

Посвящается Декстеру,  
который, я знаю, увлеченно  
прочтет эту книгу от корки до корки,  
прежде чем ему исполнится восемь.  
О, какие места ты найдешь!

# Предисловие

**Д**анная книга представляет собой введение в область компьютерных наук. В ней необходимая широта обзора предмета сочетается с достаточно глубоким проникновением в сущность излагаемого материала.

## Аудитория

Мы писали эту книгу для двух категорий читателей: для будущих специалистов по компьютерным наукам и для студентов всех других дисциплин. Первая категория читателей, включающая студентов начальных курсов, которые по окончании обучения станут специалистами в области компьютерных наук, как правило, на этом этапе обучения склонна отождествлять весь спектр компьютерных наук с программированием, просмотром веб-страниц и обменом файлами в Интернете, поскольку это, в сущности, именно то, что они видели раньше и с чем им приходилось сталкиваться. Однако область компьютерных наук — это нечто существенно большее. Поэтому студентам-компьютерщикам необходимо продемонстрировать всю глубину и обширность той области знаний, к изучению которой они приступают и в которой планируют специализироваться. В предоставлении этой столь необходимой им информации и состоит назначение данной книги. Она познакомит студентов с обзором всего спектра компьютерных наук, что даст основу для правильной оценки различных курсов, которые им предстоит изучать впоследствии. Такой исследовательский подход в действительности является типовой моделью построения вводных курсов в любой области естественных наук.

Столь широкий охват темы — это также именно то, в чем нуждаются студенты других специальностей, если они намерены стать полноправной частью нынешнего техногенного общества, в котором живут. Курс компьютерных наук для этой аудитории должен помочь ей достичь практического, реалистичного понимания того, что представляет собой все обширное поле этой области знаний, а не быть элементарным ознакомлением с тем, как пользоваться Интернетом, или приобретением минимальных практических навыков по

использованию некоторых пакетов программного обеспечения, наиболее популярных в той или иной области. Разумеется, свое место есть и для подобного обучения, однако данная книга предназначена для предоставления учащимся прочных основ компьютерного образования в целом, которые будут работать на них и в будущем.

При подготовке всех предыдущих изданий этой книги сохранение доступности материала для студентов не технических дисциплин было одной из важнейших наших целей. В результате этот учебник уже не раз успешно использовался как базовый в соответствующих курсах для студентов самого широкого круга дисциплин и уровня образования от высшей школы до выпускников обычных школ. Данное, 13-е, издание книги было подготовлено с соблюдением этой традиции.

## Что нового в 13-м издании

В 13-м издании продолжена практика использования для записи примеров программного кода на языке Python и псевдокода, выдержанного в том же стиле, впервые принятая в 12-м издании. Это изменение было введено по нескольким причинам. Прежде всего, эта книга уже содержит немало примеров программного кода на различных языках, включая детализированный псевдокод в нескольких главах. Принимая во внимание, что при работе с книгой читателям потребуется усвоить достаточное количество программного синтаксиса, разумно сделать так, чтобы этот синтаксис отвечал тому языку, с которым им с большей вероятностью придется встретиться в последующих курсах. Что еще более важно, все большее число преподавателей, использующих этот учебник, приходят к заключению, что даже в простейших вводных курсах по использованию компьютеров их слушателям сложно усвоить многие темы при отсутствии программного инструмента для проведения самостоятельных исследований и экспериментов.

Почему именно Python? Выбор языка — это всегда спорный вопрос; любое ваше решение вызовет возмущение у одних и восхищение у других, причем примерно в равной пропорции. Язык Python — это отличный выбор среднего уровня, обладающий такими достоинствами, как:

- четкий, понятный и простой в изучении синтаксис;
- простые примитивы ввода-вывода;
- типы данных и управляющие структуры, очень близкие к примитивам псевдокода, использовавшегося в предыдущих изданиях книги;
- поддержка многих парадигм программирования.

Это зрелый язык с активным сообществом разработчиков и обширным полем онлайн-ресурсов для дальнейшего изучения. Язык Python долгое время сохраняет свои позиции в пятерке наиболее широко используемых языков в индустрии программного обеспечения по многим показателям, и в последнее время наблюдается быстрый рост его использования во вводных курсах компьютерных наук. Он особенно популярен в начальных курсах для непрофильных специалистов в самых разных областях естественных наук, таких как физика и биология. Следует добавить, что язык Python сейчас очень широко используется при создании программного обеспечения для проведения различных научных расчетов в этих областях.

Хотя главный акцент в данной книге сделан на основных концепциях самого широкого диапазона компьютерных наук, использование языка Python также позволяет читателю лучше ощутить вкус программирования, чем это было возможно в предыдущих изданиях, и в то же время эта книга вовсе не является полномасштабным введением в программирование на этом языке. Обсуждение отдельных тем по языку Python организовано в соответствии с существующей структурой книги. Так, в главе 1 обсуждается синтаксис представления в Python данных — целых чисел, чисел с плавающей запятой, символов в коде ASCII и строк в кодах Unicode. В главе 2 представлены операции языка Python, которые соответствуют машинным примитивам, обсуждаемым в остальных частях этой главы. Условные операторы, операторы циклов и функции рассматриваются в главе 5, поскольку именно здесь эти конструкции необходимы для реализации достаточно полного псевдокода, предназначенного для записи алгоритмов. Короче говоря, в книге конструкции языка Python используются лишь для подкрепления рассматриваемых концепций компьютерных наук и во все не являются предметом обсуждения.

Как и в предыдущих изданиях, в книгу включены некоторые “изюминки”: в ее тексте размещены врезки, содержание которых позволяет лучше осознать связь излагаемого материала с реальным миром. Многие из этих врезок содержат ссылки на источники в Интернете, предоставляющие дополнительную информацию по обсуждаемой теме.

И наконец, при подготовке нового издания содержимое каждой главы было подвергнуто тщательному пересмотру, были внесены необходимые обновления и изменения, а также исправлены ошибки, выявленные в предыдущем издании.

## Структура книги

Материал книги упорядочен в соответствии с восходящим (по уровню сложности) подходом, предусматривающим переход от конкретного к абстрактному — именно такой способ изложения обеспечивает ясную и доступную подачу материала, когда одна тема плавно переходит в другую.

Книга начинается с фундаментальных положений в отношении кодирования информации, хранения данных и архитектуры вычислительных машин (главы 1 и 2); далее рассматриваются построение и функционирование операционных систем (глава 3) и компьютерных сетей (глава 4); изучаются такие темы, как алгоритмы, языки программирования и различные аспекты разработки программного обеспечения (главы 5–7); исследуются технологии повышения доступности информации (главы 8 и 9); рассматриваются некоторые важнейшие приложения и технологии компьютерной графики (глава 10) и искусственного интеллекта (глава 11); а в завершение предлагается введение в абстрактную теорию вычислений (глава 12).

Хотя книга в целом выдержана с соблюдением выбранного восходящего стиля изложения, отдельные ее главы и разделы на удивление независимы и вполне могут изучаться как самостоятельные единицы или произвольным образом перепорядочиваться в выбранной альтернативной последовательности изложения материала. В действительности книга часто использовалась как базовый учебник в курсах, материал в которых подавался в ином порядке. В одном случае изложение материала начиналось с глав 5 и 6, после чего мог выполняться переход к главам 1–4. Мне известен и такой курс, который начинался с абстрактной теории вычислений (глава 12). В иных случаях этот учебник использовался в качестве “отправной точки” и служил лишь своеобразной основой, от которой отходили ответвления в углубленное рассмотрение различных тем в тех или иных областях. С другой стороны, курсы для менее технически ориентированной аудитории могли ограничиваться лишь материалом глав 4 (сети и Интернет), 9 (системы баз данных), 10 (компьютерная графика) и 11 (искусственный интеллект).

На первой странице каждой главы приводится перечень рассматриваемых в ней тем — в этом списке звездочками отмечены такие темы, которые можно считать не обязательными для изучения, а лишь дополнительным, факультативным материалом. В каждой главе это разделы, в которых либо рассматриваются более специфические темы, либо предлагается более углубленное изложение уже рассмотренных тем. В наши намерения входило лишь предоставить вам свои предложения в отношении альтернативной последовательности изучения материала курса. Безусловно, могут иметь место и иные подходы к сокращению его объема. В частности, для тех, кто ищет варианты более быстрого прочтения, мы можем предложить такую последовательность изложения материала.

| Раздел    | Тема                                    |
|-----------|-----------------------------------------|
| 1.1–1.4   | Основы кодирования и хранения данных    |
| 2.1–2.3   | Архитектура компьютера и машинные языки |
| 3.1–3.3   | Операционные системы                    |
| 4.1–4.3   | Сети и Интернет                         |
| 5.1–5.4   | Алгоритмы и их создание                 |
| 6.1–6.4   | Языки программирования                  |
| 7.1–7.2   | Разработка программного обеспечения     |
| 8.1–8.3   | Представление данных                    |
| 9.1–9.2   | Системы баз данных                      |
| 10.1–10.2 | Компьютерная графика                    |
| 11.1–11.3 | Искусственный интеллект                 |
| 12.1–12.2 | Теория вычислений                       |

Существует ряд тем, которые красной нитью проходят через всю книгу. Одна из них — это то, что компьютерные науки являются весьма динамичной областью знания. Все рассматриваемые темы подаются в исторической перспективе, обсуждается достигнутый на данный момент уровень развития и указываются основные направления текущих исследований. Другая тема состоит в пояснении значения абстрактных методов и способов применения различных абстрактных инструментов для управления уровнем сложности. Эта тема вводится в главе 0, а затем вновь и вновь поднимается в контексте архитектуры операционных систем, построения компьютерных сетей, создания новых алгоритмов, разработки языков программирования, создания программного обеспечения, представления данных и компьютерной графики.

