

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 легкочитаемое 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.

Я рекомендую книгу *Java Concurrency in Practice* для практического знакомства с многопоточностью.

Другая тема, не затронутая в данной книге, — это понятие процесса программирования. Советы относительно “взаимодействия через код” применимы ко времени, когда написание программы подходит к концу после длинного цикла, или к моменту написания теста на ошибки. Всегда лучше иметь недорогое программное обеспечение, чем какие-то ни было социологические исследования, следуя которым оно было написано.

Я также вкратце очертил границы применения Java. Я вынужден быть консервативным в выборе технологии, поскольку я слишком часто терпел фиаско, используя новые возможности на всю катушку (это хорошая стратегия обучения, однако слишком рискованная для большинства разработок). Так что вы найдете только прозаичное подмножество Java. Если вы нацелены на изучение последних нововведений в Java, обратитесь к другим источникам.

Краткое содержание

Как видно на рис. 1.1, книга разделена на семь основных частей.

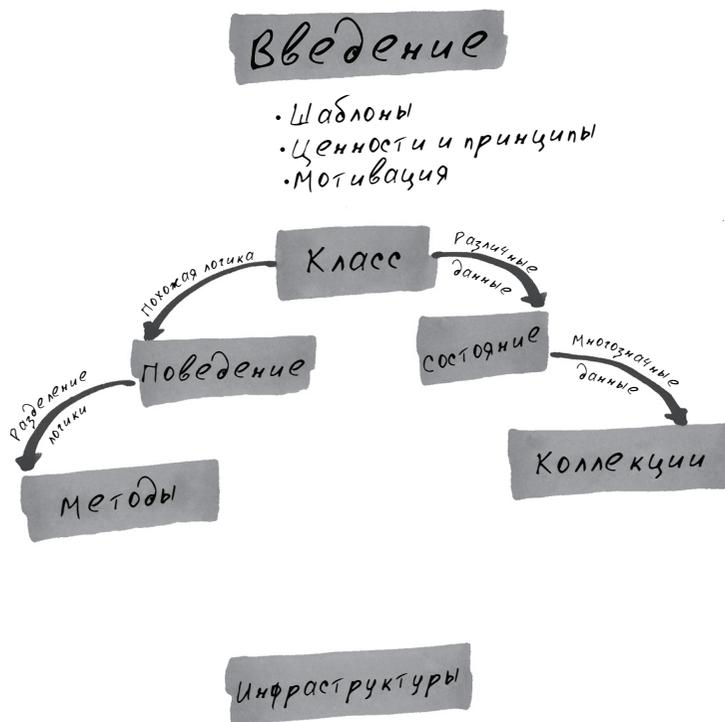


Рис. 1.1. Структура книги

Вступление. Эти короткие главы описывают важность и ценность коммуникации через код и философию, стоящую за шаблонами.

Класс. Шаблоны, описывающие, как и почему вы должны создавать классы, и каким образом в них закодирована логика.

Состояние. Шаблоны для сохранения и получения состояния.

Поведение. Шаблоны, представляющие логику, особенно альтернативные пути.

Метод. Шаблоны для написания методов, напоминающие, что хотелось бы увидеть читателям в результате вашего выбора разбиения на методы и имена.

Коллекции. Шаблоны для выбора и использования коллекций.

Развертывание инфраструктур. Вариации на тему использования шаблонов при построении инфраструктур вместо приложений.

Итак...

...приступим. Если вы читаете книгу с самого начала, всего лишь переверните страницу (я думаю, вы уже вычислили это действие сами). Если же вы хотите просмотреть сами шаблоны, начните с главы 5. Счастливой реализации.