

Предисловие

“К черту мины!
Полный вперед”!

Адмирал Фаррагут¹

Программирование — это искусство выражать решения задач так, чтобы компьютер мог их осуществить. Основные усилия программиста направлены на то, чтобы найти и уточнить решение, причем довольно часто полное понимание задачи приходит лишь в ходе программирования ее решения.

Эта книга предназначена для тех, кто еще никогда не программировал, но готов тяжело работать, чтобы научиться этому. Она поможет овладеть главными принципами и приобрести практический опыт программирования на языке C++. Моя цель заключается в том, чтобы изложить достаточный объем сведений и научить вас решать простые и полезные задачи по программированию с помощью самых лучших и современных методов. Если вы учитесь на первом курсе университета, то можете использовать эту книгу на протяжении семестра. Если самостоятельно изучаете программирование, то сможете освоить этот курс не менее чем за 14 недель при условии, что будете работать по 15 часов в неделю. Три месяца могут показаться долгими, но объем курса довольно велик, и первые простые программы вы сможете написать, проработав над книгой не менее часа. Кроме того, сложность материала постепенно возрастает: в каждой главе вводятся новые полезные понятия, которые иллюстрируются реальными примерами. Способность выражать свои идеи на языке программирования, — т.е. умение объяснять компьютеру, что от него требуется, — будет постепенно развиваться у вас по мере изучения. Я никогда не говорю: “Месяц изучайте теорию, а затем проверьте, сможете ли вы ее применить на практике”.

Зачем нужны программы? Современная цивилизация основана на компьютерных программах. Не зная, как работают эти программы, вы будете вынуждены верить в “волшебство”, и многие интересные, выгодные и социально полезные сферы деятельности останутся для вас закрытыми. Когда я говорю о программировании, то думаю о всем спектре компьютерных программ — от программ для персональных компьютеров с графическим пользовательским интерфейсом, программ для

¹ Фаррагут Дэвид Глазго (1801-1870) — первый адмирал США, герой Гражданской войны, воевал за северян. В ходе сражения за порт Мобил провел свой корабль через заминированный проход. — *Примеч.ред.*

инженерных вычислений и встроенных систем управления (например, в цифровых видеокамерах, автомобилях и мобильных телефонах) до приложений, предназначенных для манипулирования текстами. Как и математика, программирование — на высоком уровне — представляет собой полезное интеллектуальное упражнение, оттачивающее мыслительные способности. Однако благодаря обратной связи с компьютером программирование носит более конкретный характер, чем многие области математики, а значит, доступно более широкому кругу людей. С помощью программирования можно разбогатеть и изменить мир к лучшему. Кроме того, программирование — довольно увлекательное занятие.

Почему C++? Потому что невозможно научиться программировать, не зная ни одного языка программирования, а язык C++ поддерживает основные концепции и методы, используемые в реальных компьютерных программах. Язык C++ является одним из наиболее широко распространенных языков программирования. Он применяется во многих прикладных сферах. Программы, написанные на языке C++, можно встретить всюду: начиная с батискафов на дне океана до космических аппаратов на поверхности Марса. Кроме того, существует точный и полный международный стандарт языка C++, не защищенный патентом. Качественные и/или свободные реализации этого языка доступны для любых компьютеров. Большинство концепций программирования, которые вы изучите с помощью языка C++, можно непосредственно использовать в других языках, таких как C, C#, Fortran и Java. И вообще, я просто люблю этот язык за элегантность и эффективность кода.

Эту книгу нельзя назвать самым простым введением в программирование. Собственно, эту цель я перед собой не ставил. Я просто хотел написать легкую и понятную книгу, с помощью которой можно было бы освоить азы практического программирования. Это довольно амбициозная цель, поскольку современное программное обеспечение в большой степени основывается на методах, изобретенных совсем недавно.

