

Предисловие

Когда меня спрашивают о моей работе, разговор быстро переходит к вопросу о том, что такое *информатика* или *компьютерные науки*. Назвать информатику наукой о вычислительных машинах означало бы ввести собеседника в заблуждение, хотя, строго говоря, такое определение нельзя назвать неверным. Но ведь большинство людей понимают под *вычислительными машинами* или *компьютерами* персональные или переносные компьютеры, из чего делают вывод, что специалисты по информатике занимаются конструированием вычислительной аппаратуры. А с другой стороны, назвать компьютерные науки изучающими вычисления означало бы дать им поверхностное определение, поскольку тут же возник бы вопрос о том, что же такое вычисление.

С годами я пришел к выводу, что обучать, вводя одно понятие за другим, — не лучший способ, поскольку это слишком абстрактно. Ныне я, как правило, начинаю описание информатики как науки, изучающей систематическое решение задач. Всем известно, что такое задача, и все понимают, что такое решение. Пояснив представление об информатике на конкретном примере, я нередко пользуюсь удобным случаем ввести понятие алгоритма, что дает мне возможность указать на важные отличия информатики от математики. И зачастую мне не нужно даже вести речь о языках программирования, компьютерах и связанных с ними технических вопросах. Но если дело доходит до этого, то пояснить все подобные понятия совсем нетрудно на примере конкретной задачи. Именно такой подход и развит в этой книге.

Информатика является относительно новым членом клуба наук, и иногда даже кажется, что она все еще не заслужила уважительного отношения со стороны таких серьезных научных дисциплин, как физика, химия или биология. Если взять, к примеру, сцену из фильма о физиках, то можно представить себе ученого-теоретика, обсуждающего сложные формулы, написанные мелом на доске, или практика в лабораторном халате, наблюдающего за ходом эксперимента. Показанный в фильме физик — уважаемый ученый, обладающий драгоценными знаниями. А теперь представьте себе сцену со специалистом по информатике или программированию. В ней вы, вероятнее всего, обнаружите какого-нибудь занудного типа, сидящего в темной неубранной комнате и уставившегося в экран

компьютерного монитора. Он лихорадочно набирает что-то на клавиатуре, вероятно, пытаясь взломать какой-нибудь код или пароль. В обеих сценах решается важная задача, но если физик может дать сколько-нибудь понятное объяснение, что это за задача, зачем она нужна и как ее можно решить, то решение вычислительной задачи зачастую остается тайным, загадочным и обычно слишком сложным, чтобы объяснить его неспециалисту. Но если информатика необъяснима простым смертным, то зачем вообще пытаться узнать о ней больше или понять ее?

Предметом информатики является *вычисление*¹ — явление, оказывающее влияние на каждого. Речь не идет только о мобильных телефонах, переносных компьютерах или Интернете. Складывание бумажного самолетика, поездка на машине на работу, приготовление пищи и даже транскрипция ДНК (процесс, происходящий в клетках вашего организма миллионы раз за то время, пока вы изучаете данную научную дисциплину) — все это примеры вычислений в широком смысле слова, т.е. систематического решения конкретных задач, даже если бы большинство людей не восприняли их именно так.

Наука дает общее представление о мире природы и предоставляет научный метод для надежного упрочения этого представления в виде знания. И то, что относится ко всей науке в целом, касается и информатики в частности, особенно если учесть, что вычисления встречаются в самых разных формах и ситуациях. Следовательно, общее представление о вычислениях дает такие же преимущества для общего представления об окружающем мире и более эффективного решения многих практических задач, как и знание основ физики, химии и биологии. Эта сторона вычислений обычно называется *вычислительным мышлением*.

Главная цель этой книги — подчеркнуть общий характер вычислений, а следовательно, широкую применимость компьютерных наук. Автор надеется, что эта книга возбудит в вас, читатель, более широкий интерес к информатике и желание узнать о ней побольше.

Сначала в этой книге выявляется наличие вычислений в повседневной деятельности, а затем соответствующие понятия компьютерных наук поясняются на примерах известных сказок. Решения повседневных задач берутся из распорядка

¹ В контексте книги перевод английского термина *computation* следует рассматривать не только как привычные математические *вычисления*. Словарь дает следующее определение этого термина: “1. Применение последовательности операций к значению для получения другого значения. 2. Процесс обработки данных с преобразованием входных значений (сигналов) в выходные, причем эта концепция не ограничивается данными, представленными в цифровой или иной электронной форме: биологические, химические, оптические, гидравлические и даже часовые механизмы — все они выполняют вычисления определенного вида”. Именно в этой расширенной трактовке и следует воспринимать термин *вычисления* в данной книге. — *Примеч. ред.*

типичного рабочего дня: пробуждения утром, завтрака, поездки на работу, событий на рабочем месте, приема у врача, занятий любимым хобби после полудня, ужина и размышлений над событиями прошедшего дня вечером. Небольшие истории обо всем этом представлены в отдельных пятнадцати главах книги, где понятия вычислений поясняются также на примерах семи известных историй. Каждая история охватывает две или три главы и относится к конкретному вопросу информатики.

