

Глава 1

Введение

Если вы не знакомы с книгой *Design Patterns: Elements of Reusable Object-Oriented Software* [GoF95], то ее следует прочесть, прежде чем продолжить чтение данной книги. Аналогично нужно поступить и в том случае, если вы в свое время читали эту книгу, но недостаточно внимательно *изучили* ее.

Итак, далее я предполагаю, что вы не относитесь ни к одной из упомянутых выше категорий читателей, т.е. вам кое-что известно о шаблонах вообще и о нашем наборе из 23-х шаблонов проектирования в частности. Только в этом случае можно извлечь пользу из данной книги, так как она дополняет, обновляет и уточняет материалы, изложенные в *Design Patterns*. Если же вы не знакомы с вышеупомянутыми 23 шаблонами так называемой “банды четырех” (*Gang of Four – GoF*), то вам будет крайне сложно понять предлагаемый здесь материал. Лучше всего, если *Design Patterns* все время будет у вас под рукой при чтении данной книги. Наконец, я предполагаю, что вы знакомы с языком C++, тем более, что аналогичное предположение содержится и в *Design Patterns*.



А теперь в качестве проверки попробуйте с помощью 25 или меньшего количества слов описать назначение шаблона COMPOSITE.

Если вам без труда удалось выполнить это упражнение, то при чтении книги у вас не возникнет проблем. Если вы знаете о назначении данного шаблона, но затрудняетесь описать его словами, не стоит беспокоиться — эта книга предназначена и для вас.

Если же вы исписали целый лист, то вам стоит воспользоваться советом в самом начале параграфа: отложите эту книгу, возьмите *Design Patterns* и прочитайте до конца раздел, посвященный реализации данного шаблона. Необходимо также прочитать соответствующие разделы, касающиеся других шаблонов. После этого вы будете достаточно подготовлены, чтобы извлечь пользу из чтения данной книги.

Почему я выбрал такое название для своей книги? Все удачные названия уже были задействованы, и первоначально я просто написал *Pattern Hatching* (в книге рассматриваются дополнительные аспекты использования шаблонов, а одно из значений слова *hatching* — штриховка). Однако вскоре я обнаружил, что это название достаточно точно отражает мое понимание темы. Другое значение слова *hatching* (выведение птенцов) подразумевает, что речь идет не о создании чего-либо нового, а скорее о развитии имеющихся зачатков. Можно сказать, что наша книга *Design Patterns* — это коробка с яйцами, из которых, возможно, возникнет новая жизнь¹.

В своей книге я не буду просто пересказывать *Design Patterns*. Я хочу оживить ее концепции, показать, как они применяются на практике, и сделать их наиболее полезными

¹ В дальнейшем мы не будем злоупотреблять этой аналогией.

для читателя. Вы научитесь определять, какие шаблоны следует, а какие не следует применять в зависимости от конкретных обстоятельств, получите более глубокое представление о некоторых из наших исходных шаблонов и ознакомитесь с процессом создания новых шаблонов. Книга изобилует примерами. Некоторые из них — это проверенные временем конструкции, другие — так называемые “полуфабрикаты”. Еще одна категория примеров — чисто иллюстративные — представляет собой полностью “бумажные” проекты, которые, скорее всего, не вынесут столкновения с жестокой реальностью, но могут содержать зерна более устойчивых конструкций.

Предлагаю все это вашему вниманию, искренне желая повысить ваши способности в сфере проектирования, уровень знаний о шаблонах, а также расширить взгляд на разработку программного обеспечения в целом. В книге собран мой опыт использования шаблонов. Надеюсь, что он пригодится и вам.

Десять основных заблуждений

Вместе с тем шумом, который поднялся вокруг шаблонов проектирования в настоящее время, возникло немало недоразумений, опасений и неверной информации. Отчасти это явилось отражением новизны данного направления для большинства разработчиков программного обеспечения (хотя, строго говоря, это направление не такое уж новое). Оно быстро развивается, вследствие чего ощущается недостаток фактов. Наконец, несмотря на то, что о шаблонах написано немало книг [BMR+96, Coplien96, CS95, GoF95, MRB98, VCK96], создатели шаблонов действительно заслужили определенные обвинения со стороны недостаточно подготовленной аудитории.

Поэтому я чувствую себя обязанным развеять наиболее распространенные заблуждения, которые мне приходилось слышать настолько часто, что их можно считать своего рода шаблонами. Я даже забавлялся, используя форму шаблона для их описания... пока меня не осенила мысль, что сводить все к шаблонам — тоже заблуждение!