## Преподавателю

Объем материала этой книги превосходит тот, который может быть изучен за один семестр, поэтому не бойтесь пропускать темы, которые не соответствуют задачам вашего курса. Мы написали книгу для того, чтобы она служила основой для проведения курса обучения, а не определяла его содержание. Несмотря на то что изложение материала следует определенной схеме, каждая из тем подается независимо, что позволяет вам делать выбор в соответствии с вашими

вкусами. Мы также предлагаем рассматривать некоторые темы как задания для домашнего чтения. Нам кажется, что зачастую мы недооцениваем студентов, когда считаем необходимым объяснять абсолютно все непосредственно на занятиях. Я часто задаю моим студентам целую главу для чтения на дом, а затем использую время занятий для того, чтобы разъяснить определенные вопросы или подробно осветить некоторые части текста, исходя из собственного опыта. Мы прежде всего должны помочь студентам научиться учиться самостоятельно.

Я уже указывал, что книга построена по восходящему принципу “от конкретного к абстрактному”, однако позвольте мне остановиться на этом подробнее. Как ученые мы слишком часто полагаем, что студенты непременно оценят наш подход к предмету, который вырабатывался нами на протяжении многих лет работы в этой области. Однако как преподаватели мы поступим лучше, если будем подавать материал, ориентируясь на точку зрения студента. Именно поэтому книга начинается с освещения методов представления и хранения данных, архитектуры компьютеров, операционных систем и компьютерных сетей. Современные студенты наверняка уже слышали такие термины как “JPEG” и “MP3”, они знакомы с компакт-дисками и флеш-памятью, умеют работать с разными операционными системами, ежедневно пользуются смартфонами и Интернетом. Как я обнаружил на опыте, они проявляют немалый интерес к тому, что это такое и как все это работает. Начиная свой курс с этих тем, я часто наблюдал, как они находят ответы на многие свои “почему?”, после чего начинают воспринимать этот курс скорее как практический, нежели как теоретический. После такого начала вполне естественно перейти к более абстрактным темам — алгоритмам, алгоритмическим структурам, языкам программирования, методам разработки программного обеспечения, вычислимости и сложности, которые, по нашему мнению, и являются основными темами курса. Как говорилось выше, отдельные темы в книге излагаются в такой манере, которая не обязывает вас следовать выбранной нами последовательности, но мы рекомендуем вам все же попробовать применить ее на практике.

Всем нам известно, что студенты познают гораздо больше того, чему мы их собственно обучаем, и те знания, которые они получают окольным путем, зачастую воспринимаются лучше, чем те, которые подаются им непосредственно. Это становится особенно важным, когда приходит время “учить” решать задачи. Студенты не учатся решать проблемы как отдельную дисциплину путем изучения методологий по решению задач. Они учатся справляться с проблемами, решая их, а не просто заучивая типовое решение тщательно отобранных задач, приведенное в учебниках. Поэтому в эту книгу мы включили многочисленные задания и упражнения, причем некоторые из них преднамеренно сформулированы довольно расплывчато — это означает, что для них необязательно

существует единственный правильный подход или возможно только одно правильное решение. Я настоятельно рекомендую вам использовать такие упражнения и подробно пояснять методы их решения.

Другие темы, которые были отнесены к категории “неявного обучения”, — это профессионализм, этика и социальная ответственность. Мы не считаем, что подобный материал может быть представлен как отдельный предмет обсуждения, привязанный к данному курсу. Напротив, он должен выходить на поверхность только там, где это уместно, и именно такой подход выбран в данной книге. В частности, в разделы 4.5, 5.5, 7.9, 9.7 и 11.7 включены такие темы, как безопасность, конфиденциальность, ответственность, социальные аспекты, обсуждаемые в контексте операционных систем, работы в сети, разработки программного обеспечения, использования баз данных и применения искусственного интеллекта. Кроме того, каждая глава включает ряд вопросов (раздел “Общественные и социальные вопросы”), побуждающих студентов к размышлениям об отношении представленного в книге материала к жизни того общества, частью которого они являются.

Мы благодарим вас за то, что вы обратились к этой книге при подготовке собственных курсов. Независимо от того, остановите ли вы на ней свой выбор, мы надеемся, что вы расцените ее как определенный вклад в литературу, посвященную обучению компьютерным наукам.

---

## Педагогические аспекты

---

Эта книга является плодом многолетней практики преподавания, благодаря чему она богата разнообразным педагогическим материалом. В частности, весьма существенным фактором является обилие предлагаемых задач, решение которых требует активного участия обучающихся — более 1000 в этом издании. Они распределены по разделам “Вопросы и упражнения”, “Задания по материалу главы” и “Общественные и социальные вопросы”. Подразделы “Вопросы и упражнения” присутствуют в конце каждого раздела каждой главы (за исключением вводной), их назначение — стимулировать обучающихся к самостоятельному мышлению. С помощью предлагаемых задач закрепляется пройденный материал, приведенное выше обсуждение расширяется дополнительными аспектами и даются ссылки на связанный материал, который будет рассматриваться позднее. Ответы на предлагаемые здесь вопросы вынесены в приложение Е.

Разделы “Задания по материалу главы” находятся в конце каждой главы (за исключением вступительной). Они включают подборки задач, разработанных для использования в качестве домашнего задания, поскольку они относятся к содержанию всей главы. По этой же причине в книге ответы на них не приводятся.



За этими разделами следуют разделы под заголовком “Общественные и социальные вопросы”, которые содержат вопросы для обдумывания и обсуждения. Многие из этих вопросов могут быть использованы в качестве небольших заданий на проведение исследований, результаты которых должны быть представлены в виде письменных или устных отчетов.

Каждая глава завершается списком рекомендуемой литературы, включающим ссылки на материал, имеющий отношение к теме данной главы. Кроме того, хорошим источником дополнительной информации могут быть веб-сайты, упоминаемые в этом введении, в основном тексте книги и во врезках.

---

## Дополнительные ресурсы

---

Большое количество дополнительного материала для этой книги можно найти на ее веб-сайте: [www.pearsonhighered.com/brookshear](http://www.pearsonhighered.com/brookshear). На нем любой желающий может получить доступ к следующему.

- Дополнительная информация по каждой главе, дополняющая материал книги и предоставляющая возможность познакомиться со связанными и близкими темами.
- Задания для самопроверки по каждой главе, которые помогают заново обдумать тот материал, который был предложен по этой теме в книге.
- Дополнительные сведения по основам программирования на языке Python, предлагаемые в последовательности, соответствующей изложению материала в книге.

В дополнение к этому на сайте ресурсов для квалифицированных преподавателей издательства Pearson Education ([www.pearsonhighered.com](http://www.pearsonhighered.com)) вы найдете следующие материалы для этой книги.

- Руководство для преподавателей с ответами к заданиям из разделов “Задания по материалу главы” всех глав книги.
- Слайды и презентации PowerPoint для лекций.
- Банк тестов.

## Студенту

Гленн Брукшир по своему характеру является слегка инакомыслящим (некоторые его друзья считают, что более чем слегка), поэтому, приступив к написанию этой книги, он далеко не всегда следовал тем советам, которые ему давали. В частности, многие утверждали, что определенный материал был представлен на слишком высоком уровне для студентов, только начинающих обучение.

Однако мы уверены, что если тема актуальна, то она будет актуальной, даже если все академическое сообщество сочтет ее “продвинутой”. Вы заслуживаете иметь такой учебник, в котором картина компьютерных наук представлена во всей своей полноте, а не “разбавленную” его версию, содержащую сознательно упрощенное представление только тех тем, которые принято считать подходящими для изучающих вводный курс. А это значит, что авторам нельзя было избегать тех или иных тем. Вместо исключения сложной темы нам требовалось найти наилучшие способы ее разъяснения. Мы всегда стремились придать изложению необходимую глубину, предоставить вам истинную картину того, что компьютерные науки представляют собой на самом деле. Как и в случае набора специй, приведенных в рецепте, у вас всегда есть право пропустить некоторые темы, представленные на последующих страницах этой книги, но все они здесь для того, чтобы вы при желании могли попробовать их на вкус, и мы настойчиво рекомендуем вам сделать это.