Эта книга делится на две части, посвященные *алгоритмам и языкам*. Это главные опоры, на которых покоится понятие вычисления. В табл. 1 перечислены все истории и понятия информатики, которые они иллюстрируют.

Все мы отдаем должное интересной истории. Такие истории нас утешают, дают нам надежду и вдохновляют нас. Они рассказывают нам о мире, позволяют осознать задачи, с которыми нам приходится иметь дело, а иногда и предлагают их решения. Истории могут также служить руководством в нашей жизни. Если поразмыслить над тем, чему истории должны нас научить, то в нашем уме, вероятно, возникнет мысль о любви, конфликте, судьбе человека. Но в моем уме возникает также мысль о вычислениях. Когда Джульетта спрашивает “Что значит имя?”², она задает очень важный вопрос о представлении информации. В своем философском эссе “*Миф о Сизифе*” Альбер Камю поднимает вопрос не только о том, как бороться с абсурдностью жизни, но и как выявить бесконечное вычисление.

Таблица 1. Истории и темы книги

<i>История</i>	<i>Главы</i>	<i>Тема</i>
<i>Часть I</i>		
Гензель и Гретель	1, 2	Вычисления и алгоритмы
Шерлок Холмс	3, 4	Представления и структуры данных
Индиана Джонс	5, 6, 7	Решение задачи и его ограничения
<i>Часть II</i>		
Над радугой	8, 9	Язык и смысловое значение
День сурка	10, 11	Управляющие структуры и циклы
Назад в будущее	12, 13	Рекурсия
Гарри Поттер	14, 15	Типы и абстракция

У историй многоуровневое смысловое значение, и зачастую они содержат (иногда скрытый) вычислительный уровень. В этой книге предпринята попытка раскрыть данный уровень и предложить читателям взглянуть на известные истории по-новому с вычислительной точки зрения. Надеюсь, что читатели по достоинству оценят вычислительное содержимое представленных в этой книге историй, и эта новая точка зрения возбудит в них интерес к компьютерным наукам.

² Шекспир, “Ромео и Джульетта”, пер. Т. Щепкина-Куперник.

Благодарности

Мысль написать эту книгу возникла в ходе многих бесед с друзьями, студентами, коллегами и даже с людьми в автобусе по дороге на работу. Благодарю всех моих собеседников за то терпение, с которым они слушали мои пояснения сущности компьютерных наук, а также за их дружеское нетерпение, когда мои пояснения становились слишком пространными и сложными. Цель написания этой общедоступной книги об информатике была в значительной степени подкреплена опытом этих бесед.

За последнее десятилетие у меня была возможность работать со студентами многих вузов во время их летней практики, что послужило дополнительным стимулом для написания этой книги. Эти летние практики были поддержаны несколькими грантами от Национального научного фонда (National Science Foundation), и я благодарен за такую поддержку научных исследований и обучения наукам в Соединенных Штатах.

В поисках материала для этой книги я полагался, в частности, на Интернет, Википедию (wikipedia.org) и веб-сайт TV Tropes (tvtropes.org). Благодарю авторов всех этих материалов за их преданность делу и неумное желание делиться своими знаниями с остальным миром.

В процессе написания этой книги отдельные главы прочитали Эрик Уолкингшоу (Eric Walkingshaw), Пол Калл (Paul Cull) и Карл Шмельцер (Karl Smeltzer), которые дали профессиональные отзывы на прочитанное содержимое и стиль его изложения. Благодарю их за полезные советы, а также Дженнифер Пархем-Мочелло (Jennifer Parham-Mocello) не только за чтение отдельных глав, но и за проверку нескольких примеров вместе с ее учащимися первого курса колледжа. Благодарю также моего сына Александра за корректуру рукописи этой книги и дельные советы по вопросам, связанным с историей Гарри Поттера. Большая часть этой книги была написана во время, свободное от лекций в Университете штата Орегон. Благодарю руководство моей кафедры и университета за поддержку данного проекта.

