекты, такие как отсутствие четкого описания этапов проектирования, напротив, привели меня в некоторое замешательство.

Более сумасшедших обстоятельств разработки ПО, чем те, с которыми приходилось сталкиваться мне и Кену, трудно было бы придумать. Кен был признанным консультантом в области языка Smalltalk, а я — консультантом по языку C++. И этим двум мирам было затруднительно найти общие точки пересечения. Между ними находилась пропасть парадигм Кана (Kuhn)<sup>3</sup>.

При других обстоятельствах я никогда бы не попросил Кена написать статью для журнала *C++ Report*. Но наши представления о процессе разработки совпали, и именно это помогло преодолеть определенный барьер. В феврале 1999 года я встретился с Кеном в Мюнхене на конференции, посвященной объектно-ориентированному программированию. Он читал лекцию на тему XP-программирования в зале как раз напротив аудитории, в которой я читал лекцию о принципах объектно-ориентированной разработки. Поскольку я не имел возможности послушать лекцию Кена, я разыскал его во время обеда. Мы обсуждали проблемы экстремального программирования, и я попросил его написать статью для журнала *C++ Report*. Статья получилась великолепной. В ней описывался случай, в котором Кену и его сотруднику удалось коренным образом изменить ранее разработанный проект за считанные часы.

За несколько месяцев мне удалось преодолеть собственные страхи в отношении экстремального программирования. Больше всего я опасался реализации процесса, когда отсутствует четко выраженный этап предварительного проектирования. Я обнаружил, что именно из-за этого существенно тормозилась разработка программ. Чувство ответственности перед заказчиками заставляет меня убеждать всех заинтересованных лиц в том, что важность проектирования заслуживает потраченного на него времени?

Со временем я осознал, что сам не слишком часто использовал эту методику на практике. Даже в моих статьях и книгах, посвященных проектированию, диаграммам Буча и UML-диаграммам, я всегда использовал код в качестве метода подтверждения смысловой значимости диаграмм. Во всех своих консультациях с заказчиками я обычно тратил час или два на то, чтобы помочь им начертить

---

<sup>2</sup><http://c2.com/cgi/wiki>. Этот Web-узел включает множество статей, посвященных самым различным темам. Количество пишущих авторов измеряется тысячами. Именно здесь было заявлено, что Уорд Каннингем осуществил социальную революцию с помощью нескольких строк кода на Perl.

<sup>3</sup>Thomas S. Kuhn. *The Structure of Scientific Revolutions*. The University of Chicago Press, 1962.

диаграммы, а затем рекомендовал протестировать их с помощью разработанного кода. Я начал понимать, что, несмотря на то, что концепция экстремального программирования была мне чужда (по принципу Кана<sup>4</sup>), методы, лежащие в ее основе, были знакомы и давно применялись на практике.

Если ранее я предпочитал работать в одиночку, то экстремальное программирование позволило мне осознать все преимущества парного программирования. Рефакторинг, непрерывную интеграцию и присутствие заказчика в период разработки — все было воспринято мною вполне естественно.

Один из методов, практикуемых в экстремальном программировании, оказался для меня открытием — методика пробных испытаний (test-driven development). Этот метод подразумевает разработку тестовых случаев перед созданием фактического кода. При написании производственного кода первоочередной задачей становится успешный прогон сбойных тестов. Я не был готов к появлению множества вариантов выполнения кода, которые возникают в результате применения подобных методов программирования. Это полностью изменило мой собственный метод разработки программ, причем в лучшую сторону (что можно увидеть в этой книге). Некоторый объем кода, представленного в данной книге, был разработан в 1999 году. Здесь отсутствуют тестовые случаи. С другой стороны, все примеры кода, написанные после 1999 года, сопровождаются тестовыми случаями, и, как правило, эти тестовые случаи рассматриваются в первую очередь. Я уверен в том, что вы обратите внимание на данное отличие.

Итак, к осени 1999 года я имел возможность убедиться, что экстремальное программирование следует включить в модель компании Object Mentor в качестве альтернативного процесса, а также то, что мне следует отказаться от желания описать разработанный мною процесс. Сформулировав методы и описание процесса экстремального программирования, Кен выполнил отличную работу, на фоне которой мои попытки сделать нечто аналогичное выглядели явно неадекватно.

