ОГЛАВЛЕНИЕ

Предисловие

Совершенствование навыков разработки программного обеспечения включает в себя изучение целого пула разрозненных концепций. Если вы — начинающий разработчик или даже имеете определенный опыт, это может показаться непреодолимо сложной задачей. Стоит ли потратить время на изучение основных положений объектно-ориентированного мира, таких как принципы SOLID, шаблоны проектирования или разработка через тестирование? Нужно ли применять такие набирающие популярность технологии, как, например, функциональное программирование?

Часто бывает, что, даже, изучив какие-то темы в отдельности, потом трудно понять, как связать их друг с другом. В каком случае необходимо воспользоваться в проекте функциональным программированием? Когда стоит задуматься о тестировании? Как узнать, в какой момент следует внедрять или совершенствовать упомянутые методы? Нужно ли прочитать книгу по каждой из этих тем, затем изучить посвященные им блоги, или посмотреть видео, чтобы разобраться, как все это соединить воедино? И вообще, с чего начать?

Не переживайте. Данная книга создана для того, чтобы помочь вам. Вы получите ответы на свои вопросы в процессе интегрированного, проектно-ориентированного обучения. Вы изучите основные темы, которые помогут вам стать эффективным разработчиком. Кроме того, мы покажем, как объединить полученные знания и применить в больших проектах.

Почему мы написали эту книгу

За многие годы обучения программированию мы накопили богатый опыт. Мы оба написали книги по Java 8 и проводим курсы по профессиональной разработке программного обеспечения. В процессе мы были признаны Java-

чемпионами и стали востребованными спикерами на различных международных конференциях.

За все это время мы поняли, что многим разработчикам достаточно лишь введения или краткого описания некоторых основных тем. Шаблоны проектирования, функциональное программирование, принципы SOLID и тестирование — это методы, которые хороши сами по себе, но редко можно встретить демонстрацию того, как соединить их вместе. Иногда программисты перестают развивать свои навыки лишь потому, что не могут определиться, что же изучать дальше. Мы хотим не просто дать вам определенные базовые навыки, наша цель — сделать обучение легким, интересным и даже веселым.

Подход, ориентированный на разработчика

Данная книга предоставляет возможность познакомиться с ориентированным на разработчика подходом. В ней вы найдете множество примеров кода. В каждой теме рассматриваются реальные программы и проекты. Все они даются в полном объеме, так что на каждом этапе вы можете проверять код в своей *интегрированной среде разработки* (*Integrated Development Environment*, *IDE*) и запускать программы, чтобы оценить их в действии.

Есть и другая распространенная проблема технической литературы. Зачастую она написана в формальном стиле учебника, далеком от живого общения. Однако мы решили придерживаться в книге разговорного стиля, чтобы помочь вам почувствовать себя членом команды, а не «подопечным» или учеником.

Что в этой книге?

Каждая глава книги строится вокруг определенного проекта. По завершении главы вы сможете самостоятельно написать этот проект, если, конечно, хорошо изучите весь материал. При этом уровень сложности будет постепенно возрастать — от простых консольных программ до полноценных приложений.

Пректно-ориентированный подход, лежащий в основе данной книги, имеет ряд преимуществ. Прежде всего, он позволит вам увидеть, как разнообразные методики программирования работают в единой связке. Когда, ближе

к концу книги, мы начнем рассматривать функциональное программирование, речь пойдет не просто о наборе абстрактных вычислительных операций. Они будут нужны для получения определенных результатов в конкретном проекте. Такой подход выгодно отличает данное руководство от учебных материалов, демонстрирующих действительно хорошие методы, но в отрыве от реальных задач, из-за чего разработчики часто применяют их совершенно неуместно.

Проектно-ориентированный подход позволит вам на каждом этапе работать с реальными примерами. В классических учебных пособиях довольно часто можно встретить в коде метапеременные `Foo` (для обозначения классов) и `bar` (для обозначения методов). Наши примеры соответствуют реальным задачам и демонстрируют подход к решению реальных проблем, похожих на те, с которыми, возможно, вы столкнетесь в повседневной работе.

И наконец, проектно-ориентированный подход делает учебу гораздо интереснее. Каждая глава — это новый проект и новая возможность совершить для себя открытие. Мы хотим, чтобы вы дочитали все до конца, и искренне рады каждому читателю. Все главы книги начинаются с той или иной задачи, которую нужно решить. Затем рассматривается решение, а в конце подводятся итоги по пройденному материалу. Мы четко определяем задачу в начале и в конце каждой главы, чтобы убедиться, что вы хорошо ее понимаете.

Для кого эта книга?

Мы уверены, что разработчики из самых разных областей найдут для себя что-то полезное и интересное в этой книге. Среди них будут и те, кто сможет использовать ее максимально эффективно.

На наш взгляд, основная аудитория книги — это начинающие разработчики, только окончившие университет, либо имеющие за плечами пару лет опыта работы. Мы познакомим вас с основными темами, которые, как мы считаем, пригодятся вам в работе. Для взаимодействия с данной книгой не нужно иметь ученую степень, но для лучшего усвоения материала необходимо обладать базовыми знаниями в программировании. К примеру, здесь мы не будем объяснять, что такое оператор `if` или циклы.

Для того чтобы приступить, вам необязательно иметь глубокие познания в объектно-ориентированном или функциональном программировании.