Воплотить идею этой книги в жизнь оказалось намного труднее, чем я предполагал. Выражаю искреннюю признательность Мари Лафкин-Ли (Marie Lufkin-Lee), Катрин Альмейде (Katherine Almeida), Кэтлин Хенсли (Kathleen Hensley) и Кристин Саваж (Christine Savage) из издательства MIT Press за поддержку данного проекта и помощь, оказанную мне в ходе работы над ним.

И наконец, мне выпало счастье быть супругом самого терпеливого и откровенного читателя. Поэтому благодарю мою жену, Аню, за моральную поддержку на протяжении всего времени написания этой книги. Она всегда живо откликлась на мои вопросы, которые чаще всего были довольно нудными и абстрактными. Она прочитала немало черновиков и терпеливо старалась направлять мой стиль письма в нужное русло, когда он становился слишком академическим и чрезмерно испещренным техническим жаргоном. Завершением работы над этой

книгой я обязан в большей степени своей жене, чем кому-нибудь еще, поэтому я и посвящаю данную книгу ей.

Введение

Сегодня вычисления играют заметную роль в обществе. Но если только вы не стремитесь стать специалистом по информатике, то зачем вам эта книга и зачем что-то знать об этой науке? Ведь вы можете просто пользоваться с выгодой для себя технологиями, приводимыми в действие вычислениями. Вряд ли вам нужно разбираться в бортовом электронном оборудовании, чтобы воспользоваться самолетом, или стать дипломированным врачом, чтобы извлекать пользу из достижений современного здравоохранения.

Мир, в котором мы живем, состоит не только из изобретенных человеком технологий. Нам по-прежнему приходится взаимодействовать с неавтономными объектами, действующими по физическим законам. Поэтому понимать основы и механизмы действия этих объектов полезно просто для того, чтобы предвидеть их поведение и безопасно перемещаться в своей среде. То же самое можно сказать и о пользе от изучения как самих вычислений, так и связанных с ними понятий. Вычисления обнаруживаются не только в компьютерах и электронных устройствах, но и за пределами вычислительных машин. Ниже приводится краткое описание основных принципов информатики и поясняется их назначение.

Вычисление и алгоритмы

Предлагаю вам выполнить следующее упражнение. Для этого вам понадобятся линейка, карандаш и лист бумаги в клеточку. Проведите сначала горизонтальную линию длиной в 1 дм, а затем — вертикальную линию той же длины перпендикулярно первой от одного из ее концов. И наконец, соедините оба свободных конца проведенных линий диагональю, чтобы сформировать прямоугольный треугольник. А теперь измерьте длину проведенной диагонали. Примите мои поздравления: вы только что вычислили квадратный корень из 2, как показано на рис. 1.

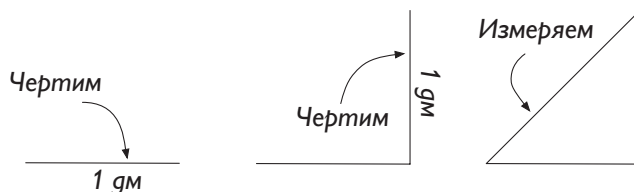


Рис. 1. Вычисление квадратного корня из 2 карандашом и линейкой

Но какое отношение это упражнение в геометрии имеет к вычислениям? Как поясняется в главах 1, “Путь к пониманию вычислений”, и 2, “От слов к делу: когда действительно происходит вычисление”, выполнение алгоритма на компьютере

приводит к вычислениям. В данном упражнении вы выступили в роли компьютера, выполнившего алгоритм черчения линий и измерения их длин, что привело к вычислению квадратного корня из 2. Наличие алгоритма имеет решающее значение, поскольку лишь в этом случае на разных компьютерах можно повторно выполнять вычисления в разное время. Важная особенность вычисления состоит в том, что для его проведения требуются ресурсы (например, карандаш, бумага и линейка) и время. И здесь очень важно описание алгоритма, поскольку оно помогает проанализировать требования к ресурсам для вычислений.

В главах 1 и 2 поясняется следующее:

- что такое алгоритмы;
- что алгоритмы применяются для систематического решения задач;
- что для проведения вычислений алгоритмы должны выполняться компьютером (человеком, машиной и т.д.);
- что для выполнения алгоритмов требуются определенные ресурсы.

Почему это так важно?

Примерами алгоритмов служат рецепты. Всякий раз, когда вы делаете сэндвич, печете шоколадный пирог или готовите свое любимое блюдо, следуя инструкциям рецепта, вы, по существу, выполняете алгоритм превращения исходных ингредиентов в конечный продукт. К числу ресурсов, которые для этого требуются, относятся ингредиенты, орудия труда, энергия и время приготовления.