Мы также должны указать на то, что, как и в случае любого другого учебника, имеющего отношение к технологиям, те сведения, которые вы почерпнете здесь сегодня, могут оказаться не совсем теми сведениями, которые потребуются вам завтра. Эта область знаний весьма динамична, и в этом состоит немалая часть ее притягательности. В нашей книге вы найдете полное отражение *текущей* картины излагаемого предмета вместе с подробной исторической перспективой. И это обеспечит вам надежное основание и подготовит к последующему самостоятельному росту по мере развития технологий. Мы советуем вам начать этот рост немедленно, прямо сейчас, изучая, осваивая и анализируя все, что осталось за рамками этого учебника. Учитесь учиться.

Спасибо за то доверие, которое вы нам оказали, выбрав для чтения эту книгу. Как ее авторы мы были обязаны предоставить вам рукопись, которая будет стоить потраченного на нее времени. И мы надеемся, что ваше решение будет положительным, — мы справились со своими обязательствами.

## Благодарности

Прежде всего я хочу поблагодарить Гленна Брукшира, который выносил и вынылывал эту книгу, его кровное детище, поскольку 11 предыдущих изданий охватывают период более четверти века стремительного роста и бурных изменений в области компьютерных наук. Поскольку это всего лишь второе издание, в подготовке которого мне было разрешено участвовать как соавтору и контролировать все вносимые изменения, страницы этого, 13-го, издания по-прежнему в большей части являются голосом самого Гленна и, как я надеюсь, следуют его видению предмета. Любые новые огрехи я принимаю на свой счет, а вся остальная элегантная базовая структура — его.

Я присоединяюсь к благодарности Гленна всем, кто поддерживал эту книгу, читая и используя в своей деятельности ее предыдущие издания. Для нас это большая честь. Тринадцатое издание учебника по компьютерным наукам? Мы, наверное, уже близки к тому, чтобы поставить своего рода рекорд.

Дэвид Т. Смит (David T. Smith) из Университета штата Пенсильвания в Индиане сыграл очень важную роль в моем соавторстве в отношении ревизии 11-издания этой книги, и это все еще заметно и в 13-м издании. Внимательное прочтение Дэвидом предыдущих изданий и тщательная проверка им дополнительных материалов явились очень важным вкладом в подготовительные работы. Эндрю Кюммель (Andrew Kuemmel), Джордж Корлисс (George Corliss) и Крис Мейфилд (Chris Mayfield) — каждый из них предоставил ценные замечания, углубленный анализ и/или поощрительный отзыв в отношении черновики этого или предыдущих изданий, тогда как Джеймс Е. Эймес (James E. Ames), Стефани Е. Оугуст (Stephanie E. August), Йонсук Чое (Yoonsuck Choe), Мелани Фейнберг (Melanie Feinberg), Эрик Д. Хэнли (Eric D. Hanley), Судхарсан Р. Йенгар (Sudharsan R. Iyengar), Рави Муккамала (Ravi Mukkamala) и Эдвард Приор (Edward Pryor) предложили нам свои ценные замечания в отношении записи программных конструкций в стиле Python в 12-м издании книги.

С каждым новым изданием пополняется список тех, кто внес свой вклад в создание этой книги. На сегодняшний день в этот список входят Дж.М. Адамс (J.M. Adams), К.М. Аллен (C.M. Allen), Д.К.С. Эллисон (D.C.S. Allison), И. Энджел (E. Angel), Р. Ашмор (R. Ashmore), Б. Ауэрнхеймер (B. Auernheimer), П. Бэнкстон (P. Bankston), М. Бернард (M. Barnard), П. Бендер (P. Bender), К. Боуер (K. Bowyer), П.У. Брешер (P. Brashear), К.М. Браун (C.M. Brown), Х.М. Браун (H.M. Brown), Б. Каллони (B. Calloni), Дж. Карпинелли (J. Carpinelli), М. Клэнси (M. Clancy), Р.Т. Клоуз (R.T. Close), Д.Х. Кули (D.H. Cooley), Л.Д. Корнелл (L.D. Cornell), М.Дж. Кроули (M.J. Crowley), Ф. Дик (F. Deek), М. Дикерсон (M. Dickerson), М.Дж. Дункан (M.J. Duncan), С. Езекиль (S. Ezekiel), К. Фокс (C. Fox), С. Фокс (S. Fox), Н.Е. Гиббс (N.E. Gibbs), Дж.Д. Гаррис (J.D. Harris), Д. Гэском (D. Hascom), Л. Хит (L. Heath), П.Б. Хендерсон (P.B. Henderson), Л. Хант (L. Hunt), М. Хатченрутер (M. Hutchenreuther), Л.А. Джен (L.A. Jehn), К.К. Колберг (K.K. Kolberg), К. Корб (K. Korb), Дж. Кренц (G. Krenz), Дж. Курозе (J. Kurose), Дж. Лью (J. Liu), Т. Дж. Лонг (T.J. Long), К. Мэй (C. May), Дж.Дж. МакКоннел (J.J. McConnell), У. МакКоун (W. McCown), С.Дж. Меррил (S.J. Merrill), К. Мессерсмит (K. Messersmith), Дж.К. Мойер (J.C. Moyer), М. Мэрфи (M. Murphy), Дж.П. Майерс (J.P. Myers), Дж.Д.С. Нунен (J.D.S. Noonan), Г. Натт (G. Nutt), У.У. Облити (W.W. Oblitey), С. Оларью (S. Olariu), Г. Риккарди (G. Riccardi), Дж. Райс (G. Rice), Н. Риккерт (N. Rickert), К. Ридесел (C. Riedesel), Дж.Б. Роджерс (J.B. Rodgers), Дж. Сайто (G. Saito), У. Савитч (W. Savitch), Р. Шлефтли (R. Schlafly), Дж.К. Шлимммер (J.C. Schlimmer), С. Селлс (S. Sells), З. Шен

(Z. Shen), Г. Шепперд (G. Sheppard), Дж.К. Симмс (J.C. Simms), М.К. Слэттери (M.C. Slattery), Дж. Слимик (J. Slimick), Дж.А. Сломка (Slomka), Дж. Солдеритч (J. Solderitsch), Р. Стейгервальд (R. Steigerwald), Л. Стайнберг (L. Steinberg), К.А. Штрубль (C.A. Struble), К.Л. Штрубль (C.L. Struble), У.Дж. Тэффе (W.J. Taffe), Дж. Толбарт (J. Talburt), П. Тонеллато (P. Tonellato), П. Тромович (P. Tromovitch), П.Г. Уинстон (P.H. Winston), И.Д. Уинтер (E.D. Winter), Э. Райт (E. Wright), М. Зиглер (M. Ziegler) и один неизвестный. Всем этим людям мы выразим свою самую искреннюю признательность.

Я также хочу поблагодарить своих друзей из издательства Pearson, оказавших поддержку этому проекту. Трейси Джонсон (Tracy Johnson), Лора Фриден-таль (Lora Friedenthal), Эрин Олт (Erin Ault), Кароль Снидер (Carole Snyder) и Скотт Дизанно (Scott Disanno) в особенности приложили свои знания и усилия, внеся множество улучшений в эту книгу в процессе ее подготовки.

И наконец, я благодарю мою жену, Петру, которая целиком брала на себя заботу о наших троих детях на протяжении многих долгих дней и вечеров, пока я работал над этим изданием. Она — моя главная опора.

*D.W.B.*

*Университет Маркетт*

*1 января 2018 года*

---

## **Ждем ваших отзывов!**

---

Вы, читатель этой книги, и есть главный ее критик. 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 интересны любые ваши замечания в наш адрес.

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

Отправляя письмо или оставляя сообщение, не забудьте указать название книги и ее авторов, а также свой электронный адрес. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию новых книг.

Наши электронные адреса:

E-mail: [info@dialektika.com](mailto:info@dialektika.com)

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



## Глава

# 0

**В** этой, вводной, главе мы познакомимся с областью компьютерных наук, рассмотрим ее в исторической перспективе и зложим необходимые основы, которые позволят приступить непосредственно к изучению материала данного курса.

---

# Введение

---

0.1 ЗНАКОМСТВО С АЛГОРИТМАМИ

0.2 ПРОИСХОЖДЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ  
МАШИН

0.3 ОБЗОР ЭТОГО КУРСА

0.4 ДОПОЛНИТЕЛЬНЫЕ ТЕМЫ  
КОМПЬЮТЕРНЫХ НАУК

Алгоритмы

Роль абстракции

Способность к творчеству

Данные

Программирование

Интернет

Влияние на общество

Компьютерные науки — это дисциплина, назначение которой состоит в создании научной основы для таких предметов, как проектирование электронно-вычислительных машин, разработка программного обеспечения, обработка информации, алгоритмическое решение задач, а также алгоритмический процесс сам по себе. Следовательно, она представляет собой фундамент для успешного применения современных компьютерных приложений, а также закладывает основы для разработки завтрашней компьютерной инфраструктуры.

