Содержание

Глава 1

НАЧАЛО

1.1. Введение

1.1.1. Почему именно эта книга?

Эта книга предназначена тем, кто интересуется разработкой программного обеспечения, но имеет мало практического опыта или вовсе не имеет такового.

Книга не содержит никакой новой информации по сравнению с той, которую можно найти в Интернете или в других книгах. Она преследует две цели:

- объединить под одной обложкой то, что я считаю важным для разработки программного обеспечения, из большого числа тем, избавляя читателя от дилеммы “что мне изучать”;
- собрать воедино актуальную информацию из множества источников, что избавит читателя от необходимости просматривать несколько отдельных сайтов и книг.

В книге содержится обзор практически всего, что, по моему мнению, важно для разработчиков программного обеспечения. Это далеко не *всё*, но этого должно быть достаточно для того, чтобы читатель понимал, что такое разработка программного обеспечения, и мог при необходимости ознакомиться с любой темой более подробно.

Основное внимание в книге уделяется обучению основным принципам разработки программного обеспечения. Для этого используются несколько технологий (например, C, Python, JavaScript, HTML и т. д.), но эта книга не о самих технологиях. По ходу изложения материала читатель узнает основы (а в некоторых случаях и больше) различных технологий, но главное внимание уделяется “созданию фундамента”, получению базовых знаний для разработки программного обеспечения.

Цель этой книги — поддержать заинтересованных людей в получении работы в области разработки программного обеспечения,

хотя я, конечно, не могу гарантировать получение работы. Подробнее об этом я расскажу ниже.

Почему эта книга написана?

Моя жена спрашивала, может ли она стать инженером-программистом, как я.

У меня несколько лет опыта работы в области разработки программного обеспечения, и сейчас я нанимаю инженеров-программистов и, соответственно, провожу собеседования с ними. Я проверил программы обучения в сфере информатики (компьютерных наук, Computer Science) в местном университете, но мне кажется, что существует некоторый разрыв между тем, что реально нужно промышленности, и тем, что преподают в учебных заведениях — по крайней мере, в моем регионе. В какой-то степени это ожидаемо: университеты живут в академическом мире, часто не ставя перед собой цели обеспечить выпускников работой в отрасли. Чтобы смягчить ситуацию, университеты стараются предлагать специальности, более ориентированные на реальный мир, но и они могут оказаться недостаточно актуальными и подходящими для работодателей.

В итоге я стал учить свою жену тому, что считаю нужным. Поскольку у нас есть маленькая дочь, найти время для занятий иногда было непросто, поэтому я записывал для нее некоторые заметки или упражнения, которые она могла выполнять, когда у супруги появлялось свободное время. В итоге я писал все больше и больше, структурировал все лучше и лучше, и все это вылилось в данную книгу.

Таким образом, я написал эту книгу с целью научить свою жену необходимым основам разработки программного обеспечения, чтобы она смогла найти работу. Однако книга может быть полезна и другим людям.

Нет, правда, зачем вы написали эту книгу?

Если уж говорить совсем откровенно, отчасти и в личных интересах, потому, что я нанимаю инженеров-программистов, и мне трудно найти хороших специалистов. Не вдаваясь в конкретные требования, я ставил целью включить в книгу достаточно информации, чтобы читатель мог выполнить все нижеперечисленные действия:

- прочитать и понять все, что написано в книге;
- выполнить все упражнения;

- написать какой-нибудь персональный мини-проект на каком-нибудь языке (после выполнения двух вышеперечисленных пунктов это не должно быть проблемой);
- взаимодействовать с остальными членами сообщества разработчиков программного обеспечения через Reddit или другие социальные сети, чаты, Stack Overflow или GitHub.

После этого я буду считать, что данный человек обладает *техническими* навыками, чтобы по крайней мере серьезно рассматриваться в качестве претендента на должность инженера-программиста.

Помимо технических навыков вам понадобятся навыки межличностного общения, здравый смысл, аналитическое и критическое мышление и т. д., прежде чем я рассмотрю вашу кандидатуру на должность инженера-программиста. Я рассчитываю, что читатель получит их иным способом, нежели прочитав эту книгу.

То есть если я выполню все вышеперечисленные пункты, я тоже смогу стать профессиональным инженером-программистом?

Я, конечно, не могу ничего гарантировать. На самом деле получение работы зависит от состояния экономики и рынка труда, от места проживания, от того, что вы готовы делать и за что работать и т. д. Но я думаю, что после выполнения всех вышеперечисленных пунктов читатель будет обладать достаточными или почти достаточными *техническими* навыками для многих профессий, связанных с разработкой программного обеспечения.

Помимо пробуждения надежд на получение работы в качестве инженера-программиста еще одна причина написания этой книги заключается в том, что мне не нравится, как растет разрыв между “технарями” и “нетехнарями”. Я считаю, что больше людей должны быть знакомы с программным обеспечением. Я ставил своей задачей написать книгу, которая не требует особых предварительных условий для прочтения и понимания и рассказывает о программном обеспечении и технологиях достаточно, чтобы “нетехнарь” смог лучше понять, как программное обеспечение влияет на мир, в котором мы живем. Так что даже если вы не найдете ра-

боту, то вы, по крайней мере, сможете гордиться тем, что лучше понимаете программное обеспечение и его роль современном в мире.