Зная алгоритмы, мы можем точнее ответить на вопросы, касающиеся правильности выбранного метода и требований к его ресурсам. Такое знание помогает нам выявить возможности для улучшения процессов во всех областях жизни через (ре)организацию отдельных стадий и материалов. Например, при вычислении квадратного корня геометрическим способом можно было бы опустить рисование диагональной линии, просто измерив расстояние между двумя несоединенными конечными точками.

В приготовлении пищи улучшение может состоять в простой и очевидной экономии времени на посещение холодильника, если заранее предусмотреть и подготовить необходимые ингредиенты. Кроме того, вы можете заранее продумать, как эффективнее пользоваться своей печкой или плитой и сэкономить время на распараллеливании стадий процесса приготовления пищи, предварительно разогревая печку или плиту или промывая салат, пока готовится картофель. Аналогичные методики применяются и во многих других областях: от простых инструкций по сборке мебели до процессов организации работы конторы или производственного цеха.

В области технологий алгоритмы, по существу, управляют всеми вычислениями на компьютерах. Характерным тому примером служит сжатие данных, без которого передавать музыку и фильмы через Интернет было бы практически

невозможно. Алгоритм сжатия выявляет в данных часто встречающиеся образцы и заменяет их небольшими кодами. Сжатие данных позволяет решить задачу ресурсов для вычислений, сокращая память, требующуюся для представления музыкальных композиций и фильмов, а следовательно, и время, затрачиваемое на их загрузку через Интернет. Еще одним характерным примером служит принятый в компании Google алгоритм составления рейтинга веб-страниц, определяющий порядок представления полученных результатов пользователю. По этому алгоритму важность веб-страницы оценивается путем подсчета количества ссылок на нее с учетом веса, обозначающего важность этих ссылок.

Представление и структуры данных

Следует ожидать, что числовые вычисления выполняются с помощью чисел, поскольку мы пользуемся индо-арабской системой счисления, а вычислительные машины оперируют нулями и единицами. Таким образом, геометрический способ вычисления квадратного корня из 2 может показаться необычным. Но данный пример вычисления наглядно демонстрирует, что одно и то же понятие (например, количество) можно представить разными способами (числовыми знаками или отрезками).

Сущность вычисления состоит в преобразовании представления. В главе 3, “Тайна знаков”, поясняется, что такое представление и как они употребляются в вычислениях. Многие вычисления оперируют большими объемами информации, и поэтому в главе 4, “Записная книжка сыщика”, поясняется, как эффективно организовать коллекции данных. Этот вопрос усложняется тем, что в любой конкретной организации данных одни формы эффективного доступа к данным могут поддерживаться, а другие формы — не поддерживаться.

В главах 3 и 4 обсуждаются следующие вопросы:

- формы представления;
- способы организации коллекций данных и доступа к ним;
- преимущества и недостатки видов организации данных.

Почему это так важно

Ингредиенты в рецептах могут измеряться по весу или объему. Это разные формы представления, требующие разной кухонной утвари (весов или мерных чашек) для правильного исполнения рецепта или алгоритма. Что же касается организации данных, то способ хранения продуктов в холодильнике или кладовой оказывает влияние на быстроту извлечения ингредиентов, требующихся по рецепту. С другой стороны, можно рассмотреть вопрос представления применительно к самому рецепту. В частности, рецепт может быть представлен в виде последовательного ряда рисунков или даже в форме видеозаписи, выложенной в YouTube. Эффективность алгоритмов нередко зависит от выбора способа представления.

С одной стороны, вопрос организации коллекции элементов или людей находит немало примеров применения, включая наведение порядка на рабочем столе или в гараже, помогающее быстрее находить нужные вещи, или организация книжных полок в библиотеке. С другой стороны, можно рассмотреть разные случаи ожидания в очереди: в продовольственном магазине (стоя в очереди), приемной врача (сидя в ожидании вызова вашего номера) или же при посадке на самолет (несколько очередей).

В области информатики к самым удачным инструментальным средствам программирования относятся электронные таблицы. Этой удачей они во многом обязаны организации данных в табличной форме, поскольку она поддерживает быстрое и простое формулирование сумм по строкам и столбцам, а также представляет данные и результаты вычислений в одном месте. С другой стороны, Интернет является одним из самых преобразовывающих изобретений конца XX века, позволивших организовать веб-страницы, компьютеры и соединения между ними в сеть. Такое представление поддерживает гибкий доступ к информации и эффективную передачу данных.