Внимательно изучив мнения различных людей о шаблонах на протяжении ряда лет, я пришел к выводу, что основные заблуждения можно отнести к одной из трех категорий: заблуждения относительно того, *что* собой представляют шаблоны; заблуждения относительно того, что они *могут делать*, и заблуждения относительно тех, кто эти шаблоны предлагает. Каждое заблуждение из моей “лучшей десятки” также попадает в одну из этих категорий. Начнем с рассмотрения заблуждений, относящихся к первой категории.

Заблуждение 1:

“Шаблон — это решение определенной проблемы в определенных обстоятельствах.”

Данное определение принадлежит Кристоферу Александру (Christopher Alexander, [AIS+77]), поэтому критические высказывания о нем могут некоторым показаться ересью. Однако простой пример поможет прояснить, что я имею в виду.

<i>Проблема:</i>	Как получить выигрыш по лотерейному билету до истечения срока его предъявления?
<i>Ситуация:</i>	Собака съела билет за час до истечения указанного срока.
<i>Решение:</i>	Вскрыть собаку, извлечь билет и бежать к ближайшему пункту выплаты выигрышей.

Это и есть решение проблемы в определенных обстоятельствах. Но это не шаблон. Чего же не хватает? По крайней мере трех составляющих:

1. Повторяемости, которая делает решение применимым в ситуациях, отличных от рассматриваемой.
2. Обучения, которое дает понимание, как данное решение связано с конкретным вариантом проблемы. (В реальных шаблонах обучающая часть состоит главным образом в описании и выделении движущих сил и/или последствий его применения.)
3. Имени, с помощью которого можно ссылаться на данный шаблон.

Как свидетельствует продолжающаяся дискуссия в Интернет-форуме (*patterns-discussion@cs.uiuc.edu*²), до сих пор не удалось дать исчерпывающее определение шаблона. Дело усложняется тем, что шаблон — это и конкретная сущность, и описание аналогичных сущностей. Один из способов различать их состоит в том, чтобы использовать термин *шаблон* по отношению к описанию, а термин *экземпляр шаблона* — по отношению к конкретной реализации этого шаблона.

Разработка определений для терминов — неблагодарный труд, поскольку определение, понятное одной аудитории (скажем, программистам), может показаться полностью бессмысленным другим специалистам. Я не ставлю перед собой задачу дать исчерпывающее определение шаблона. Достаточно сказать, что в любом определении, перечисляющем составные части шаблона, наряду с проблемой, решением и контекстом необходимо сказать о повторяемости, обучении и именовании.

Заблуждение 2:

“Шаблоны — это просто жаргон, правила, приемы программирования, структуры данных...”

Я называю это “отторжением вследствие недооценки”. Стремление свести нечто непознанное к известному является совершенно естественным, тем более, если вы не заинтересованы в его исследовании. Кроме того, люди слишком часто помещают старое вино в новую упаковку и называют это новшеством, поэтому настороженное отношение ко всяким новинкам вполне оправдано.

В данном случае отторжение не является следствием опыта. Зачастую оно основано на поверхностном знакомстве и изрядной доле цинизма. В действительности, нет ничего полностью нового; человек создавал шаблоны в своей голове с тех пор, как у него появилась голова. Новое — в том, что мы предложили именовать шаблоны и описывать их.

Относительно приведенных высказываний. Действительно, существует определенный жаргон, связанный с шаблонами, в который входят термины “шаблон”, “движущие силы”, используемое Александером (Alexander) “качество без названия” и т.д. Но было бы неверно сводить шаблоны проектирования к жаргону! По сравнению с остальными областями информатики в данной области не так уж много новых терминов. И это весьма симптоматично. Хорошему шаблону внутренне присуща доступность для тех, кому он предназначен, в нем могут использоваться термины целевой области, но вряд ли возникнет необходимость в специальной терминологии из области шаблонов.

Шаблоны не являются правилами, которые можно бездумно применять (об этом свидетельствует наличие обучающего компонента), они также не сводятся к приемам

² Чтобы подписаться, нужно послать электронное сообщение по адресу *patterns-discussion-request@cs.uiuc.edu*, указав единственное слово “subscribe” (без кавычек!) в качестве темы сообщения.

программирования, несмотря на то, что “идиоматическое” направление данной дисциплины занимается зависящими от языка программирования шаблонами. “Приемы” — также слишком узкое определение, поскольку в нем преувеличивается значение решения за счет недооценки проблемы, контекста, обучения и именования.