Эта книга создана в целях предоставления всеобъемлющего введения в эти науки. В ней рассматривается широкий диапазон тем, включая большинство тех, которые составляют типичную университетскую учебную программу для специалистов по компьютерным наукам. Авторы стремились не только охватить весь спектр, но и показать динамику развития этой области знаний. Вот почему, помимо собственно изложения материала каждой темы, здесь вы найдете исторические сведения о ее развитии, информацию о текущем состоянии проводимых исследований и оценку перспектив ее развития в будущем. Нашей целью было обеспечить конструктивное, деятельное понимание компьютерных наук — такое, которое составит прочную основу для тех, кто желает продолжить свою профессиональную специализацию в этих науках, а также такое, которое позволит тем, кто выбрал иную специализацию, успешно достигать своих целей в современном обществе, которое становится все более и более технократическим.

## 0.1. Знакомство с алгоритмами

Мы начнем с самого фундаментального понятия в области компьютерных наук — **алгоритма**. Говоря неформально, алгоритм — это последовательность действий, которая определяет способ решения некоторой задачи.<sup>1</sup> Например, существуют алгоритмы для приготовления блюд (в этом случае их называют *рецептами*), поиска пути к нужному месту в неизвестном городе (в этом случае их называют *маршрутами*), для управления стиральными машинами (обычно помещаемые на внутреннюю сторону крышки машины), для воспроизведения музыки (изображаемые в виде *музыкальных нот*), а также для выполнения различных фокусов. Пример алгоритма последнего типа приведен на рис. 0.1.

Прежде чем машина, такая как компьютер, сможет выполнить некоторое задание, необходимо определить алгоритм выполнения этого задания и предоставить его машине в том виде, который будет с ней совместим. Представление алгоритма в таком виде называется **программой**. Для удобства человека текст компьютерных программ обычно печатают на бумаге или отображают на

---

<sup>1</sup> Если быть более точными, алгоритм — это упорядоченное множество однозначных, выполнимых манипуляций, определяющих некоторое конечное действие. Подробнее эта тема обсуждается в главе 5.



дисплее компьютера. Для удобства машины программы должны быть закодированы в соответствии с технологическими принципами построения машины. Процесс разработки программы, включая ее кодирование в совместимую с выбранной машиной форму, а также процедуру ввода программы в эту машину, принято называть **программированием** или, как еще говорят, **кодированием**. Программы и алгоритмы, которые они представляют, в совокупности принято называть **программным обеспечением**, в противоположность самой машине, которую принято называть **аппаратным обеспечением**.

**Описание фокуса.** Исполнитель фокуса выкладывает на стол несколько карт из обычной игровой колоды, помещая их лицевой стороной вниз. При этом колода карт постоянно тщательно перетасовывается. Затем зрителям предоставляется право выбора, какого цвета должна быть масть очередной открываемой карты — черного или красного, после чего исполнитель переворачивает на столе требуемую карту.

**Секрет фокуса и последовательность его выполнения.**

**Этап 1.** Из обычной колоды карт выберите десять карт красной масти и десять черной. Уложите их в две стопки, согласно цвету масти, лицевой стороной вверх.

**Этап 2.** Объявите, что вы выбрали несколько карт черной масти и несколько красной.

**Этап 3.** Выберите карты красной масти. Под видом упорядочения в одну маленькую стопку возьмите карты в левую руку лицевой стороной вниз, после чего большим и указательным пальцами правой руки отогните один из концов этой стопки вниз так, чтобы каждая карта стала слегка выпуклой. Затем со словами: “В этой стопке карты красной масти” положите колоду карт на стол лицевой стороной вниз.

**Этап 4.** Возьмите карты черной масти. Таким же образом, как и в предыдущем случае, отогните уголок стопки вверх, что сделает карты слегка вогнутыми. Затем положите карты на стол лицевой стороной вниз со словами: “А в этой стопке карты черной масти”.

**Этап 5.** Сразу после того, как карты черной масти будут выложены на стол, перетасуйте обе стопки и приступайте раскладывать их на столе (опять же, лицевой стороной вниз). Раскладывая карты, демонстрируйте зрителям, как тщательно вы их перетасовываете.

**Этап 6.** Когда все карты будут разложены, выполните следующие действия.

**6.1.** Попросите зрителей выбрать карту красной или черной масти.

**6.2.** Если запрашиваемый цвет масти красный и есть карта с выпуклой поверхностью, переверните ее со словами: “Вот карта красной масти”.

**6.3.** Если запрашиваемый цвет черный и есть карта с вогнутой поверхностью, переверните ее со словами: “Вот карта черной масти”.

**6.4.** В противном случае заявите, что больше нет карт требуемого цвета, и переверните все оставшиеся карты, чтобы доказать это.

**Рис. 0.1.** Алгоритм для выполнения карточного фокуса

Изучение алгоритмов первоначально составляло один из разделов математики. Действительно, поиск алгоритмов привлекал пристальное внимание математиков задолго до того, как появились современные вычислительные машины. Основной целью этих поисков было определение общего набора указаний, описывающих способ решения задачи некоторого типа. Одним из наиболее известных результатов ранних поисков является алгоритм деления столбиком для определения частного двух многозначных чисел. В качестве еще одного примера можно привести алгоритм Евклида, предложенный этим древнегреческим математиком для определения общего наибольшего делителя двух положительных целых чисел. Описание этого алгоритма представлено на рис. 0.2.

**Описание.** В этом алгоритме предполагается, что входные данные представляют собой два целых положительных числа, для которых требуется определить наибольший общий делитель.

**Порядок выполнения.**

**Этап 1.** Присвойте переменным  $M$  и  $N$  значения двух введенных чисел (большого и меньшего соответственно).

**Этап 2.** Разделите  $M$  на  $N$  и присвойте значение остатка переменной  $R$ .

**Этап 3.** Если значение  $R$  не равняется 0, присвойте переменной  $M$  значение переменной  $N$ , затем переменной  $N$  присвойте значение остатка  $R$  и вернитесь к этапу 2. В противном случае наибольшим общим делителем заданной пары чисел является значение, присвоенное в данный момент переменной  $N$ .

**Рис. 0.2.** Алгоритм Евклида для поиска наибольшего общего делителя двух положительных целых чисел

Как только алгоритм решения задачи будет найден, само выполнение предусмотренных этим алгоритмом действий уже не потребует понимания законов, по которым данный алгоритм был построен. Напротив, решение задачи сводится до простого выполнения установленной последовательности инструкций. (Мы можем применять алгоритм деления столбиком для поиска частного двух многозначных чисел или Евклидов алгоритм определения наибольшего общего делителя, даже не понимая принципов, на основании которых эти алгоритмы работают.) По сути, в алгоритме закодированы все сведения, необходимые для решения поставленной задачи.

Фиксация и передача интеллектуальной информации (или по крайней мере правил интеллектуального поведения) с помощью алгоритмов позволяет создавать машины, способные выполнять полезные задания. Следовательно, уровень интеллекта, проявляемый машинами, ограничен той информацией, которая может быть передана им через алгоритмы. Только после того, как будет найден алгоритм, позволяющий решить поставленную задачу, может быть сконструировано некоторое устройство, предназначенное для ее решения. В свою

очередь, если не существует алгоритма выполнения определенного задания, то его выполнение оказывается за пределами возможностей машин.

Выявление ограничений алгоритмических возможностей сформировалось в математике как самостоятельная тема в 1930-х годах после публикации Куртом Гёделем теоремы о неполноте. Эта теорема по существу утверждает, что в любой математической теории, охватывающей нашу традиционную арифметическую систему, существуют утверждения, истинность или ложность которых алгоритмически невозможно ни доказать, ни опровергнуть. Осознание этого факта потрясло самые основы математики, и исследования в области изучения алгоритмических возможностей, которые последовали вслед за этим, собственно и положили начало новой области знаний, известной теперь как компьютерные науки. Как видите, именно изучение алгоритмов сформировало ядро компьютерных наук.

## 0.2. Происхождение вычислительных машин

Современные компьютеры имеют весьма обширную генеалогию. Одним из первых вычислительных устройств является абак, т.е. *счеты*. Историки говорят, что, вероятно, впервые подобные устройства появились в Древнем Китае, но достоверно известно, что счеты уже использовались в ранний период древнегреческой и древнеримской цивилизаций. Само это устройство довольно простое и состоит из бусин, нанизанных на прутья, которые вставлены в прямоугольную рамку (рис. 0.3). Перемещение бусин взад и вперед по прутьям позволяет представлять сохраняемые значения. Именно расположение бусин этот “компьютер” использует для представления и суммирования данных. Управление выполнением требуемого алгоритма с помощью этой машины возлагается на человека-оператора. Таким образом, сами счеты являются просто системой хранения данных, и только сочетание человека и счет образует полную вычислительную машину.

