

Предисловие

Учебные курсы и материал книги

Читая на протяжении девяти лет курс по структурам данных в университете, я заметил, что студенты делятся на две категории. К первой относятся энтузиасты программирования. Студенты, принадлежащие ко второй категории, напротив, программировать не любят и посещают лекции лишь потому, что курс является обязательным. Эти студенты надеются впоследствии найти работу, непосредственно не связанную с написанием программ. Кроме того, есть и такие, которых нельзя отнести ни к первой, ни ко второй категории. Они не прочь написать программу, но предпочитают делать это с помощью специализированных инструментальных средств.

Как это ни странно, но именно тем студентам, для которых программирование составляет смысл их жизни, курс структур данных поначалу кажется трудным. Они уже умеют достаточно быстро писать программы, но когда сталкиваются с проблемами построения законченной системы, у них не хватает терпения довести работу до конца. К счастью, большинство из них вскоре осознают преимущества подхода, согласно которому сложные системы строятся из простых компонентов, для которых созданы спецификации.

С другой стороны, студенты, испытывающие отвращение к самому процессу программирования, понимают, что если они мыслят в терминах абстрактных типов данных, то когда дело доходит до написания программ, благодаря наличию подробных описаний работа существенно упрощается. Конечно, уйти от программирования им не удастся, но вместо одной глобальной проблемы они сталкиваются с набором небольших задач, каждая из которых оказывается вполне разрешимой.

На мой взгляд, предмет структур данных выходит за рамки “традиционного” построения систем на языках программирования общего назначения. Например, когда перед студентами ставится задача проектирования и реализации веб-узлов, то те из них, кто смог овладеть принципами работы со структурами данных, гораздо быстрее понимают возможности нелинейных структур. Их воображение изначально не сковано, и они могут найти выгодные решения. Даже работая над динамическим связыванием страниц средствами Java, они понимают, что задача разрешима, и видят пути к достижению результата.

Заметьте, что эта книга не называется *Структуры данных для чайников* или *Структуры данных — это очень просто!*. Тем, кто относит себя к “чайникам”, придется поискать другие книги. Я старался изложить суть структур данных как можно проще, но, чтобы изучить этот материал, все же надо приложить большие усилия. Если студент обладает способностями к программированию и хочет стать специалистом, то, тщательно изучив данную книгу, он получит базовые знания, которые существенно помогут в работе. В частности, рассматриваемые в книге примеры имеют много общего с реальными задачами.

Представленный здесь материал базируется на курсе “Структуры данных” и некоторых других курсах, которые я читаю в Оксфордском университете Брукса. Кроме того, существенное влияние на содержание книги оказал курс “Программирование и программные языки”, который я читал в течение шести лет в Открытом университете.

На кого рассчитана эта книга

Студенты

В качестве потенциальных читателей этой книги я в основном рассматривал студентов, изучающих курс структур данных, которые хотели бы глубже изучить динамические структуры и использование указателей. В частности, я имел в виду студентов факультетов компьютерных наук и информационных систем, специализирующихся на разработке программ. Эта книга может стать хорошим пособием по курсу структур данных, рассчитанному на один семестр.

Как я уже говорил, студенты различаются по своему отношению к программированию. Как подтвердит любой преподаватель, читающий соответствующие курсы, некоторые из студентов занимаются написанием программ с большой неохотой.

Другая крайность — это фанатики программирования, которые, не проведя необходимой подготовительной работы, сразу же приступают к написанию кода.

Данная книга пригодится студентам обеих категорий, а также тем студентам, которых нельзя причислить ни к одной из них.

Те, кто еще не испытывает удовольствия от написания работающих программ, увидят, как, следуя подходу, описанному в книге, можно спроектировать и реализовать программу, которая выполняет именно то, что от нее требуется. Ведь часто отвлечение к программированию возникает из-за того, что когда-то не удалось справиться с поставленной задачей и понять, почему большая программа не работает так, как это предполагалось.