Надеюсь, что вы — люди ответственные и хотите создавать программы, предназначенные для других пользователей, стараясь обеспечить при этом их высокое качество. Иначе говоря, я предполагаю, что вы желаете достичь определенной степени профессионализма. По этой причине книга начинается с действительно нужных вещей, а не просто с самых легких для обучения тем. В ней описаны методы, необходимые для правильного программирования, а также приведены связанные с ними понятия, средства языка и упражнения. Надеюсь, что вы обязательно выполните их. Люди, интересующиеся лишь игрушечными программами, извлекут из книги намного меньше, чем в нее заложено. С другой стороны, я бы не хотел, чтобы вы растрачивали свое время на материал, который редко находит применение на практике. Если в книге изложена какая-то идея, значит, я считаю, что она почти наверняка понадобится в реальных приложениях.

Если хотите использовать результаты работы других людей, не вникая в детали и не желая добавлять к ним свой собственный код, то эта книга не для вас. Если это

так, то подумайте, не следует ли вам выбрать другую книгу и другой язык программирования. Кроме того, задумайтесь над тем, почему вы придерживаетесь такой точки зрения и соответствует ли она вашим потребностям. Люди часто переоценивают сложность программирования, а также его стоимость. Я не хотел бы вызывать у читателей отвращение к программированию из-за несоответствия между их потребностями и содержанием книги. Существует множество областей мира “информационных технологий”, в которых программировать совершенно не требуется. Напоминаю, что эта книга предназначена для тех, кто хочет писать или понимать нетривиальные программы.

Благодаря структуре и предназначению книги ее могут также использовать люди, уже знакомые с основами языка C++ или владеющие другим языком программирования и желающие изучить C++. Если вы попадаете в одну из этих категорий, то мне сложно предположить, сколько времени у вас займет чтение этой книги, но я настоятельно рекомендую обязательно выполнить упражнения. Это поможет решить широко распространенную задачу: адаптировать программы, написанные в старом стиле, с учетом более современных технологий. Если вы овладели языком C++, используя традиционные способы обучения, то, возможно, найдете нечто удивительное и полезное в первых шести главах. Здесь рассматриваются темы, которые никак нельзя назвать “C++ времен вашего отца” (если только ваша фамилия не Страуструп).

Изучение программирования сводится к разработке программ. Этим программирование похоже на другие виды деятельности, требующие практических занятий. Невозможно научиться плавать, играть на музыкальном инструменте или водить автомобиль, просто прочитав учебник, — необходима практика. Точно так же невозможно научиться программировать, не прочитав и не написав большое количество программ. Основное внимание в книге сосредоточено на программах, которые сопровождаются пояснениями и диаграммами. Вы должны понять идеи, концепции и принципы программирования, а также овладеть языковыми конструкциями, необходимыми для их выражения. Это очень важно, но само по себе не может дать практического опыта программирования. Для того чтобы приобрести такой опыт, следует выполнить упражнения, используя средства редактирования, компиляции и выполнения программ. Вы должны делать свои собственные ошибки и учиться их исправлять. Заменить разработку собственных программ нельзя ничем. Кроме того, это так увлекательно!

С другой стороны, программирование нельзя сводить к изучению нескольких правил и чтению справочника. В этой книге специально не акцентируется синтаксис языка C++. Для того чтобы стать хорошим программистом, необходимо понимать основные идеи, принципы и методы. Только хорошо разработанный код имеет шанс стать частью правильной, надежной и легкой в эксплуатации системы. Помимо прочего, основы — это то, что останется даже после того, как современные языки и средства программирования будут усовершенствованы или сойдут с арены.

Что можно сказать о компьютерных науках, разработке программного обеспечения, информационных технологиях и т.д.? Сводятся ли эти отрасли знаний к программированию? Разумеется, нет! Программирование — это один из фундаментальных предметов, лежащих в основе всех областей, связанных с использованием компьютеров. Оно занимает свое законное место в курсе компьютерных наук. Я привожу в книге краткий обзор основных понятий и методов, связанных с алгоритмами, структурами данных, пользовательским интерфейсом и программным обеспечением. Тем не менее эта книга не может заменить подробного и сбалансированного учебника по этим темам.

Программа может быть как прекрасной, так и полезной. Надеюсь, эта книга поможет вам понять эту истину. Я старался объяснить, какие программы можно назвать прекрасными, изложить основные принципы их разработки и помочь овладеть практическими навыками по разработке таких программ. Удачи!