В период с начала эпохи Возрождения и до конца Нового времени неоднократно предпринимались попытки построения более сложных вычислительных машин. В этот период технология создания вычислительных машин строилась на использовании зубчатых колес. Среди создателей подобных механизмов были француз Блез Паскаль (1623–1662), немец Готфрид Вильгельм Лейбниц (1646–1716) и англичанин Чарльз Бэббидж (1792–1871). В их устройствах данные представлялись с помощью определенного расположения зубчатых колес, а исходные значения вводились механически, вручную, посредством приведения колес в необходимое положение. Результаты вычислений в машинах Паскаля и Лейбница определялись путем считывания конечного положения колес, примерно так, как мы сегодня определяем суммарный пробег автомобиля по

показаниям спидометра. Однако Бэббидж предвидел создание машин, которые будут печатать результаты вычислений на бумаге, что позволит устранить возможность ошибок при считывании.

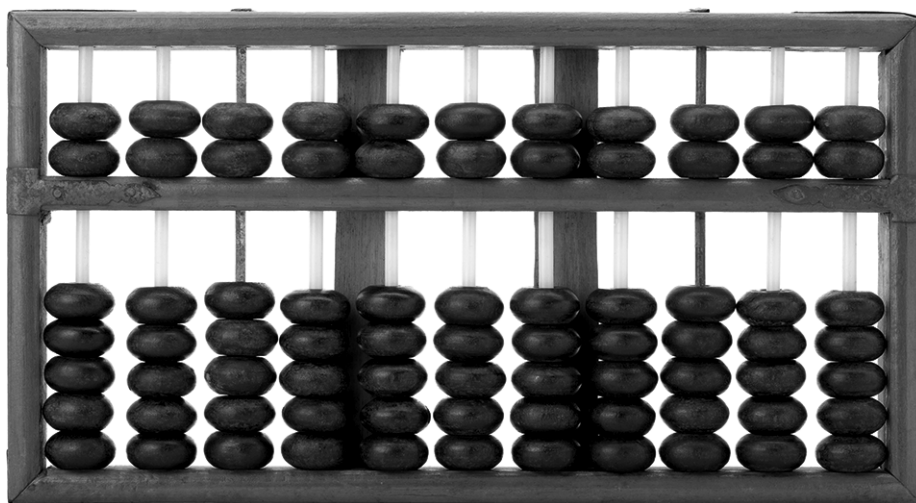


Рис. 0.3. Китайский вариант деревянных счет

Что касается способности следовать алгоритму, то в этих машинах уже явно виден определенный прогресс. Машина Паскаля могла выполнять только алгоритм суммирования, поэтому средства для выполнения соответствующей последовательности действий были встроены в саму машину. Аналогичным образом в архитектуру машины Лейбница был встроен набор неизменных алгоритмов, позволяющих выполнять множество арифметических действий по выбору оператора. Машина Бэббиджа, в отличие от двух предыдущих машин, была сконструирована таким образом, что последовательность выполняемых действий могла быть передана с помощью отверстий в бумажных картах. Таким образом, машина Бэббиджа была уже *программируемой*. Именно по этой причине ассистентка Бэббиджа, Августа Ада Байрон, опубликовавшая статью о том, как именно можно было программировать работу аналитической машины Бэббиджа для выполнения различных вычислений, теперь считается первым в мире программистом.

Идея передачи алгоритма с помощью отверстий в бумажных картах не является собственным открытием Бэббиджа. В 1801 году француз Джозеф Жаккард (1752–1834) применил подобную технологию для управления ткацкими станками. В частности, он разработал ткацкий станок, процесс плетения которого определялся узором из отверстий, нанесенных на последовательность больших карт из толстой бумаги. Благодаря этому алгоритм, по которому работала машина, можно было легко изменить, что позволяло на одном и том же

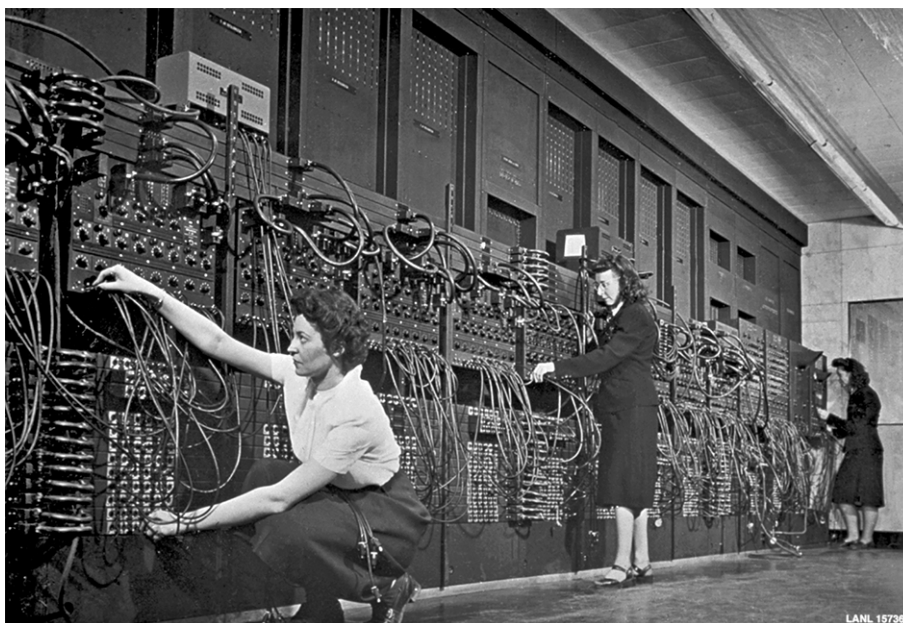
станке производить множество различных типов тканей. Позднее Герман Холлерит (1860–1929) использовал идею представления информации с помощью отверстий в бумажных картах для ускорения составления таблиц статистических сводок при переписи населения США в 1890 году. Фактически именно эта разработка Холлерита привела к созданию корпорации IBM. Такие карты в конечном итоге стали известны как перфокарты и сохранились как популярный способ общения с компьютерами вплоть до 1990-х годов.

Технологии тех времен не обеспечивали необходимого уровня точности, который позволил бы сделать сложные шестеренчатые калькуляторы Паскаля, Лейбница и Бэббиджа достаточно эффективными и популярными. И до тех пор, пока в начале 1900-х годов электроника не расширила возможности механических устройств, технология не позволяла поддерживать те теоретические разработки, которые появлялись в зарождающейся компьютерной науке того времени. Примерами такого прогресса могут служить электромеханическая машина Джорджа Стибица, созданная в 1940 году в лабораториях компании Bell, и машина Mark I, созданная в 1944 году в Гарвардском университете Говардом Айкеном совместно с группой инженеров корпорации IBM. В этих машинах широко использовались механические реле, работой которых управляла электроника. В этом смысле они устарели практически сразу же после создания, так как другие исследователи в это же время уже использовали технологию электровакуумных приборов для конструирования полностью электронных цифровых вычислительных машин. Первым таким устройством была машина Атанасова-Берри, создававшаяся с 1937 по 1941 год в колледже шт. Айова (сегодня — Университет шт. Айова). Создателями машины были Джон Атанасов и его ассистент Клиффорд Берри. Другим аналогичным устройством является машина COLLOSSUS, созданная в Англии под руководством Томми Флаурса в конце второй мировой войны для расшифровки перехватываемых немецких шифрованных сообщений. (В действительности было создано, вероятно, не менее 10 таких машин, однако режим военной секретности и вопросы национальной безопасности требовали сохранения их существования в тайне, из-за чего эти машины так и не стали общеизвестной частью “генеалогического древа” компьютеров.) За ними вскоре последовали другие, более универсальные машины, например ENIAC (электронный цифровой интегратор и калькулятор), разработанный Джоном Мочли и Дж. Преспером Экертом в электротехнической школе Мура, Университет шт. Пенсильвания (рис. 0.4).

С этого момента дальнейшая история вычислительных машин в большей степени определялась прогрессом в области передовых технологий, включая изобретение транзисторов (за что физики Вильям Шокли, Джон Бардин и Уолтер Братайн получили Нобелевскую премию) и последующее развитие технологии интегральных схем (за что Нобелевскую премию в области физики



получил уже Джек Кибли). Благодаря этим разработкам громоздкие, размером с комнату, машины 1940-х годов за пару десятилетий уменьшились до размеров обычного шкафа. В то же время вычислительная мощность этих компьютеров удваивалась примерно каждые два года (эта тенденция сохраняется и до наших дней). По мере развития технологии создания интегральных схем многие компоненты компьютеров были выпущены на рынок в виде отдельных интегрированных элементов, заключенных в небольшие пластиковые корпуса, получившие название “чипы”.



