Предисловие

Предисловие к третьему изданию

В 1997 году, когда язык Java был еще новинкой, отец Java, Джеймс Гослинг (James Gosling), описал его как “весьма простой язык для синих воротничков” [14]. Примерно в то же время отец C++, Бьярне Страуструп (Bjarne Stroustrup), описал C++ как “мультипарадигменный язык”, который “сознательно сделан отличным от языков, созданных для поддержки единого способа написания программ” [47]. Касаясь языка программирования Java, Страуструп предупредил:

> Большая часть относительной простоты языка программирования Java — как в большинстве новых языков — отчасти иллюзия, а отчасти результат его незавершенности. Со временем размеры и сложность Java значительно вырастут. Его размер удвоится или утроится, как увеличится и количество зависящих от реализации расширений или библиотек [46].

Сейчас, двадцать лет спустя, можно сказать, что и Гослинг, и Страуструп были правы. Java теперь большой и сложный язык программирования с множеством абстракций для множества вещей — от параллельного выполнения до итераций и представления даты и времени.

Мне все еще нравится Java, хотя мой пыл несколько остыл с его ростом. Учитывая увеличение размера и сложности языка, все более важной является необходимость руководства по этому языку программирования, охватывающего наилучшие современные практики его применения. В третьем издании данной книги я сделал все, чтобы предоставить вам именно такое руководство. Я надеюсь, что это издание, оставаясь верным духу первых двух изданий, удовлетворит всем вашим потребностям.

Сан-Хосе, Калифорния
Ноябрь 2017

P.S. Было бы упущением не упомянуть о передовой практике, которая ныне отнимает изрядное количество моего времени. С момента рождения нашей области в 1950-е годы мы, программисты, свободно обменивались и перепроектировали API друг друга. Эта практика имела и имеет решающее значение

для быстрых успехов компьютерных технологий. Я активно работаю для того, чтобы сохранить эту свободу и далее [9], и призываю вас присоединиться ко мне. Если мы сохраним право повторной реализации API друг друга, это будет иметь решающее значение для дальнейшего здоровья всей нашей отрасли.

Предисловие ко второму изданию

С тех пор как я написал первое издание этой книги в 2001 году, в платформе Java произошло много изменений, и я решил, что настала пора написать второе издание. Наиболее значимыми изменениями стали добавление обобщенных типов, типов перечислений, комментариев, автоматической инкапсуляции, циклов по диапазону в Java 5. Новшеством стало добавление новой библиотеки `java.util.concurrent`, также появившейся в Java 5. Мне повезло, что вместе с Гиладом Брачей я смог возглавить команду, разрабатывавшую новые возможности языка. Мне также повезло работать в команде, разрабатывавшей библиотеку параллельных вычислений и возглавляемой Дугом Ли (Doug Lea).

Другим значительным изменением платформы стало ее оснащение современными интегрированными средами разработки, такими как Eclipse, lntelliJ IDEA и NetBeans, а также инструментами статического анализа, как, например, FindBugs. Хотя я и не принимал в этом участия, но смог извлечь для себя огромную выгоду и узнал, как все эти инструменты влияют на опыт разработки на языке программирования Java.

В 2004 году я перешел из Sun в Google, однако продолжал принимать участие в разработке платформы Java в течение последних четырех лет, работая в области API для параллельных вычислений и коллекций, используя офисы Google и Java Community Process. Я также имел удовольствие использовать платформу Java для разработки библиотек для внутреннего использования в Google. Теперь я знаю, что чувствуют пользователи.

Когда в 2001 году я писал первое издание книги, моей основной целью было поделиться моим опытом с читателями, чтобы они могли повторить мои успехи и избежать моих неудач. Новый материал продолжает традицию использования реальных примеров из библиотек платформы Java.

Успех первой редакции превзошел все мои ожидания, и, освещая новый материал, я сделал все возможное, чтобы сохранить дух предыдущего издания. Избежать увеличения книги было невозможно — и теперь в ней вместо 57 уже 78 разделов. И я не просто добавил 23 новых раздела, а тщательно переработал исходное издание, в том числе удалив некоторые ставшие неактуальными разделы. В приложении вы можете увидеть, как соотносятся материалы данного издания с материалом первого издания.