Решения задач и их ограничения

Алгоритм — это средство решения задач, будь то извлечение квадратного корня из числа или выпекание пирога. Информатика — это научная дисциплина, занимающаяся систематическим решением задач.

Среди многих задач, которые могут быть решены с помощью алгоритмов, подробного рассмотрения заслуживают две задачи. В главе 5, “Поиск идеальной структуры данных”, поясняется задача поиска — один из наиболее часто выполняемых видов вычислений данных, а в главе 6, “Сортировка алгоритмов сортировки” — задача сортировки (также в ней демонстрируется эффективный способ ее решения и рассматриваются трудности, присущие этой задаче). В главе 7, “Трудноразрешимые задачи”, описывается класс так называемых трудноразрешимых задач. Алгоритмы существуют и для этих задач, но для их выполнения требуется слишком много времени, и поэтому на практике их можно считать неразрешимыми.

В главах 5–7 проясняются следующие вопросы:

- причины, по которым поиск может быть трудным и долгим;
- методы усовершенствования поиска;
- алгоритмы сортировки;
- что в вычислениях одного вида могут поддерживаться другие виды, например сортировка может поддерживать поиск;
- что алгоритмы со временем выполнения, изменяющимся по экспоненциальному закону, на самом деле считаются решениями задач не могут.

Почему это так важно

Мы тратим бесчисленные часы своей жизни на поиски, будь то ключи от автомашины или сведения в Интернете. Поэтому полезно разбираться в поиске и знать методики, способные помочь сделать его более эффективным. Кроме того, задача поиска демонстрирует, каким образом выбор способа представления оказывает влияние на эффективность алгоритмов, отражая следующее наблюдение Джона Дьюи (John Dewey): “Правильно поставленная задача означает половину ее решения” [1]³.

Знать, когда задачу *нельзя* решить эффективно, так же важно, как и знать алгоритмы для решаемых задач, поскольку это помогает нам не заниматься поиском эффективных решений там, где их не существует. Это предполагает, что в некоторых случаях мы должны довольствоваться приблизительными ответами.

В области информатики наиболее очевидным примером поиска служат механизмы поиска в Интернете, например в Google. Результаты поиска по запросу представлены не в произвольном порядке, а, как правило, отсортированными по их предполагаемой важности или релевантности. Знание трудности задач используется при разработке алгоритмов, вычисляющих приблизительные решения там, где вычисление точных решений отнимает слишком много времени. Одним из известных примеров служит задача коммивояжера, которая состоит в определении кругового маршрута для посещения определенного количества городов, который минимизирует общее пройденное расстояние.

Знанием об отсутствии эффективного алгоритма решения задачи можно эффективно воспользоваться и положительным образом. Одним из известных тому примеров служит шифрование с открытым ключом, обеспечивающее закрытые транзакции в Интернете, включая ведение банковских счетов и совершение покупок в оперативном режиме. Но такое шифрование действует лишь потому, что в настоящее время не существует эффективного алгоритма, обеспечивающего разложение большого числа на простые сомножители (т.е. дающего представление числа в виде произведения простых чисел). Если это положение изменится, то шифрование с открытым ключом станет ненадежным.

Языки: смысловое значение

Любой алгоритм может быть выражен на каком-нибудь языке. Современные компьютеры нельзя запрограммировать на человеческом языке, поскольку естественные языки содержат слишком много неоднозначностей, с которыми могут легко справиться люди, но не компьютеры. Следовательно, алгоритмы, которые предполагается выполнять на вычислительных машинах, должны быть написаны на языках, имеющих вполне определенную структуру и смысловое значение.

³ Здесь следует упомянуть еще одну мысль, на этот раз — фантаста Роберта Шекли из рассказа “Верный вопрос”: “Чтобы правильно задать вопрос, нужно знать большую часть ответа”. — *Примеч. ред.*

В главе 8, “Сквозь призму языка”, поясняется, что такое язык и как мы можем определить его синтаксис. Благодаря определению синтаксиса языка обеспечивается вполне определенная структура каждого из его предложений, и на этом основании удастся понять и определить смысловое значение предложений и языков. А в главе 9, “Поиск нужного тона: смысл звука”, обсуждаются смысловое значение языков и проблема неоднозначности.

В главах 8 и 9 описывается:

- каким образом грамматика определяет язык;
- как с помощью грамматики построить все предложения, принадлежащие языку;
- что такое синтаксические деревья;
- каким образом синтаксические деревья представляют структуру предложений и разрешают неоднозначность в смысловом значении предложений.