## Структура книги

- Часть I. *Быстрая разработка ПО*. Приводится описание концепции быстрой разработки и принципов экстремального программирования (XP), причем особое внимание уделено тем из них, которые оказывают влияние на выбор метода проектирования и написания кода.
- Часть II. *Быстрое проектирование*. Здесь идет речь об объектно-ориентированном проектировании ПО, в частности о том, что представляет собой проектирование, какие существуют методы преодоления различных проблем. Здесь же вы найдете описание принципов проектирования на уровне объектно-ориентированных классов.

---

<sup>4</sup>Дважды упомянутое имя Кана говорит о многом. . .

- Часть III. *Практическое занятие: программа расчета зарплаты*. Здесь вы найдете самое объемное и полное практическое занятие из представленных в данной книге. Описываются методы объектно-ориентированного программирования и реализации на языке C++ простой пакетной системы, реализующей расчет зарплаты.
- Часть IV. *Упаковка программы расчета зарплаты*. Эта часть начинается с описания *принципов объектно-ориентированного проектирования пакетов*. Затем эти принципы иллюстрируются на примере инкрементной упаковки классов, приведенном в предыдущей части.
- Часть V. *Практическое занятие: моделирование метеостанции*. В этой части рассматривается одно из практических занятий, которое изначально планировалось поместить в книге Буча. В данной части идет речь о компании, принявшей важное деловое решение, и о том, каким образом команда разработчиков ПО на языке Java отреагировала на такое решение. Как обычно, часть начинается описанием шаблонов проектирования, используемых в процессе разработки ПО, а завершается — рассмотрением методов разработки и реализации проекта.
- Часть VI. *Практическое занятие: система ETS*. Приводится описание реального проекта, в котором принимал участие автор. Речь идет об автоматизированной системе тестирования, используемой для подсчета количества баллов, набранных на экзаменах, и внедренной в Национальном совете регистрационной коллегии по архитектуре (National Council of Architectural Registration Boards).
- Приложения, описывающие стандарты UML-записи. Здесь вы найдете краткое описание UML-языка, построенное на рассмотрении небольших практических примеров.
- Приложения, посвященные рассмотрению различных вопросов. Здесь вы найдете материалы, “разбавляющие” основную тематику данной книги.

## **Для кого предназначена эта книга**

### **Если вы разработчик. . .**

Прочитайте эту книгу “от корки до корки”. Она была написана в первую очередь для разработчиков и содержит информацию, которая необходима вам для создания ПО с помощью методов быстрой разработки. Прочитав эту книгу полностью, вы получите представление о методах, принципах и шаблонах, а также ознакомитесь с конкретными примерами, которые связывают их в одно целое. Воспользовавшись всеми полученными знаниями, вы сможете *довести* работу над своими проектами *до конца*.

---

## **Если вы менеджер или бизнес-аналитик. . .**

Прочтите часть I, “*Быстрая разработка ПО*”. Главы этой части содержат подробное описание принципов и методов быстрой разработки, требования к планированию, тестированию, рефакторингу и программированию. Вы получите инструкции о том, каким образом следует организовывать команды разработчиков и управлять проектами. Этот материал поможет вам *довести* работу над своими проектами *до конца*.

## **Если вы хотите изучить UML-язык. . .**

Сначала прочтите приложение А. Затем обратитесь к материалу приложения Б. После этого прочтите все главы из части III. В результате вы получите представление об основах синтаксиса и использования UML-языка. Вы сможете осуществлять преобразование UML-записи в программный код на языках Java и C++.

## **Если вы хотите изучить шаблоны проектирования. . .**

Чтобы найти определенный шаблон, воспользуйтесь “Перечнем шаблонов проектирования”.

Чтобы ознакомиться с шаблонами в общих чертах, прочтите часть II, чтобы получить представление о принципах проектирования, а затем прочитайте часть III, часть IV, часть V и часть VI. В материале этих частей определены все шаблоны и продемонстрирован принцип их использования в типичных ситуациях.

## **Если вы хотите изучить принципы объектно-ориентированного программирования. . .**

Прочитайте часть II, часть III и часть IV. Здесь приводится описание принципов объектно-ориентированного проектирования, а также продемонстрирован принцип их использования.

## **Если вы хотите изучить методы быстрой разработки. . .**

Прочитайте часть II. Здесь описываются принципы быстрой разработки, включая требования, планирование, тестирование, рефакторинг и программирование.