Что входит в эту книгу?

Книга охватывает много тем, но довольно поверхностно.

Она не содержит *всего,* что я знаю, но ее достаточно для того, чтобы читатель мог следить за ходом обсуждения вопросов разработки программного обеспечения и, что еще важнее, узнать больше о конкретном объекте разработки программного обеспечения, не прибегая к интенсивному специализированному обучению.

Рассматриваются различные языки программирования, такие как C, Python и JavaScript. Затрагиваются некоторые традиционные темы информатики или компьютерных наук (Computer Science), такие как алгоритмы и структуры данных. Затронуты некоторые темы, характерные для современных сред разработки программного обеспечения, такие как базы данных, контроль версий и веб-разработка. Практически ни одна из тем не рассматривается подробно.

Хотя многие темы излагаются почти в “комически легкой форме”, в целом представленного материала должно быть достаточно для того, чтобы читатель получил общее представление о разработке программного обеспечения.

Эта книга не является заменой образования в области компьютерных наук, хотя и включает (неглубоко) некоторые его разделы, а в иных случаях дополняет его.

Вы говорите “инженерия”, “программирование”, “разработка” и “компьютерные науки” (Computer Science) — чем они отличаются?

Эта книга учит разработке программного обеспечения. В ней не преподается инженерное дело или проектирование (Engineering), но термин “инженер-программист” (software engineer) используется для определения человека, разрабатывающего программное обеспечение, поскольку именно к этому мы, похоже, пришли как отрасль.

Более конкретно я бы определил терминологию следующим образом.

- Инженерное дело, проектирование (Engineering) — это нечто весьма структурированное, и изучать его нужно также

структурированно. Другими словами, эта книга точно ему не учит.

- Разработка программного обеспечения (Software development) — это решение проблемы с помощью программного обеспечения: это не идеальное определение, решения, как правило, не совсем “инженерны” и не очень научны, но это именно то, что требуется на практике и на что есть спрос.

Программирование (Programming, Coding) является одной из составляющих разработки программного обеспечения, наряду с проектированием, тестированием, написанием спецификаций и т. д. Эта книга посвящена разработке программного обеспечения, но именно программирование играет в ней основную роль.

Есть еще “информатика”, или “компьютерные науки” (Computer Science), которая, пожалуй, отделена и от разработки программного обеспечения, и от инженерии/проектирования. (Как говорится, “компьютерные науки относятся к компьютерам не больше, чем астрономия к телескопам”).

В одной книге, наверное, невозможно собрать весь необходимый материал!

Нет, но можно попробовать.

Какой бы код вы ни стали писать после прочтения этой книги, вам придется более подробно изучить технологию по каждой конкретной теме, которая потребуется. Цель этой книги не в том, чтобы рассказать обо всем; цель — создать основу, чтобы читатель мог научиться писать практически любые программы без существенных препятствий, если это необходимо.

Каковы предварительные требования к читателям этой книги?

Книга рассчитана на людей, не имеющих опыта и навыков разработки программного обеспечения.

В некоторых частях книги используется математика на уровне средней школы, поэтому вы должны быть в какой-то степени знакомы с ней. Необходимо иметь компьютер и уметь им пользоваться (устанавливать программы и т. д.).

Для работы с книгой требуется, чтобы на вашем компьютере были установлены оболочка Unix и стандартные инструменты Unix. Для этого подойдут Mac, Linux или Windows 10. Более старые версии Windows также могут хорошо подойти для целей данной книги, но установка необходимого программного обеспечения может оказаться нетривиальной.

Вы должны обладать некоторыми навыками аналитического мышления и решения проблем. Я не знаю, как этому научить. Полагаю, что эта книга в какой-то степени учит им, хотя бы в качестве побочного эффекта.

Я верю, что практически любой человек, способный без особых проблем закончить среднюю школу, может научиться разрабатывать программное обеспечение. Это не волшебство. Однако это требует упорства; вы должны быть в состоянии приложить усилия при работе с книгой, и это займет время. Я считаю, что человек, способный работать над книгой полный рабочий день, достаточно настойчивый и получающий помощь там, где это необходимо, сможет закончить книгу, включая все упражнения, в течение нескольких месяцев.

Возможно, будет очень полезно иметь наставника или человека, знакомого с разработкой программного обеспечения, который сможет ответить на ваши вопросы. Используйте Интернет в своих интересах: в сети множество технарей, готовых поделиться своими знаниями и ответить на вопросы новичков. Попробуйте обратиться к Stack Overflow, подразделу learnprogramming, GitHub или различным IRC-каналам, например, посвященным конкретным языкам программирования.