Например, в главе 2 от вас не потребуется ничего сверх знания о том, что такое класс, и умения работать с базовыми типами (такими как `List<String>`). Только самые основы.

Данная книга может также представлять интерес и для разработчиков, пришедших в Java из других языков программирования, таких как C#, C++ или Python. Она поможет вам быстро освоить необходимые конструкции языка, принципы, методы и ключевые моменты, необходимые для написания хорошего кода на Java.

Если вы являетесь более опытным Java-разработчиком, то вполне можете пропустить главу 2, чтобы не повторять базовый материал, который вы и так уже знаете. Однако начиная с главы 3 в книге будут рассматриваться многие концепции и подходы, полезные для всех разработчиков.

Мы считаем, что обучение может быть одним из самых интересных этапов разработки программного обеспечения, и верим, что во время чтения книги вы с нами согласитесь. Надеемся, вам понравится это путешествие.

Условные обозначения, используемые в книге

В книге используются следующие типографские условные обозначения:

Курсив

Обозначает новые понятия и термины, URL ссылки, адреса электронной почты, имена файлов, расширения файлов.

`Моноширинный шрифт`

Используется в листингах программ, а также в тексте книги для обозначения элементов программ, таких как имена переменных или функций, базы данных, типы данных, переменные среды, операторы и ключевые слова.

`Моноширинный полужирный`

Обозначает команды или другой текст, который должен быть введен пользователем.

`Моноширинный курсивный`

Обозначает текст, который должен быть заменен пользовательскими значениями или значениями, уточняемыми в контексте.

 Данный символ обозначает примечание.

Использование примеров кода

Дополнительные материалы (такие как примеры программ, упражнения и т. д.) доступны для скачивания на *https://github.com/Iteratr-Learning/Real-World-Software-Development*.

Данная книга создана для того, чтобы помочь вам в работе. Поэтому, если вам подходят некоторые примеры программ, вы можете использовать их в своих приложениях и в документации. Вам не нужно связываться с нами и получать для этого специальное разрешение, оно может понадобиться только при использовании внушительного объема кода. Например, если в своей программе вы используете несколько кусочков кода из этой книги, разрешение вам не нужно. Использование большого объема примеров кода в документации к вашему продукту требует получения разрешения.

ГЛАВА 1

Начало путешествия

В этой главе мы познакомим вас с устройством книги и правилами работы с ней. В целом основной подход к материалу в книге можно описать следующим образом: *практика и общие принципы выше конкретной технологии*. На сегодняшний день существует множество книг по каким-то конкретным направлениям, и мы не стремимся стать частью этого огромного собрания. Мы не хотим сказать, что узкопрофильные знания по какому-то конкретному языку, среде или библиотеке не будут полезны. Просто их жизненный цикл короче, чем у общих знаний и понятий, которые можно применять к разным языкам программирования и технологиям в течение длительного периода времени. Вот в чем вам должна помочь эта книга.

Темы

Мы использовали в книге проектно-ориентированную структуру, чтобы вам было удобнее учиться. Такой подход позволяет объединить разные темы, заставляет задуматься о том, как они связаны друг с другом и почему мы их выбрали. Ниже перечислены четыре темы, которые переплетаются в последующих главах.

Особенности Java

Структурирование кода при помощи классов и интерфейсов обсуждается в главе 2. Исключения и пакеты рассматриваются в главе 3, из которой вы также узнаете о лямбда-выражениях. Глава 5 познакомит вас с выводом типа переменной и `switch`-выражениями. И, наконец, в главе 7

уже более детально рассматриваются лямбда-выражения и ссылки на методы.

Большое количество программных продуктов написаны на Java, поэтому очень важно знать особенности этого языка и понимать, как он работает. Многие из этих особенностей характерны и для других языков программирования, таких как C#, C++, Ruby или Python. Несмотря на то что перечисленные языки имеют различия, понимание того, как использовать классы и основные принципы ООП, будут полезны при работе с любым из них.

Разработка программного обеспечения и архитектура

По ходу книги вы познакомитесь с рядом шаблонов проектирования, которые помогут вам находить стандартные решения распространенных проблем, встречающихся в разработке. Это важно, поскольку, несмотря на то, что каждый проект является индивидуальным и решает конкретные задачи, на самом деле многие из этих задач уже встречались ранее. Понимание стандартных проблем и существующих методов их решения, придуманных другими программистами, убережет вас от изобретения колеса и позволит существенно сэкономить время на разработке.

В главе 2 представлены концепции связанности и связности. Шаблон уведомления рассматривается в главе 3. Как построить «дружелюбный» Fluent API и шаблон Builder, описано в главе 5. В главе 6 рассматривается концепция «целостной картины» событийно-ориентированной и гексагональной архитектур, а в главе 7 представлен шаблон Repository. Наконец, также в главе 7 вы познакомитесь с функциональным программированием.

SOLID

В разных главах книги мы рассмотрим все принципы SOLID, представляющие собой набор правил и наилучших подходов, разработанных с целью облегчения обслуживания программного обеспечения. Если программа, которую вы написали, стала успешной, она потребует развития и сопровождения. Попытки сделать программное обеспечение максимально простым в обслуживании способствуют его «эволюции», добавлению возможностей и функций на долгосрочную перспективу.