В предисловии к первому изданию я писал, что язык программирования Java и его библиотеки очень способствуют качеству и производительности труда и что работать с ними — одно удовольствие. Изменения в версиях 5 и 6 сделали их еще лучше. Сейчас платформа Java гораздо больше и сложнее, чем в 2001 году, но, познакомившись с ее проектными шаблонами и идиомами, использующими новые возможности, вы улучшите свои программы и упростите свою жизнь. Надеюсь, что это издание передаст вам мой энтузиазм и поможет более эффективно и с б*о*льшим удовольствием использовать платформу и ее новые возможности.

Сан-Хосе, Калифорния
Апрель 2008

Предисловие к первому изданию

В 1996 году я направился на запад работать в компании JavaSoft (как она тогда называлась), поскольку для меня было очевидно, что именно там происходят главные события. На протяжении пяти лет я работал архитектором библиотек для платформы Java. Я проектировал, реализовывал и сопровождал множество библиотек, а также давал консультации по многим другим библиотекам. Руководить этими библиотеками в ходе становления платформы языка Java — такая возможность выпадает только раз в жизни. Не будет преувеличением сказать, что я имел честь работать с некоторыми великими программистами нашего поколения. В процессе работы я узнал о языке программирования Java очень многое: что хорошо работает, а что — нет и как пользоваться языком и его библиотеками для получения наиболее эффективного результата.

Эта книга является попыткой поделиться с вами моим опытом, чтобы вы могли повторить мои успехи и избежать моих ошибок. Формат книги я позаимствовал из книги Скотта Мейерса (Scott Meyers) *Effective C++* [33], состоящей из разделов, каждый из которых посвящен одному конкретному правилу, позволяющему улучшить программы и проекты. Такой формат кажется мне необычайно эффективным, и, надеюсь, вам он также понравится.

Во многих случаях я иллюстрирую разделы реальными примерами из библиотек платформы Java. Говоря, что нечто можно сделать лучше, я стараюсь привести исходный текст, который я писал сам, однако иногда я брал разработки моих коллег. Приношу мои искренние извинения, если, несмотря на все старания, я кого-либо при этом обидел. Отрицательные примеры приведены не для того, чтобы кого-то опорочить, а в соответствии с духом сотрудничества, чтобы все мы могли извлечь пользу из опыта тех, кто прошел этот путь ранее.

Хотя эта книга предназначена не только для людей, занимающихся разработкой повторно используемых компонентов, она неизбежно отражает мой опыт в написании таковых, накопленный за последние два десятилетия. Я привык думать в терминах интерфейсов прикладного программирования (API) и предлагаю вам поступать так же. Даже если вы не занимаетесь разработкой повторно используемых компонентов, мышление в соответствующих терминах может повысить качество разрабатываемых вами программ. Более того, нередко случается писать многократно используемые компоненты, даже не подозревая об этом: вы написали что-то полезное, поделились своим результатом с приятелем, и вскоре оказывается, что у вашего кода уже с полдюжины пользователей. С этого момента вы лишаетесь возможности свободно менять этот API и благодарите сами себя за усилия, которые потратили на его первоначальную разработку и которые позволили получить качественный продукт.

Мое особое внимание к разработке API может показаться несколько противоестественным для ярых приверженцев новых облегченных методик разработки программного обеспечения, таких как "*Экстремальное программирование*" [2]. В этих методиках особое значение придается написанию самой простой программы, какая только сможет работать. Если вы пользуетесь одной из таких методик, то вскоре обнаружите, что особое внимание к разработке API вернется сторицей при последующем *рефакторинге* программы. Основной задачей рефакторинга является усовершенствование системной структуры, а также исключение дублирования кода. Этой цели невозможно достичь, если у системных компонентов нет хорошо продуманного и спроектированного API.

Ни один язык не идеален, но некоторые из них великолепны. Я обнаружил, что язык программирования Java и его библиотеки в огромной степени способствуют повышению качества и производительности труда и что работать с ними — одно удовольствие. Надеюсь, эта книга сможет передать вам мой энтузиазм и сделает вашу работу с языком Java более эффективной и приятной.

Купертино, Калифорния
Апрель 2001