Почему это так важно

Мы употребляем языки для передачи смыслового значения. А для того чтобы такая передача работала, общающиеся стороны должны прийти к согласию относительно того, что следует считать истинным предложением и что означает каждое предложение. Например, инструкции в рецептах должны быть точны в отношении мер измерений продуктов, температуры в печке, времени приготовления и так далее, чтобы получить желаемые результаты, а не нечто несъедобное и обгорелое.

В большинстве областей нашей жизни мы выработали специальную терминологию и языки, облегчающие общение и делающие его более эффективным. Это особенно справедливо в отношении информатики, где наиболее важная часть общения происходит через вычислительные машины. А поскольку вычислительные машины обладают более скромными способностями в смысле обработки естественных языков, чем люди, то очень важно дать точное определение языка, чтобы обеспечить желаемое поведение запрограммированных вычислительных машин.

В области информатики широко распространен такой язык программирования, как язык формул электронных таблиц. Всякий, кому доводилось когда-нибудь вводить формулу в электронную таблицу, уже написал программу для электронных таблиц. Но электронные таблицы печально известны тем, что они порой действуют ошибочно и это приводит к убыткам в миллиарды долларов из-за неточных формул. Еще одним распространенным языком является HTML (язык разметки гипертекста). Всякий раз, когда вы загружаете веб-страницу на свой персональный или переносной компьютер или же сотовый телефон, то, вероятнее всего, содержимое этой страницы представлено в вашем браузере на языке HTML, ясно выражающем структуру веб-страницы и однозначно ее

представляющем. И если язык HTML служит лишь для представления информации и сам не описывает вычисления, то динамическое поведение веб-страниц определяется на JavaScript — другом языке, понятном любому современному браузеру и созданном специально для возможности динамического поведения веб-страниц.

Управляющие структуры и циклы

Инструкции выполняют в алгоритме две разные функции: непосредственно манипулируют данными и решают, какие именно инструкции должны быть выполнены следующими и как часто это должно происходить. Последняя разновидность инструкций называется управляющими структурами. Подобно тому, как сюжет фильма или рассказа связывает отдельные действия и сцены в слаженное повествование, управляющие структуры составляют алгоритмы из отдельных инструкций.

В главе 10, “Намылить, смыть, повторить”, поясняются различные управляющие структуры с особым акцентом на циклы, с помощью которых выражается повторение действий. А в главе 11, “Счастливый конец не гарантируется”, обсуждается важный вопрос: завершается ли цикл или он выполняется бесконечно и можно ли выяснить это с помощью алгоритма.

В главах 10 и 11 обсуждаются следующие вопросы:

- что такое управляющие структуры;
- почему управляющие структуры играют решающую роль в любом языке для выражения алгоритмов;
- каким образом повторяющиеся задачи могут быть выражены с помощью циклов;
- что такое задача останова и каким образом она иллюстрирует основополагающее свойство вычислений.

Почему это так важно

Прежде чем печь блины, нужно смазать маслом сковородку. Порядок следования стадий приготовления пищи в рецептах имеет особое значение. Более того, рецепты иногда содержат решения, которые принимаются, исходя из свойств имеющихся ингредиентов или употребляемой кухонной утвари. Так, если вы пользуетесь конвекционной печью, то должны сократить время или понизить температуру выпекания или же сделать и то, и другое. Рецепт содержит цикл, если он предписывает выполнять повторяющиеся действия, например добавлять яйцо и взбивать смесь ингредиентов для приготовления кекса.

Отличие управляющих структур от других операций, по существу, означает различие между каким-то действием и организацией порядка и частоты его выполнения. Для выполнения любого процесса или алгоритма требуется знать,

делает ли он именно то, что должен делать, а еще проще — завершится ли он вообще. Этот довольно простой вопрос, который ставит проблема останова, служит лишь одним из примеров многих свойств алгоритмов, о которых хотелось бы знать. Зная, какие именно свойства одних алгоритмов могут быть автоматически определены с помощью других алгоритмов, можно выяснить пределы их досягаемости и ограничения вычислений.

В технологических областях управляющие структуры применяются там же, где и алгоритмы, т.е. везде. Любая информация, посылаемая через Интернет, передается циклически, повторяясь до тех пор, пока не будет правильно принята. Светофор работает под управлением бесконечно повторяющихся циклов, а многие производственные процессы состоят из заданий, которые повторяются до тех пор, пока не будет достигнута требуемая мера качества. Прогнозирование поведения алгоритмов в отношении неизвестных данных, которые могут быть введены впоследствии, имеет немало приложений в области безопасности. Например, хотелось бы заранее знать, уязвима ли система к атакам злоумышленников. То же самое применимо и к роботам-спасателям, которые должны использоваться в ситуациях, отличающихся от тех, в которых они обучены. Точность прогнозирования поведения робота в неизвестных ситуациях может, по существу, означать жизнь или смерть.