Вы, без сомнения, слышали о трех стадиях принятия любого новшества: сначала оно отрицается, как не содержащее ничего ценного, затем — считается нежизнеспособным и, наконец, становится очевидным и тривиальным — “Мы всегда так делали”. Шаблоны к настоящему времени еще находятся на первой стадии своего развития.

Заблуждение 3:

“Достаточно посмотреть на один из них, чтобы получить представление об остальных.”

Стричь всех под одну гребенку неправильно, однако по отношению к шаблонам это неправильно вдвойне. Существует широкий диапазон всевозможных шаблонов, которые различаются областью применения, содержимым, масштабом, стилем и качеством — достаточно пролистать одну из книг серии *Pattern Languages of Program Design* [CS95, MRB98, VCK96], чтобы убедиться в этом. Шаблоны столь же различны, как и люди, которые их пишут, а может быть даже больше. Такие авторы, как Элистер Кокборн (Alistair Cockburn), Джим Коплин (Jim Coplien), Нэйл Харрисон (Neil Harrison) и Ральф Джонсон (Ralph Johnson) смогли выйти за рамки написания шаблонов определенного стиля для отдельных предметных областей, поэтому было бы неправильно судить о шаблонах в целом, рассмотрев лишь несколько примеров.

Заблуждение 4:

“Для успешного применения шаблонов необходимы соответствующие инструментальные средства или методологическая поддержка.”

Последние пять лет я занимался созданием и применением шаблонов проектирования, помогал использовать их другим, а также участвовал в подготовке по меньшей мере одного основанного на шаблонах инструментального средства [BFV+96], поэтому могу со всей ответственностью заявить, что выигрыш при применении шаблонов получается в основном за счет самих шаблонов, т.е. они не нуждаются ни в какой специальной поддержке.

Обычно я указываю четыре основных преимущества шаблонов:

1. Они позволяют суммировать опыт экспертов и сделать его доступным рядовым разработчикам.
2. Имена шаблонов образуют своего рода словарь, который позволяет разработчикам лучше понимать друг друга.
3. Если в документации системы указано, какие шаблоны в ней используются, это позволяет читателю быстрее понять систему.
4. Шаблоны упрощают реструктуризацию системы независимо от того, использовались ли шаблоны при ее проектировании.

Долгое время я считал главным первое из названных преимуществ. Теперь я понимаю, что второе не менее важно. Представьте, сколькими байтами информации обмениваются разработчики (не важно, в вербальной или электронной форме) в процессе работы над проектом! Я думаю, что это мегабайты, если не гигабайты. (У меня хранится несколько десятков мегабайт электронной корреспонденции, которой обменивались участники “банды четырех” в процессе написания книги *Design Patterns*. По приблизительным оценкам, такая работа эквивалентна среднему проекту разработки программ

ного обеспечения.) При такой интенсивности обмена информацией все, что может сделать этот обмен хоть немного более эффективным, приведет к существенной экономии времени. Таким образом, шаблоны проектирования способствуют повышению интенсивности человеческого общения. Моя оценка преимуществ 3 и 4 также повышается со временем, особенно по мере роста масштабов проектов и увеличения продолжительности жизни их программного обеспечения.

Короче говоря, шаблоны проектирования — это главным образом пища для ума, а не материал для инструментального средства. Выгода от методологической или инструментальной поддержки также возможна, но это только розочки на торте, а не сам торт, и даже не слой крема на нем.



Рассмотренные до сих пор заблуждения касались того, что представляют собой шаблоны. Теперь перейдем к заблуждениям относительно возможностей шаблонов. Речь может идти как о недооценке, так и о переоценке этих возможностей.

Заблуждение 5:

“Шаблоны гарантируют возможность повторного использования программного обеспечения, повышение производительности, отсутствие разногласий и т.п.”

Это заблуждение очевидно, поскольку шаблоны *вообще ничего не гарантируют*. Можно говорить только о вероятности получения неких преимуществ. Шаблоны никоим образом не могут заменить человека в творческом процессе. Они просто предоставляют дополнительные возможности недостаточно опытному или малоинициативному, но в целом толковому разработчику.

Хороший шаблон вызывает восторженную реакцию у пользователей, но это может произойти только тогда, когда данный шаблон встречает отклик читателя. Если же этого не происходит, шаблон напоминает говорящее дерево в лесу, где его никто не слышит. Как бы хорошо ни был написан шаблон, от него в этом случае не будет толку.

Шаблоны существенно отличаются от других элементов арсенала разработчика. Не стоит слишком полагаться на них. Меньше пообещать и больше сделать — вот лучшая защита от неоправданного оптимизма и последующего разочарования.