Обращение к студентам

Обучение по этой книге уже прошли более тысячи студентов Техасского университета агрокультуры и машиностроения (Texas A&M University). Из них около 60% уже имели опыт программирования, а остальные 40% до обучения не написали ни одной строчки программы в своей жизни. Большинство из них вполне успешно справились с заданиями, значит, справитесь и вы.

Вы не обязаны читать эту книгу как часть учебного курса. Я предполагаю, что эта книга будет широко использоваться для самообучения. Однако, независимо от того, учитесь ли вы в университете или овладеваете программированием самостоятельно, постарайтесь работать в группе. Программирование часто совершенно неправильно считают занятием одиночек. Большинство людей лучше работают и быстрее обучаются в коллективе, имеющем общую цель. Совместное обучение и обсуждение задач с друзьями нельзя сводить к обмену программами для обмана преподавателей! Это один из наиболее эффективных, а также наиболее приятных способов совершенствования своего профессионального мастерства. Кроме того, коллективная работа приучает студентов правильно выражать свои идеи, что является одним из наиболее эффективных способов самопроверки и запоминания. Не обязательно самостоятельно искать решения давно известных задач, связанных с языком программирования или особенностями сред для разработки программ. Однако не следует также обманывать себя, не выполняя упражнения (даже если преподаватель их не проверяет). Помните: программирование (помимо всего прочего) — это практический навык, которым следует овладеть. Если вы не пишете программ (т.е. не выполняете упражнения, приведенные в конце каждой главы), то чтение книги сведется к бессмысленному теоретическому занятию.

У большинства студентов — особенно хороших студентов — иногда возникает вопрос, стоит ли так тяжело работать. Если у вас возникнет такой вопрос, сделайте перерыв, перечитайте предисловие и просмотрите главу 1 “Компьютеры, люди и про-

граммирование”, а также главу 22 “Идеалы и история”. В этих главах я попробовал объяснить, чем меня восхищает программирование и почему я считаю, что именно программирование играет ключевую роль в улучшении мира. Если вас интересуют мои педагогические принципы, прочитайте главу 0 “Обращение к читателям”.

Возможно, вы сочтете книгу слишком большой. Частично это объясняется тем, что я либо многократно повторяю объяснения, либо иллюстрирую их дополнительными примерами, вместо того чтобы заставлять читателей удовлетвориться единственным толкованием. Кроме того, часть II представляет собой справочник и содержит дополнительный материал, позволяющий читателям углубить свои знания в конкретных областях программирования, например в области встроенных систем программирования, анализа текстов или математических вычислений.

Пожалуйста, сохраняйте терпение. Изучение новых идей и приобретение важных практических навыков требует времени, но результат стоит затраченных усилий.

Обращение к преподавателям

Нет, это не традиционный курс по компьютерным наукам, принятый в США (Computer Science 101). Эта книга о том, как создать работающее программное обеспечение. Поэтому за ее рамками осталось многое из того, что обычно включается в учебник по компьютерным наукам (сложность алгоритмов по Тьюрингу, конечные автоматы, дискретная математика, грамматики Хомского и т.д.). В книге проигнорирована даже тема, связанная с аппаратным обеспечением, поскольку я полагаю, что студенты с детства умеют работать с компьютерами. В книге даже не упоминается большинство важных тем из области компьютерных наук. Это книга о программировании (а точнее, о том, как разработать программное обеспечение), поэтому в ней нет того, что обычно включается в традиционные учебники. Поскольку компьютерные науки невозможно изложить в рамках одного курса, я не стал распылять внимание. Если вы будете использовать эту книгу как часть курса по компьютерным наукам, вычислительной технике, электротехнике (большинство наших первых студентов специализировались именно по электротехнике), информатике или какой-либо иной научной дисциплине, то предлагаю выделить ее в виде самостоятельного введения.

Пожалуйста, прочитайте главу 0, “Обращение к читателю”, чтобы понять мои педагогические принципы, общий подход и т.д. Я прошу вас передать эти идеи вашим студентам.