Рекурсия

Принцип сведения, т.е. процесс объяснения или построения системы из более простых частей, играет важную роль в большей части науки и техники. Рекурсия — это особая форма сведения, обращающегося к самому себе. Многие алгоритмы рекурсивны. Примером тому служат следующие инструкции для поиска слова в словаре, где на каждую статью приходится одна страница: “Открыть словарь. Если слово найдено, прекратить его поиск. В противном случае искать слово в части словаря, предшествующей или следующей после текущей страницы”. Обратите внимание на то, что инструкция для поиска в последнем предложении означает рекурсивную ссылку на весь процесс, который возвращает к началу инструкций. Это избавляет от необходимости вводить в описание нечто вроде “повторять до тех пор, пока слово не будет найдено”.

В главе 12, “Своевременный стежок вычисляется впрок”, поясняется рекурсия, которая представляет собой управляющую структуру, которая применяется также для определения организации данных. А в главе 13, “Все дело в интерпретации”, представлены разные подходы к пониманию рекурсии.

В главах 12 и 13 исследуется следующее:

- понятие рекурсии;
- различие между разными формами рекурсии;

- два метода раскрытия и осмысления рекурсивных определений;
- каким образом эти методы помогают понять рекурсию и взаимосвязь между разными ее формами.

Почему это так важно

Инструкции “приправить по вкусу” можно дать следующее рекурсивное определение: “Попробовать блюдо. Если оно приятно на вкус, на этом остановиться. В противном случае добавить щепотку специй, а затем приправить по вкусу”. Любое повторяющееся действие может быть описано рекурсивно с помощью повторяемого действия (в данном случае — “приправить по вкусу”) в его определении и условия, когда следует остановиться.

Рекурсия является важным принципом для получения конечных описаний потенциально бесконечных данных и вычислений. В грамматике языка рекурсия упрощает бесконечное количество предложений, а рекурсивный алгоритм позволяет обрабатывать входные данные произвольного объема.

Рекурсия представляет собой общую управляющую структуру и механизм организации данных и поэтому является неотъемлемой частью многих программных систем. Впрочем, рекурсия находит и ряд других непосредственных применений. Например, эффект Дросте (Droste), означающий наличие в изображении его более мелкого варианта, может быть получен в результате образования петли обратной связи между сигналом (изображением) и получателем (фотокамерой). Петля обратной связи служит рекурсивным описанием повторяющегося эффекта. Фракталы являются самоподобными геометрическими рисунками, которые могут быть описаны с помощью рекурсивных уравнений. В природе примерами фракталов могут служить снежинки и кристаллы. Фракталы применяются также при анализе белков и структур ДНК. Кроме того, фракталы применяются в нанотехнологиях для проектирования самособирающихся наносхем. В настоящее время проводятся исследования самовоспроизводящихся машин с точки зрения их применимости для освоения космоса.

Типы и абстракция

Вычисления состоят в преобразовании представлений. Но не всякое преобразование применимо к каждому представлению. Так, хотя можно умножать числа, умножать отрезки нельзя. И аналогично можно вычислить длину линии или площадь прямоугольника, но для числа это не имеет смысла.

Представления и преобразования могут быть разделены на группы с целью упростить отличие целесообразных преобразований от нецелесообразных. Эти группы называются типами, а правила, определяющие допустимые сочетания преобразований и представлений, — правилами типизации. Типы и правила типизации поддерживают разработку алгоритмов. Так, если требуется вычислить

число, то следует выполнить операцию, производящую числа. А если требуется обработать список чисел, то придется выполнить операцию, принимающую в качестве входных данных список чисел.

В главе 14, “Волшебный тип”, поясняется, что такое типы и как можно пользоваться ими для формулирования правил, описывающих закономерности вычислений. Такие правила могут быть использованы для обнаружения ошибок в алгоритмах. Истинный потенциал типов лежит в их способности пренебрегать подробными сведениями об отдельных объектах, а следовательно, правила могут быть сформулированы на более общем уровне. Процесс пренебрежения подробностями называется абстракцией, которой посвящена глава 15, “С высоты птичьего полета: от абстракции к деталям”, в ней поясняется, почему абстракции принадлежит центральное место в информатике и как она применяется не только в типах, но и в алгоритмах и даже в языках и компьютерах.