Заблуждение 6:

“Шаблоны порождают готовые архитектурные решения.”

Это заблуждение аналогично предыдущему, только менее категорично по форме.

Порождающие аспекты шаблонов периодически обсуждаются в соответствующих Internet-форумах. Насколько я понимаю, под порождающей способностью подразумевается способность шаблона создавать результирующее поведение, т.е. некий шаблон помогает решать проблемы, для которых он изначально не был предназначен. В некоторых работах утверждается, что истинная порождающая способность носит объективный характер, и нужное поведение формируется практически независимо от желания пользователя шаблона.

Порождающая способность в основном заключена в частях шаблона, предназначенных для обучения — там, где рассматриваются движущие силы или последствия. Это особенно полезно при определении и уточнении архитектуры. Но сами по себе шаблоны ничего не создают, это делают люди и только в том случае, если они достаточно квалифицированы, а используемые ими шаблоны достаточно эффективны. Более того, вряд ли шаблоны покрывают все аспекты архитектуры. Любой нетривиальный проект содержит множество аспектов проектирования, для которых не существует шаблонов. Возможно, эти аспекты не являются стандартными или повторяющимися, или же шаб-

лоны для них еще только предстоит написать. В любом случае, вам придется проявить свои творческие способности и заполнить пробелы между шаблонами самостоятельно.

Заблуждение 7:

“Шаблоны предназначены для объектно-ориентированного проектирования и реализации.”

Это заблуждение относится к разряду тех, где возможности шаблонов чересчур ограничиваются.

Шаблоны не представляют собой никакой ценности, если они не содержат практического опыта. Какой опыт будет заключен в шаблоне, зависит от его разработчика. Безусловно, в объектно-ориентированном проектировании существует огромный опыт, но не менее ценные наработки имеются в проектировании, не основанном на объектно-ориентированной парадигме, и не только в проектировании, но и в анализе, сопровождении, тестировании, документировании, разработке организационной структуры и т.д. В настоящее время появляются шаблоны и в этих областях. Уже вышло по крайней мере две книги, посвященные шаблонам анализа [Fowler97, Hay96], а каждая конференция по языкам шаблонов программ (Pattern Languages of Programs — PLoP) открывает все новые виды шаблонов. (Например, на конференции 1996 года рассматривались шаблоны для создания музыкальных композиций!)

В каждом заблуждении содержится рациональное зерно. В данном случае, если проанализировать используемые формы шаблонов, можно выделить вариации двух основных стилей: строго структурированного стиля, предложенного “бандой четырех” в книге *Design Patterns*, и почти беллетристического стиля Кристофера Александера (Christopher Alexander), практически лишенного структуры и напоминающего простой комментарий. Занявшись написанием шаблонов в сферах, отличных от объектно-ориентированного проектирования, я понял, насколько предложенный “бандой четырех” стиль привязан к своей области. Эта схема совершенно не срабатывает в других областях, где я пытался ее применить. Что такое структурная диаграмма для идиомы C++? Реализационные компромиссы в шаблоне музыкальной композиции? Взаимодействия в шаблоне для составления описаний?

Очевидно, что определенный формат не может охватить все разнообразие шаблонов. Действительно всеобъемлющим является понятие шаблона как средства сбора и передачи опыта в любой сфере деятельности.

Заблуждение 8:

“Нет доказательств того, что шаблоны хоть кому-нибудь помогают.”

В прошлом это заблуждение имело под собой основания, но сейчас ситуация изменилась. О преимуществах, полученных при использовании шаблонов, сообщается в журналах (в частности, в журнале *Software – Practice and Experience* [Kotula96]) и на конференциях (OOPSLA [HJE95, Schmid95] и ICSE [BCC+96]). Даг Шмидт (Doug Schmidt) указал на определенные преимущества от применения шаблонов при обучении информатике новичков и старшекурсников [PD96]. Хотя большинство доказательств носит качественный характер, мне известна по крайней мере одна группа, которая проводит эксперименты с целью получить количественные результаты [Pretchelt97, PUS97].

С течением времени станут более понятны преимущества и опасности, возникающие при использовании шаблонов. Хотя первые результаты весьма обнадеживающие, для более тщательной оценки необходим дополнительный опыт. Тем не менее, было бы неправильно отказываться от шаблонов на том основании, что выгоды от их использования не измерены количественно.



От ошибочных суждений относительно возможностей шаблонов перейдем к рассмотрению двух последних заблуждений, которые касаются не самих шаблонов, а тех, кто эти шаблоны создает и поддерживает идеи их использования.