## **Если хочется повеселиться. . .**

Прочтите приложение В.

## Благодарности

Мы выражаем сердечную благодарность следующим людям:

Лоуэллу Линдстрому (Lowell Lindstrom), Брайану Баттону (Brian Button), Эрику Миду (Erik Meade), Майку Хиллу (Mike Hill), Майклу Фезерсу (Michael Feathers), Джиму Ньюкирку (Jim Newkirk), Мику Мартину (Micah Martin), Анжелике Товенин Мартин (Angelique Thouvenin Martin), Сьюзанне Россо (Susan Rosso), Талишу Джефферсону (Talisha Jefferson), Рону Джеффрису (Ron Jeffries), Кену Беку (Kent Beck), Джеффу Лангр (Jeff Langr), Дэвиду Фарберу (David Farber), Бобу Коссу (Bob Koss), Джеймсу Греннингу (James Grenning), Лансу Уэлтеру (Lance Welter), Паскалю Рою (Pascal Roy), Мартину Фаулеру (Martin Fowler), Джону Гудсену (John Goodsen), Алану Апту (Alan Apt), Полу Ходжеттсу (Paul Hodgetts), Филу Маркграфу (Phil Markgraf), Питу Мак-Брину (Pete McBreen), Х.С. Ламэну (H. S. Lahman), Дэйву Харрису (Dave Harris), Джеймсу Кэйнзу (James Kanze), Марку Вэбстеру (Mark Webster), Крису Бигей (Chris Biegay), Алану Френсису (Alan Francis), Фрэн Даниэл (Fran Daniele), Патрику Линдеру (Patrick Lindner), Джейку Варду (Jake Warde), Ами Тодд (Amy Todd), Лауре Стил (Laura Steele), Уильяму Пьетр (William Pietr), Камилле Трентакоста (Camille Trentacoste), Винсу О'Брайану (Vince O'Brien), Грегори Дуллесу (Gregory Dulles), Линде Кастилло (Lynda Castillo), Крэйгу Лармэну (Craig Larman), Тиму Оттингеру (Tim Ottinger), Крису Лопесу (Chris Lopez), Филу Гудвину (Phil Goodwin), Чарльзу Тоулэнду (Charles Toland), Роберту Эвансу (Robert Evans), Джону Роту (John Roth), Дебби Утьев (Debbie Utiey), Джону Брюэр (John Brewer), Руссу Рутер (Russ Ruter), Дэвиду Выдре (David Vydra), Лану Смиты (Lan Smith), Эрику Эвансу (Eric Evans), а также всем членам группы "The Silicon Valley Patterns group", Питу Бриттингэму (Pete Brittingham), Грэхэму Перкинсу (Graham Perkins), Флипу (Phlip) и Ричарду Мак-Дональду (Richard MacDonald).

Книгу рецензировали:

Пит Мак-Брин (Pete McBreen)/McBreen Consulting

Стефан Дж. Меллор (Stephen J. Mellor)/Projtech.com

Брайан Баттон (Brian Button)/Object Mentor Inc.

Бьерн Страуструп (Bjarne Stroustrup)/AT&T Research

Микей Мартин (Micah Martin)/Object Mentor Inc.

Джеймс Греннинг (James Grenning)/Object Mentor Inc.

Особая благодарность выражается Гради Бучу и Полу Бекеру за то, что они разрешили мне включить в эту книгу главы, которые изначально предназначались для 3-го издания книги Гради *Объектно-ориентированный анализ и проектирование приложений*.

Особая благодарность выражается Джеку Ривзу за то, что он любезно разрешил мне воспользоваться материалом его статьи "Что такое проектирование?" (What is Design?).

Особую благодарность также выражаю Эриху Гамма за написание предисловия к этой книге. Я надеюсь, Эрих, что в этот раз шрифты для книги подобраны лучше!

Превосходные иллюстрации, приведенные в начале каждой главы, принадлежат Дженнифер Конке (Jennifer Kohnke). Декоративные иллюстрации, приведенные в главах, — это восхитительная работа Анжелы Дон Мартин Брукс (Angela Dawn Martin Brooks), моей дочери, которая составляет одну из наибольших радостей моей жизни.

## **Источники**

Все примеры исходного кода, приведенные в этой книге, находятся на Web-узле [www.objectmentor.com/PPP](http://www.objectmentor.com/PPP).