**Рис. 0.4.** Три женщины контролировали работу машины ENIAC, пользуясь ее главной панелью управления, пока эта машина находилась в школе Мура. Позднее этот компьютер был перевезен в лабораторию баллистических исследований армии США

Огромным шагом вперед в популяризации вычислительных машин было создание настольных компьютеров. Начало созданию таких малогабаритных машин положили те люди, для которых компьютеры были предметом увлечения. Именно они начали экспериментировать с машинами, сконструированными в домашних условиях в виде комбинации чипов. Благодаря этой “подпольной” любительской деятельности Стив Джобс (Steve Jobs) и Стефен Возняк (Stephen Wozniak) впервые построили коммерчески жизнеспособный домашний компьютер и в 1976 году основали компанию Apple Computer, Inc., специализировавшуюся на изготовлении и продаже подобных изделий. Другими

### Разностная машина Бэббиджа

Машины, сконструированные Чарльзом Бэббиджем, по своей структуре были настоящими предшественниками современных компьютеров. Если бы технологии тех времен позволяли выпускать эти машины экономически выгодным образом, а требования к обработке данных со стороны коммерческих и правительственных кругов были сопоставимы с современными, идеи Бэббиджа могли бы привести к компьютерной революции еще в 1800-х годах. В действительности во время его жизни была создана только демонстрационная модель его разностной машины. Эта машина определяла числовые значения посредством вычисления “конечных разностей”. Получить представление об этом методе можно, например, рассмотрев проблему вычисления квадратов целых чисел. Начнем с известного нам факта, что квадрат нуля равен нулю, квадрат 1 равен 1, квадрат 2 равен 4, а квадрат 3 равен 9. Приняв это во внимание, можно вычислить квадрат 4 следующим образом (см. рисунок ниже). Сначала находим разность уже известных нам квадратов:  $1^2 - 0^2 = 1$ ,  $2^2 - 1^2 = 3$  и  $3^2 - 2^2 = 5$ . Затем вычисляем разности этих результатов:  $3 - 1 = 2$  и  $5 - 3 = 2$ . Обратите внимание, что обе эти разности равны 2. Предположив, что это постоянство значения разностей сохраняется и далее (математики могут доказать, что это именно так), мы можем заключить, что разность между значениями  $(4^2 - 3^2)$  и  $(3^2 - 2^2)$  также должна быть равна 2. Тогда разность  $(4^2 - 3^2)$  должна быть на 2 больше, чем разность  $(3^2 - 2^2)$ , а значит,  $4^2 - 3^2 = 7$ , а отсюда  $4^2 = 3^2 + 7 = 16$ . Теперь, когда мы знаем значение квадрата 4, мы можем повторить эту процедуру для вычисления квадрата 5, воспользовавшись уже известными значениями  $1^2$ ,  $2^2$ ,  $3^2$  и  $4^2$ . (Хотя более подробное обсуждение метода конечных разностей выходит за рамки этого курса, студенты, знающие дифференциальное исчисление, могут заметить, что предыдущий пример построен на том факте, что производной функции  $y = x^2$  является прямая линия с наклоном 2.)

| $x$ | $x^2$ | Первая разность | Вторая разность |
|-----|-------|-----------------|-----------------|
| 0   | 0     |                 |                 |
| 1   | 1     | 1               |                 |
| 2   | 4     | 3               | 2               |
| 3   | 9     | 5               | 2               |
| 4   | 16    | 7               | 2               |
| 5   |       |                 |                 |

компаниями, которые вслед за Apple начали выпускать подобную продукцию, были Commodore, Heathkit и Radio Shack. Несмотря на то что продукция этих компаний была популярна среди многочисленных любителей, она не получила широкого признания в деловых кругах, которые продолжали рассматривать респектабельную корпорацию IBM как главный источник удовлетворения своих потребностей в вычислительной технике.

### Августа Ада Байрон

Августа Ада Байрон, графиня Лавлейс, многие годы является объектом полемики и восторженных комментариев в компьютерных кругах. Она прожила относительно недолгую, 37 лет (1815–1852), и в чем-то трагическую жизнь, осложненную слабым здоровьем и тем фактом, что была инакомыслящей в обществе, которое жестко ограничивало профессиональные возможности женщин. Хотя Ада проявляла интерес к широкому спектру наук, главные силы она направила на изучение математики. Интерес к “вычислительной науке” появился у нее после того, как она была очарована машиной Чарльза Бэббиджа на демонстрации прототипа его разностной машины в 1833 году. Ее вклад в компьютерные науки связан с выполненным ею переводом с французского на английский статьи, в которой описывался проект аналитической машины Бэббиджа. Бэббидж предложил ей добавить к этому переводу свои комментарии, в которых описывались бы возможные применения машины и приводились бы примеры того, как эту машину можно запрограммировать для выполнения различных заданий. Энтузиазм Бэббиджа по отношению к работе Ады Байрон был, по всей видимости, вызван его надеждами получить в результате этой публикации финансовую поддержку, необходимую для построения его аналитической машины. (Как дочь лорда Байрона Ада Байрон имела статус знаменитости с потенциально значимыми финансовыми связями.) Эти надежды так никогда и не оправдались, однако комментарий Ады Байрон сохранился и со временем был признан как документ, содержащий первый пример настоящей компьютерной программы. Степень влияния Бэббиджа на работу Ады Байрон все еще дискутируется историками. Одни утверждают, что именно Бэббидж сделал главный вклад, тогда как другие — что он был скорее препятствием, чем помощью в работе Ады. В любом случае сегодня Аду Байрон считают первым в мире программистом, и этот статус был подтвержден Министерством обороны США, когда оно в ее честь дало название популярному языку программирования — Ada.

В 1981 году корпорация IBM представила свой первый настольный персональный компьютер, который так и назывался — “Personal Computer” (персональный компьютер), для краткости — “PC” (ПК). Базовое программное обеспечение для этого компьютера было разработано молодой энергичной компанией, ныне известной как Microsoft. Эта модель персонального компьютера очень быстро получила признание и возвела настольный компьютер в ранг общепризнанного



предмета потребления для деловых кругов общества. Сегодня термин “ПК” широко используется для обозначения всех тех устройств (изготовленных различными производителями), базовые модели которых ведут свое начало от первоначального настольного компьютера корпорации IBM. Большинство таких машин по-прежнему выпускается на рынок с программным обеспечением от фирмы Microsoft. Со временем термин “ПК” и исходный термин “настольный компьютер” (desktop) стали практически взаимозаменяемыми.

К концу XX столетия возможность соединить отдельные компьютеры в единую систему мирового масштаба, получившую название **Интернет** (Internet), привела к настоящей революции в средствах коммуникации. В этом контексте Тим Бернерс-Ли (Tim Berners-Lee), британский ученый, предложил развернуть систему, посредством которой документы, хранящиеся на компьютерах, подключенных к Интернету, можно было связать между собой, создав обширную сеть связанной информации, получившей название **веб** (Web) — сокращение от полного названия “World Wide Web” (Всемирная паутина). Чтобы сделать информацию в вебе доступной тем, кто в ней нуждается, были разработаны программные системы, получившие название **поисковые машины** (search engines). Их назначение — “просеивать” содержимое веба, “классифицировать” найденное и использовать полученные результаты для оказания помощи тем, кто ищет информацию по определенным темам. Главными игроками на этом поприще сейчас являются Google, Яндекс, Yahoo и Microsoft. Эти компании продолжают расширять свою активность в вебе, часто в таких направлениях, которые бросают вызов нашему традиционному образу мышления.

Пока настольные, а затем и переносные (лэптопы и ноутбуки) персональные компьютеры получали все большее распространение и популярность в частных домах, миниатюризация вычислительных машин продолжалась во все возрастающей степени. Сегодня крошечные компьютеры непосредственно встраиваются в широчайший спектр электронных устройств и приборов. Обыкновенный автомобиль сейчас может содержать десятки маленьких компьютеров, обеспечивающих работу службы GPS (Global Positioning Systems — глобальная система позиционирования), контролирующих работу двигателя и даже обеспечивающих обработку голосовых команд для управления аудиосистемой автомобиля и его системой телефонной связи.

Вероятно, наиболее революционным применением компьютерной миниатюризации явилась разработка и последующее расширение возможностей **смартфонов**, наручных универсальных компьютеров, в которых собственно телефония является лишь одним из множества доступных приложений. Более мощные, чем суперкомпьютеры предыдущих десятилетий, эти устройства карманных размеров оборудованы множеством датчиков и интерфейсов, включая

## Google

Основанная в 1998 году, корпорация Google LLC (ранее — Google, Inc.) в настоящее время стала одной из наиболее известных и влиятельных технологических компаний. Ее основной сервис, поисковая машина Google, ежедневно используется многими миллионами людей для поиска документов в World Wide Web. Кроме того, корпорация Google предоставляет всем желающим бесплатную службу электронной почты (под названием “Gmail”), развернутую в Интернете чрезвычайно популярную в мире службу хранения и просмотра видеоклипов (под названием “YouTube”), а также обеспечивает работу многих других популярных интернет-сервисов, таких как Google Maps, Google Calendar, Google Earth, Google Books или Google Translate.