Заблуждение 9:

“Сообщество, поддерживающее идеи применения шаблонов, представляет собой узкий элитарный круг.”

Интересно узнать, откуда возникло это заблуждение, поскольку на самом деле для сообщества приверженцев идеи использования шаблонов характерна максимальная открытость. Достаточно посмотреть на данные об участниках конференций PLoP: это люди со всего мира, из больших корпораций и крохотных фирм; аналитики, проектировщики и разработчики; студенты и профессора; признанные авторитеты и неоперившиеся юнцы. Я был крайне удивлен, когда узнал, что некоторые регулярные участники даже не являются специалистами в области информатики! Сообщество постоянно находится в движении, состав участников конференции год от года заметно меняется.

При таком обилии публикаций некоторые могут удивиться определенному недостатку академичности. Дело в том, что большинство участников конференций PLoP — практики. По-видимому, так и должно быть. Никто из известных разработчиков первых шаблонов программного обеспечения — включая Кента Бека (Kent Beck), Питера Коуда (Peter Coad) и Уорда Каннингэма (Ward Cunningham) — не был выходцем из академических кругов. Только один из “банды четырех” — Ральф (Ralph Johnson) — академический ученый, и он самый лучший прикладник из всех ученых, которых я знаю. Таким образом, сама природа сообщества приверженцев шаблонов опровергает все подозрения относительно его однородности и элитарности.

Заблуждение 10:

“Сообщество приверженцев шаблонов преследует корыстные цели.”

Мне неоднократно приходилось слышать обвинения в том, что шаблоны служат источником наживы для тех, кто пишет о них книги. Иногда даже утверждают, что “движение” приверженцев шаблонов имеет низменные цели.

Какая чушь!

Все члены “банды четырех” не менее других были удивлены реакцией читателей на появление книги *Design Patterns*. Никто из нас не был готов к ажиотажу, вызванному ее представлением на OOPSLA '94, даже издатели были поражены таким высоким спросом. При работе над книгой мы старались сделать ее как можно лучше. Мы были слишком заняты содержанием, чтобы задумываться о том, как эта книга будет продаваться.

Теперь, когда термин “шаблон” стал популярным, кое-кто пытается использовать это слово не совсем в альтруистических целях. Но если внимательно читать работы ведущих специалистов в области шаблонов, можно заметить, что всех авторов объединяет стремление собрать и передать другим приобретенный опыт, лучшие практические наработки, имеющиеся преимущества — результат многих лет работы — и не просто описать их, но и *приобщить* к ним всех читателей.

Именно желание облегчить жизнь разработчиков движет настоящими авторами работ по шаблонам. Другие мотивы пагубны и ведут к неправильным представлениям о шаблонах.

Наблюдения

Как правило, люди реагируют на шаблоны проектирования одним из двух способов, которые я попытаюсь проиллюстрировать с помощью аналогии.

Представьте себе электронщика-любителя, который, не имея формального образования, сумел за несколько лет разработать и сделать уйму полезных приспособлений: любительское радио, счетчик Гейгера, домашнюю сигнализацию и т.д. И вот однажды любитель решает, что пора добиться официального признания своего таланта, пройти курс обучения и получить специальность электронщика. Приступив к учебе, он поражается тому, насколько ему знаком материал. Конечно, ему знакома не терминология или способ изложения, а лежащие в основе концепции. Электронщик-любитель видит названия и усовершенствования тех вещей, которыми он пользовался годами. Для него это просто одно откровение за другим!

Теперь представим себе новичка, который посещает те же занятия и изучает тот же материал. Он много знает о роликовых коньках, но не имеет представления об электронике. Ему будет очень сложно усваивать предлагаемый материал не потому, что он тупой, а потому, что этот предмет ему совершенно незнаком. Новичку потребуется гораздо больше времени, чтобы понять и оценить материал. Но он все-таки может это сделать, если проявит настойчивость и упорно поработает.

Если вы можете сравнить себя с любителем шаблонов — тем лучше для вас. Но если вы ощущаете себя скорее новичком, не стоит расстраиваться: те усилия, которые вы потратите на изучение хороших шаблонов, будут окупаться всякий раз, когда вы будете применять эти шаблоны в своих проектах. Я вам это обещаю.

Возможно, электроника с ее чисто техническим содержанием — не лучшая аналогия. Альфред Норт Уайтхед (Alfred North Whitehead) сказал в 1943 году:

“Искусство — это наложение шаблона на опыт и наше эстетическое наслаждение от узнавания этого шаблона...”

Хотя эта фраза была сказана совсем по другому поводу, она может иметь к нашему случаю самое непосредственное отношение.