Вы должны уметь находить информацию в Интернете. Например, обратите внимание, как я использовал термин “IRC-каналы”. Если у вас когда-нибудь возникнет необходимость задать вопрос и вы решите исследовать IRC дальше, вам придется воспользоваться Интернетом, чтобы а) узнать, что такое IRC, б) выяснить, какие существуют IRC-каналы, например, для языка программирования, с которым у вас возникли проблемы, в) как подключиться к такому каналу и задать свой вопрос. В этой книге такой информации не будет.

бом носителе и изменять ее для любых целей, даже коммерческих, при условии, что вы укажете соответствующую ссылку и ваше распространение будет осуществляться по той же лицензии. Более подробную информацию можно найти на сайте http://creativecommons.org/licenses/ by-sa/4.0/. Код, содержащийся в этой книге, распространяется под лицензией MIT.

Как работать с этой книгой?

Информация в книге изложена очень насыщенно. Большинство предложений очень важны для рассматриваемой темы, информация редко дублируется. Я ожидаю, что читатель прочтет книгу несколько раз. Возможно, в первый раз или во второй вы пропустите много деталей, но в конечном итоге вы сможете понять всё.

В книге используется педагогический подход ассимиляции, или конструктивизма: информация предоставляется читателю в большом объеме, а усвоение материала облегчается упражнениями, в которых “студенту” предлагается самостоятельно подумать, пытаясь усвоить материал. Если вы чего-то не понимаете, это не страшно. Оставьте это и вернитесь к сложному для понимания материалу позже. В конце книги приведена диаграмма зависимости глав, которая позволяет понять, какие главы необходимо изучить, прежде чем продолжать обучение. Различные темы в той или иной степени чередуются, что позволяет читателю усваивать одни области, работая над другими, и лучше понимать взаимосвязи и взаимоотношения между темами.

Здесь следует отметить, если это еще не ясно, что автор не имеет реального педагогического опыта.

В целом книга состоит из четырех этапов. Вот основные темы различных этапов:

Этапы	Основные темы
Начало	• Основы компьютерной грамотности и программирования на языках C и Python • Основы Unix
Этап 1	• Введение в алгоритмы и язык JavaScript • Некоторые более глубокие концепции языков C и Python

Этапы	Основные темы
Этап 1.5	• Веб-разработка с использованием JavaScript и Python • Статически типизированные языки, особенно C++
Этап 2	• Крупные программы на Python и C++ (или, как вариант, на Java) • SQL • Различные промежуточные темы (парсинг, потоки и т. д.)

Если вы считаете, что в каком-то месте книги что-то неясно изложено либо у вас есть вопросы или комментарии, дайте мне знать. Мой адрес электронной почты — ajsalonen@gmail.com. Вы также можете оставить вопрос или запрос в GitHub. Исходный код книги доступен по адресу https://github.com/anttisalonen/progbook. Домашняя страница книги — https://progbook.org.

Об авторе

Меня зовут Антти Салонен. Я несколько лет проработал инженером-программистом и в настоящее время являюсь руководителем инженерного отдела в технологической компании.

Я начал писать код, когда мне было около шести лет, или 28 лет назад, и создавал нетривиальные коды примерно на 14 различных языках программирования. Я профессионально писал программное обеспечение для медицинских приборов, внутренних инструментов компании, программное обеспечение для управления телескопом и другое. Я руководил разработкой веб-приложения, был сетевым администратором, руководителем инженерной группы и архитектором ПО. В качестве хобби я написал несколько скромных незатейливых игр с открытым исходным кодом и внес небольшой вклад в некоторые проекты с открытым исходным кодом.

Отказ от ответственности: Эта книга была написана от моего личного имени. Все взгляды и мнения, высказанные в этой книге, являются моими собственными и могут не совпадать с мнением моего прошлого или нынешнего работодателя.

Хотя я начал программировать в юном возрасте, это не означает, что ваш случай безнадежен, если вы начинаете только сейчас. Более того, я знаю нескольких великих разработчиков программного обеспечения, которые начали программировать, будучи уже зрелыми людьми.

1.1.2. Что такое программное обеспечение (Software)?

Разговоры — это дешево. Покажите мне код.
Линус Торвальдс

Программное обеспечение (Software) — это код, который указывает компьютерному оборудованию (Hardware), что делать.

Для начала давайте напишем очень простую программу.

Для этого прежде всего необходимо запустить терминал с оболочкой Unix.

Я не могу сказать, как именно это сделать, поскольку это зависит от операционной системы (ОС). По крайней мере, в некоторых версиях Mac OS терминал можно запустить с помощью поиска “terminal” в Spotlight. В Windows 10 можно установить “Подсистему Windows для Linux” (WSL). В старых версиях Windows можно получить Unix-оболочку, например Cygwin, но это более сложная задача. Поиск в Интернете вам поможет.

В итоге это может выглядеть примерно так:

В частности, cmd.exe или PowerShell в Windows *не* являются оболочками Unix. Типичными характеристиками оболочек Unix являются знак “at” (“@”), обозначающий имя пользователя и имя хоста компьютера, на который вы вошли, тильда (“~”), обозначающая домашний каталог, и знак доллара (“$”), обозначающий приглашение. С их помощью можно определить, запущена ли оболочка Unix.