Однако, помимо того что ее можно считать истинным образцом духа предпринимательства, корпорация Google является еще и ярким примером того, как стремительное развитие и повсеместное распространение технологий может оказывать влияние на жизнь и состояние общества. Например, всемирная экспансия поисковой машины Google привела к появлению вопросов о том, в какой степени международная компания должна соблюдать пожелания отдельных правительств; в связи с ростом популярности сервиса YouTube были подняты вопросы в отношении того, в какой степени компания должна нести ответственность за информацию, которую сторонние пользователи распространяют через ее сервисы, а также о той степени, в которой компания может заявить о своем праве собственности на эту информацию; сервис Google Books вызвал озабоченность в отношении объема и ограничений прав на интеллектуальную собственность, а сервис Google Maps обвиняют в нарушении прав на неприкосновенность частной жизни.

многокомпонентные фото- и видеокамеры, микрофоны, компасы, сенсорные экраны, датчики GPS, акселерометры (для определения ориентации и перемещения устройства), а также поддерживающие разнообразные беспроводные технологии, обеспечивающие возможность коммуникации с другими смартфонами и компьютерами. Многие считают, что появление смартфонов оказало на глобальное сообщество даже большее влияние, чем революция, вызванная появлением персональных компьютеров.

## 0.3. Обзор этого курса

В этой книге материал упорядочен в соответствии с подходом к изучению компьютерных наук, предусматривающим переход от конкретного к абстрактному. Она начинается с таких практических тем, как устройство и оборудование компьютеров, а завершается рассмотрением таких абстрактных аспектов, как сложность алгоритмов и вычислимость функций. В результате изучение

предмета следует схеме построения все более и более абстрактных инструментов, поскольку наше понимание предмета расширяется.

В начале рассматриваются темы, относящиеся к проектированию и конструированию машин для выполнения алгоритмов. В главе 1, “Хранение данных”, речь идет о том, как информация кодируется и сохраняется в современных компьютерах, а в главе 2, “Обработка данных”, рассмотрены основные внутренние операции в простом компьютере. Хотя определенная часть материала этой книги связана с технологиями, общая ее тема не зависит от конкретных технологий. То есть такие темы, как структура цифровых электронных схем, кодирование данных, способы их сжатия, а также архитектура компьютеров, фактически относятся к широкому спектру технологий и, скорее всего, будут оставаться актуальными независимо от направления развития технологий в будущем.

В главе 3, “Операционные системы”, рассказывается о программном обеспечении, управляющем работой компьютера в целом. Такое программное обеспечение принято называть *операционной системой*. Именно операционная система компьютера управляет работой интерфейса между этой машиной и внешним миром, защищая машину и хранящиеся в ней данные от несанкционированного доступа, позволяя пользователю компьютера запускать на выполнение различные программы и координируя внутреннее взаимодействие элементов компьютера, необходимое для выполнения запросов пользователя.

В главе 4, “Компьютерные сети и Интернет”, объясняется, как компьютеры могут быть соединены между собой для образования компьютерных сетей и как сети подключаются одна к другой с целью образования их объединений — *интернетов*. В этой главе также обсуждаются такие темы, как сетевые протоколы, структура и принципы управления Всемирной сети — Интернета и развернутой на ее основе распределенной системы World Wide Web, а также даются необходимые сведения по вопросу защиты.

В главе 5, “Алгоритмы”, изучение алгоритмов проводится с более формальной точки зрения. В ней поясняется как алгоритмы создаются, приводятся несколько фундаментальных алгоритмических структур и обсуждаются простейшие способы представления алгоритмов. В заключение вводятся такие важные понятия, как эффективность и правильность алгоритмов.

В главе 6, “Языки программирования”, рассматриваются представление алгоритмов и разработка программ. Вы узнаете, что поиск лучших методов программирования в конечном счете привел к выработке различных методологий программирования или *парадигм*, для каждой из которых был разработан собственный набор языков программирования. Помимо парадигм и отдельных языков, в главе обсуждаются вопросы построения грамматики различных языков и их трансляции.

В главе 7, “Технология разработки программного обеспечения”, обсуждается та ветвь компьютерных наук, которая связана с проблемами, возникающими при создании больших программных систем. Причина их возникновения состоит в том, что проектирование и создание крупных программных систем является сложной, комплексной задачей, порождающей проблемы, выходящие за рамки традиционной инженерии. В результате задача разработки новых технологий создания программного обеспечения становится важным полем исследований в рамках компьютерных наук, затрагивающим такие аспекты, как методы проектирования, управление проектами, управление персоналом, разработка языка программирования и даже архитектура систем.

В следующих двух главах рассматриваются способы организации хранения данных и доступа к ним в компьютерных системах. В главе 8, “Структуры данных”, речь идет о методах, традиционно используемых для организации данных в основной памяти компьютера, и прослеживается эволюция абстракции данных от концепций примитивов до современных объектно-ориентированных методологий. В главе 9, “Системы баз данных”, рассматриваются методы, традиционно используемые для организации данных в массовой памяти компьютеров и объясняется, как могут быть реализованы чрезвычайно большие и сложные системы современных баз данных.

В главе 10, “Компьютерная графика”, речь идет о компьютерной графике и анимации — области компьютерных наук, которая имеет дело с созданием и отображением виртуальных миров. Основываясь на достижениях в более традиционных областях компьютерных наук, таких как архитектура машин, разработка алгоритмов, структуры данных и технология разработки программного обеспечения, результаты исследований в области компьютерной графики и анимации продемонстрировали колоссальный прогресс, превратив это направление в динамичную, захватывающую и процветающую область применения компьютеров. Более того, здесь можно найти яркие иллюстрации тому, как различные дисциплины компьютерных наук успешно сочетаются с дисциплинами других областей науки и культуры, такими как физика, искусство и фотография, давая поразительные результаты.

В главе 11, “Искусственный интеллект”, рассказывается, что с целью разработки более полезных человеку машин исследователи в области компьютерных наук обратились к изучению человеческого интеллекта, стремясь углубить свое понимание этого феномена. Главная цель — понять, как в нашем сознании осуществляются рассуждение и восприятие, что позволит исследователям разработать алгоритмы, копирующие эти процессы, а значит, передать сравнимые возможности машинам. Результатом явилось появление еще одной области компьютерных наук, получившей название *искусственный интеллект*, в значительной степени опирающейся на исследования в таких областях знаний, как психология, биология и лингвистика.

Завершается эта книга главой 12, “Теория вычислений”, в которой исследуются теоретические основы компьютерных наук — предмет, который позволяет понять ограничения, существующие в отношении алгоритмов (а значит, и алгоритмических машин). Здесь приводятся примеры некоторых проблем, которые невозможно решить алгоритмически (а значит, решение которых лежит и за пределами возможностей машин), наравне с демонстрацией того, что решение многих других проблем может потребовать таких невероятных затрат времени или пространства, что их также можно считать неразрешимыми с практической точки зрения. Таким образом, именно благодаря подобным исследованиям мы можем понять область применения и существующие ограничения возможностей алгоритмических систем.

В каждой главе нашей целью было исследовать тему с глубиной, достаточной для достижения истинного ее понимания. Мы стремились предоставить практические, действующие знания в области компьютерных наук — такие, которые позволят читателю лучше понимать то технократическое общество, в котором он живет, и создадут прочную основу, на которую он сможет уверенно опереться в своем дальнейшем продвижении в науке и технике.

## 0.4. Дополнительные темы компьютерных наук

В каждой главе, помимо основных тем, перечисленных выше, мы также надеемся расширить ваше понимание компьютерных наук посредством включения нескольких всеобъемлющих тем. Миниатюризация компьютеров и постоянное расширение их возможностей вывело компьютерные технологии на передний край в современном обществе, и эти технологии настолько превалируют, что близкое знакомство с ними является совершенно необходимым условием для того, чтобы ощущать себя частью современного мира. Компьютерные технологии изменили возможности правительств осуществлять контроль; оказали колоссальное влияние на мировую экономику; привели к поразительным достижениям в области научных исследований; внесли революционные изменения в сбор, хранение и использование данных; предоставили людям новые средства общения и взаимодействия и неоднократно бросили вызов текущему состоянию общества, вынуждая его изменяться. Результатом стало бурное развитие областей знаний, прилежащих к компьютерным наукам, каждая из которых сейчас сама по себе является важной областью изучения и исследований. Более того, как и в случае машиностроения и физики, часто трудно провести разграничительную линию между этими научными областями и компьютерными науками. Таким образом, чтобы дать правильные перспективы, в этой книге следовало не только рассматривать основные темы ядра компьютерных наук, но и проводить экскурсии в различные дисциплины, связанные как с

практическим приложением, так и с последствиями развития науки. Действительно, любое введение в компьютерные науки должно пониматься как задача междисциплинарная.