Любителям как можно скорее приступить к кодированию я также советую внимательно рассмотреть подход, предложенный в данной книге. Возможно, они поймут, что удовольствие просиживать ночи, составляя “заплаты” к коду, решая одни проблемы и создавая при этом новые, относится к разряду иллюзорных. Гораздо приятнее осознавать, что в результате систематической работы над проектом удалось написать надежно работающую программу. Такая программа изящна, как изящен механизм, созданный талантливым инженером.

Наконец, те студенты, которые уже успели понять, что такое структурное программирование и какие выгоды оно сулит, получают ясное и краткое руководство, позволяющее поднять свой профессиональный уровень на новую ступень.

Прочие читатели

Для того чтобы получить пользу от чтения данной книги, не обязательно быть студентом. Для этого достаточно иметь компьютер с установленным компилятором C++ и возможность хотя бы один раз обратиться к World Wide Web, чтобы скопиро-

вать коды, которые иллюстрируют материал, изложенный в книге. Однако помните, что данная книга не является введением в программирование. Это скорее пособие по курсу структурного программирования. Если же вы начинаете изучение программ “с нуля”, обратитесь к пособиям по C++, ориентированным на начинающих.

Для чего написана эта книга

Я часто слышу от студентов, что в настоящее время нет необходимости вникать в тонкости структур данных и использования указателей; для каждой задачи можно подыскать подходящий пакет. Я совсем не против программных пакетов, но необходимо сознавать, что пакеты ориентированы на определенный класс задач. Поэтому для большинства реальных проблем пакеты позволяют найти в лучшем случае приблизительное решение. Программист, знакомый с языком общего назначения, каковым является C++, может решить задачу гораздо лучше (если, конечно, он обладает достаточной для этого квалификацией). Даже тому, кто выбирает себе профессию, непосредственно не связанную с программированием, знание языка общего назначения не будет лишним: ведь он сможет ясно представить себе решение задачи, пусть даже реализовать это решение придется кому-то другому.

Как аргумент против изучения структур данных мне приводят тот факт, что в библиотеках классов для новых языков программирования уже реализованы основные структуры; примером может служить класс `Vector` в языке Java. Что можно на это возразить? Я сам использую класс `Vector`, если пишу программу на Java. Но это не значит, что я не должен знать принципы построения динамических последовательных структур данных. Более того, не исключено, что в очередном проекте мне понадобится построить сложнейшую структуру данных, которая наверняка не предусмотрена ни в одной из библиотек классов и не описана ни в одной из книг. Я знаю, как, используя понятие абстрактного типа данных, строить из простых структур более сложные, и благодаря этому могу решать самые различные задачи.

Иногда мне говорят, что указатели, вроде бы, устарели, ведь в языке Java они не используются. Работу с указателями сравнивают с попытками завести автомобиль вручную. Но динамические структуры, например деревья, приходится создавать и в Java. На мой взгляд, язык Java не так прост, как C++, но я всегда помню разницу между указателем и объектом, на который он указывает. Проблема возникает тогда, когда мне нужно изменить значение указателя, а не значение объекта, передаваемое в качестве параметра. Причиной тому — мой предыдущий опыт работы с указателями. Несмотря на то что ссылки Java — это всего лишь указатели, на их использование налагаются строгие ограничения; программист не может обращаться с помощью указателя к любой переменной, ссылке или примитиву. Я не говорю, что такой подход плох, но понимать, что такое ссылки, адреса и указатели, не менее важно, чем знать правила работы с ними в языке Java.

Я надеюсь, что вы уже поняли, что, по моему мнению, вы должны извлечь из этой книги. Вы должны понять, что из ограниченного набора “строительных блоков”, в который входят записи, массивы и указатели, вы можете построить сколь угодно сложную структуру. Приступая к работе, вы должны представлять себе, насколько упрощается разработка программы при использовании абстрактных типов данных. Вы должны уметь применять структуры, разработанные вами в предыдущих проектах,

предоставленные вашими коллегами и предлагаемые в составе библиотек. Даже если вы надеетесь быстро подняться до уровня руководителя проекта и выше, помните, что вам все равно придется часто обсуждать с рядовыми программистами возможности программ и детали их реализации.