В главах 14 и 15 обсуждаются следующие вопросы:

- что такое типы и правила типизации;
- каким образом они могут быть использованы для описания законов вычислений таким образом, чтобы это помогало выявлять ошибки в алгоритмах и составлять надежные алгоритмы;
- что типы и правила типизации являются лишь особым случаем более общей концепции понятия абстракции;
- что алгоритмы служат абстракцией вычисления;
- что типы служат абстракцией представлений;
- что трудности, возникающие во время выполнения, служат абстракцией времени исполнения.

Почему это так важно

Если по рецепту требуется открыть консервную банку с бобами, то вы удивитесь, если кто-нибудь попытается решить эту задачу ложкой. Ведь это нарушило бы правило типизации, согласно которому ложка вряд ли пригодна для решения данной задачи.

Типы и прочие абстракции для описания правил и процессов применяются повсеместно. Любая процедура, которая должна быть повторена, предполагает алгоритмическую абстракцию, т.е. описание, пренебрегающее излишними подробностями и заменяющее переменные части параметров. Рецепты также содержат алгоритмические абстракции. Так, во многих кулинарных книгах содержится раздел, описывающий основные приемы приготовления пищи, например снятие кожуры и очистка помидоров от семян, а в самих рецептах делается ссылка на этот раздел, когда в них указывается, сколько требуется помидоров,

В сфере технологий существует множество примеров типов и абстракций. Примерами физических типов являются все разновидности электровилок и розеток; винтов, отверток и бит шуруповертов; замков и ключей. Разные формы предназначены для предотвращения неправильного сочетания инструментов. В программном обеспечении примеры типов можно найти в веб-формах, которые требуют ввода номера телефона или адреса электронной почты в определенном формате. Существует множество примеров дорогостоящих ошибок, которые были вызваны игнорированием типов. Например, в 1998 году NASA потеряло спутник Mars Climate Orbiter стоимостью 655 миллионов долларов из-за несовместимых представлений чисел. Это была ошибка типа, которую можно было бы предотвратить при применении системы типов. Наконец, понятие компьютера само является абстракцией людей, машин или других субъектов, которые способны выполнять алгоритмы.

На рис. 2 даны общее схематическое представление понятий, обсуждаемых в этой книге, а также взаимосвязи между ними. В главах 7, 11 и 13, показанных в темно-серых прямоугольниках на рис. 2, представлено более формальное описание этих понятий. Эти главы можно пропустить, поскольку их чтение необязательно для понимания материала остальной книги.

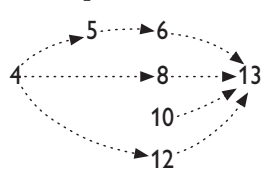


Несмотря на то что материал этой книги упорядочен определенным образом, ее совсем необязательно читать именно в таком порядке. Многие ее главы можно читать независимо, даже если в последующих главах иногда делаются ссылки на понятия и примеры, представленные в предыдущих главах.

Ниже даются рекомендации по выбору глав и порядка их изучения, а также точки выхода и сокращения, дающие читателям возможность продвигаться вперед, читая избранные главы. Когда понятия вычисления обсуждаются в этой книге с использованием событий, людей и объектов, возникающих в историях, то иногда вводится новое обозначение и на конкретных примерах более подробно демонстрируются важные особенности. Поэтому одни части этой книги читать легче, чем другие. Как читатель многих научно-популярных книг я хорошо осознаю, что у читателей этой книги может быть разный интерес к таким подробностям. Именно поэтому я и надеюсь, что приведенные здесь рекомендации помогут читателям лучше ориентироваться в содержимом книги.

Сначала рекомендуется прочитать главы 1 и 2, поскольку в них представлены такие основополагающие понятия вычисления, упоминаемые далее в этой книге, как алгоритм, параметр, компьютер и временная сложность. Усвоить материал этих двух глав будет нетрудно.

Остальные темы в светло-серых прямоугольниках на рис. 2 в основном не зависят одна от другой, но главы в пределах каждой темы следует читать по порядку. Так, в главе 4 представлен ряд структур данных, и поэтому ее следует прочитать прежде глав 5, 6, 8, 12 и 13, как показано на рисунке.



И наконец, в словаре специальных терминов, приведенном в конце книги, представлены дополнительные сведения о взаимосвязи глав книги. С этой целью определения понятий сгруппированы по темам и перекрестным ссылкам на статьи о них в словаре.