Веб-сайт

Книге посвящен отдельный веб-сайт www.stroustrup.com/Programming², содержащий дополнительные материалы для обучения программированию. Со вре-

² Актуальность ссылок и содержания веб-страниц не гарантируется. — *Примеч. ред.*

менем этот материал, по-видимому, будет уточняться, но в данный момент читатели найдут там следующие материалы.

- Слайды лекций по этой книге.
- Справочник преподавателя.
- Заголовочные файлы и реализации библиотек, использованных в книге.
- Тексты программ, используемых в книге в качестве примеров.
- Решения избранных упражнений.
- Потенциально полезные ссылки.
- Список найденных ошибок.

Любые замечания и предложения по улучшению книги будут приняты с благодарностью.

Благодарности

Я особенно благодарен моему коллеге Лоуренсу “Питу” Петерсену (Lawrence “Pete” Petersen) за то, что он вдохновил меня взяться за обучение новичков и полезные практические советы по преподаванию. Без его помощи первый вариант этого курса оказался бы неудачным. Мы работали вместе над первым вариантом курса, которому посвящена эта книга, а затем совершенствовали ее, учитывая полученный опыт. Местоимение “мы”, использованное в книге, означает “Пит и я”.

Выражаю признательность студентам, ассистентам и преподавателям Техасского университета агрокультуры и машиностроения (курс ENGR 112), которые вольно или невольно помогли написать эту книгу, а также Уолтеру Догерити (Walter Daugherty), прослушавшему этот курс. Кроме того, я благодарен Дэмиану Дечеву (Damian Dechev), Трейси Хэммонд (Tracy Hammond), Арне Толstrupу Мэдсену (Arne Tolstrup Madsen), Габриэлю Дос Рейосу (Gabriel Dos Reis), Николасу Страуструпу (Nicholas Stroustrup), Дж. К. ван Винкелю (J. C. van Winkel), Грэгу Версундеру (Greg Versoonder), Ронни Уарду (Ronnie Ward) и Леору Зольману (Leor Zolman) за конструктивные замечания к рукописи книги. Большое спасибо Могенсу Хансену (Mogens Hansen) за объяснение принципов работы программного обеспечения по управлению двигателем, а также Элу Ахо (Al Aho), Стивену Эдвардсу (Stephen Edwards), Брайану Кернигану (Brian Kernighan) и Дэйзи Нгуен (Daisy Nguyen) за то, что помогли мне спрятаться от того, что могло отвлечь от работы на книгой на протяжении летних месяцев.

Благодарю рецензентов, которых издательство Addison-Wesley подыскало для меня: Ричарда Энбоди (Richard Enbody), Дэвида Густафсона (David Gustafson), Рона Мак-Карти (Ron McCarty) и К. Нараянаасвами (K. Narayanaswamy). Их комментарии, основанные в основном на преподавании языка C++ или курса Computer Science 101 на уровне колледжа, были очень полезными. Я признателен также моему редактору Питеру Гордону (Peter Gordon) за многочисленные комментарии

и терпение (не в последнюю очередь). Большое спасибо техническому персоналу издательства Addison–Wesley. Они много сделали для повышения качества книги: корректору Джулии Грейди (Julie Grady), верстальщику Крису Кини (Chris Keane), художнику Робу Мохару (Rob Mauhar), техническому редактору Джулии Нахил (Julie Nahil) и литературному редактору Барбаре Вуд (Barbara Wood).

В дополнение к моим несистематическим попыткам проверить тексты программ Башар Анабтави (Bashar Anabtawi), Юнан Фан (Yinan Fan) и Юрий Солодкий (Yuriy Solodkyu) проверили все фрагменты программ с помощью компиляторов Microsoft C++ 7.1 (2003), 8.0 (2005) и GCC 3.4.4.

Я хотел бы также поблагодарить Брайана Кернигана и Дуга Мак-Элроя (Doug McProy) за очень высокие стандарты качества, установленные ими для программирования, а также Денниса Ритчи (Dennis Ritchie) и Кристена Нийгарда (Kristen Nygaard) за ценные уроки по практической разработке языков программирования.