Выбор языка

Выбор языка — вопрос спорный. Среди языков программирования C++ пользуется большой популярностью, но это не значит, что он лучше всех других. На мой взгляд, поддержка объектно-ориентированного подхода гораздо лучше реализована в Eiffel, но, тем не менее, все больше программистов ориентируются на C++. Я старался добиться того, чтобы инструменты и подходы, изучаемые в нем, были наилучшим образом усвоены студентами и, насколько это возможно, помогли им преодолеть недостатки, присущие тому или иному языку программирования. Учитывая, что большинство читателей уже знакомы с C++ и будут использовать его в своей работе, было принято решение иллюстрировать материал книги примерами на этом языке.

Не секрет, что многие программисты, использующие C++, на самом деле пишут программы в C в среде C++. Это неоднократно наблюдал я, и это подтверждают мои коллеги. В особенности сказанное относится к начинающим программистам. В этом нет ничего плохого, но мне пришлось учитывать данный факт, принимая решение о том, с чего надо начинать изложение материала и следует ли объяснять языковые конструкции. Я решил начать с того, чем заканчивается большинство книг, представляющих собой введение в C++. Студенты, имеющие опыт программирования на таких языках, как Pascal и Modula-2, могут быстро освоить C++ на необходимом уровне. Они могут изучить C++ с самого начала или перейти к нему, рассматривая различия в синтаксисе. Я решил начать изложение материала непосредственно со структур данных, потому что эти вопросы гораздо более важные, чем рассмотрение особенностей того или иного языка. Я также решил не использовать в книге классы. Классы — важное средство, но для понимания абстрактных типов данных достаточно механизма структур. Я еще раз подчеркиваю, что при написании данной книги я не ставил перед собой цель обучить читателя языку программирования. Я лишь использовал языковые средства, имеющие отношение к структурам данных.

И, наконец...

Я буду благодарен вам за все замечания, касающиеся данной книги. Информировать меня, пожалуйста, о всех замеченных вами ошибках. Я очень надеюсь, что ошибок в книге не будет, но если даже корректно составленный фрагмент текста или программы дает читателю повод предположить, что в нем есть ошибка, то этот факт требует рассмотрения.

Кен Браунси

Оксфордский университет Брукса

kwb@brookes.ac.uk

Благодарности и посвящение

Знания о структурах данных, которые мне удалось накопить за 13 лет, я смог получить благодаря разным людям. Во-первых, это преподаватели, которые учили меня еще тогда, когда я был студентом. Во-вторых, это мои коллеги по работе, усилиями которых мое обучение продолжилось. И, наконец, это мои студенты, чьи любознательность и усердие вдохновляли меня на поиски новых путей решения проблем. Спасибо им всем.

Я также благодарен редакторам Джеки Харбору (Jackie Harbor) и Кейт Бревин (Kate Brewin) за помощь и большую работу, проделанную ими при подготовке данной книги.

Эта книга посвящается Джейн, Люси и Вильяму. Спасибо вам!

От издательства

Вы, читатель этой книги, и есть главный ее критик и комментатор. Мы ценим ваше мнение и хотим знать, что было сделано нами правильно, что можно было сделать лучше и что еще вы хотели бы увидеть изданным нами. Нам интересно услышать и любые другие замечания, которые вам хотелось бы высказать в наш адрес.

Мы ждем ваших комментариев и надеемся на них. Вы можете прислать электронное письмо или просто посетить наш веб-сайт, оставив свои замечания, — одним словом, любым удобным для вас способом дайте нам знать, нравится или нет вам эта книга, а также выскажите свое мнение о том, как сделать наши книги более подходящими для вас.

Посылая письмо или сообщение, не забудьте указать название книги и ее авторов, а также ваш e-mail. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию последующих книг. Наши координаты:

E-mail: info@dialektika.com

WWW: <http://www.dialektika.com>