Приступая к изучению области компьютерных наук во всей ее широте, полезно всегда держать в уме те главные темы, которые объединяют компьютерные науки в единое целое. Их всего семь — “Большая семерка” основных концепций включает: алгоритмы, абстракция, творчество, данные, программирование, Интернет и воздействие. В следующие главы мы включили ряд тем, каждый раз представляя их центральные положения, текущую область исследований и некоторые из методов, используемых для расширения знаний в этой области. Помните о Большой семерке, потому что мы будем возвращаться к ней снова и снова.

---

## Алгоритмы

---

Ограниченные возможности хранения данных и использование детального, требующего больших затрат времени, программирования ограничивали сложность алгоритмов, которые могли выполнять первые вычислительные машины. Однако по мере того как эти ограничения преодолевались, компьютеры стали применяться к решению все более и более сложных задач. Когда попытки выразить структуру этих задач в алгоритмической форме стали требовать чрезмерных умственных усилий, все больше и больше исследований было направлено на изучение самих алгоритмов и процесса программирования.

Именно тогда теоретическая работа математиков начала приносить свои плоды. В результате появления теоремы Геделя о неполноте к моменту создания первых вычислительных машин математики уже в достаточной степени исследовали те аспекты алгоритмических процессов, которые были необходимы развивающейся технологии. Этим была заложена основа для появления новой дисциплины, ныне известной как *компьютерные науки*.

Сегодня эта дисциплина зарекомендовала себя как наука об алгоритмах. Как мы уже убедились, ее границы достаточно широки, поскольку она использует знания из таких дисциплин, как математика, инженерное искусство, психология, биология, менеджмент и языкознание. На самом деле исследователи в различных ответвлениях компьютерных наук могут иметь очень отличающееся определение для этой области науки. Например, тот, кто работает в области архитектуры компьютеров, может сфокусироваться на задаче миниатюризации компьютерных схем, а значит, видит компьютерные науки как развитие и применение технологии. Однако специалист, работающий в области систем баз данных, может представлять компьютерные науки как поиск возможностей сделать информационные системы более полезными. Наконец, исследователь в

области искусственного интеллекта может понимать компьютерные науки как изучение интеллекта и интеллектуального поведения.

Но как бы там ни было, все эти специалисты в той или иной степени связаны с определенными аспектами науки об алгоритмах.

Принимая во внимание ту центральную роль, которую алгоритмы играют в компьютерных науках (рис. 0.5), будет целесообразно определить некоторые вопросы, которые помогут сосредоточиться в нашем изучении на этой важнейшей идее.

- Какие проблемы могут быть решены с помощью алгоритмических процессов?
- Как можно упростить задачу поиска требуемого алгоритма?
- Каким образом можно усовершенствовать технологию представления и передачи алгоритмов?
- Как можно анализировать и сравнивать свойства различных алгоритмов?
- Как алгоритмы можно использовать для манипулирования информацией?
- Как можно применить алгоритмы к демонстрации интеллектуального поведения?
- Какое воздействие алгоритмы оказывают на общество?



Рис. 0.5. Основная роль алгоритмов в компьютерных науках

---

## Роль абстракции

---

Термин **абстракция** в том понимании, которое мы ему здесь придаем, ссылается на различие между внешними свойствами объекта и деталями внутреннего устройства этого объекта. Именно абстракция позволяет игнорировать внутренние детали устройства сложных объектов, таких как компьютер, автомобиль



или микроволновая печь, и использовать их как целостную, доступную пониманию вещь. Более того, именно посредством абстракции такие сложные системы могут быть разработаны и изготовлены. Компьютеры, автомобили и микроволновые печи сконструированы из компонентов, каждый из которых представляет некоторый уровень абстракции, на котором использование этих компонентов рассматривается изолировано от их внутреннего устройства.

Именно благодаря использованию абстракции мы получаем возможность конструировать, анализировать и управлять большими, сложными компьютерными системами, что было бы совершенно невыполнимым, если бы они рассматривались на их внутреннем, детальном уровне. На каждом уровне абстракции мы рассматриваем систему в терминах компонентов, называемых **абстрактными инструментами**, внутреннее устройство которых игнорируется. Это позволяет сконцентрироваться на том, как каждый компонент взаимодействует с другими компонентами на том же уровне и как вся их совокупность образует компонент более высокого уровня. Действуя таким образом, мы можем понять работу той части системы, которая имеет отношение к задаче, и не потеряться в море деталей.

Следует подчеркнуть, что использование абстракции не ограничивается лишь наукой и технологией. Абстракция является важным методом упрощения, с помощью которого наше общество создало тот образ жизни, который иначе создать было бы просто невозможно. Например, немногие из нас понимают, как на самом деле реализуются различные удобства, которыми мы пользуемся в повседневной жизни. Мы употребляем пищу и носим одежду, которую не способны производить самостоятельно. Мы используем электроприборы, не понимая принципов, положенных в основу их функционирования. Мы пользуемся услугами разных людей, не вникая в подробности их деятельности. С каждым новым достижением лишь небольшая часть общества стремится профессионально специализироваться в этой области, в то время как остальные лишь учатся пользоваться достигнутыми результатами, воспринимаемыми как абстрактные инструменты, внутреннее устройство которых нам понимать не требуется. В результате накопленный объем абстрактных средств расширяется, повышая способность общества к дальнейшему продвижению вперед.

В этой книге абстракция встречается вновь и вновь. Вы узнаете, что компьютерное оборудование конструируется на уровне абстрактных инструментов, а разработка больших систем программного обеспечения осуществляется по модульному принципу, в котором каждый модуль воспринимается как абстрактный инструмент построения более крупных модулей. Более того, абстракция играет важную роль в задаче развития самих компьютерных наук, позволяя исследователям сосредоточить свое внимание на определенных аспектах сложной задачи. В действительности даже сама организация этой книги отражает



данное свойство науки. Каждая глава, фокусируясь на отдельном аспекте компьютерных наук, часто оказывается на удивление независимой от остальных, тогда как все вместе эти главы образуют исчерпывающий обзор всей этой области знаний.

---

## Способность к творчеству

---

Хотя компьютеры могут быть просто сложными машинами, механически выполняющими алгоритмические инструкции, вы увидите, что область компьютерных наук является, по сути, несомненно, творческой. Создание и применение новых алгоритмов является сугубо человеческой деятельностью, которая определяется нашим врожденным желанием применить свои инструменты для решения различных задач в окружающем мире. Компьютерные науки не только расширяют формы выражения, охватывающие изобразительное, литературное и музыкальное искусство, но также предоставляют новые способы их цифрового представления, пронизывающие весь современный мир.

Разработка крупной программной системы мало напоминает приготовление блюда по рецепту, найденному в поваренной книге, — гораздо больше она напоминает создание новой изысканной скульптурной группы. Обдумывание и компоновка ее общей структуры и функций требуют тщательного планирования. Создание ее компонентов требует времени, внимания к деталям и практических умений. Конечный продукт воплощает дизайнерскую эстетику и эмоциональные качества ее создателей.

---

## Данные

---

Компьютеры способны представить любую информацию, которую можно представить дискретными значениями и оцифровать. Алгоритмы позволяют обрабатывать или преобразовывать такую представленную в цифровом виде информацию бесчисленным количеством способов. Результатом этих действий является не просто перенос цифровых данных из одной части компьютера в другую; компьютерные алгоритмы позволяют находить шаблоны, создавать имитации и сопоставлять связи такими способами, которые генерируют новые знания и понимание. Громадные объемы сохраняемых данных, высокоскоростные компьютерные сети и мощные вычислительные инструменты позволяют делать важнейшие открытия во многих других областях естественных наук, техники и гуманитарных наук. Независимо от того, предсказывается ли эффект от воздействия нового лекарственного препарата путем моделирования сложной объемной структуры белка, статистически анализируется эволюция языка на протяжении столетий на основе оцифрованных книг или визуализируется

трехмерные изображения внутренних органов при неинвазивном медицинском сканировании, именно данные ведут к современным открытиям по всему фронту исследовательской деятельности человека.

Вот некоторые из вопросов о данных, которые обсуждаются в этой книге.

- Как в компьютерах сохраняются данные обычных цифровых образов таких типов информации, как числа, текст, изображения, звук и видео?
- Как в компьютерах аппроксимируются данные различных аналоговых образов, полученных из реального мира?
- Как компьютеры обнаруживают и предотвращают появление ошибок в данных?
- Какими могут быть последствия от наличия постоянно растущей в объеме и взаимосвязанной цифровой совокупности данных, имеющихся в нашем распоряжении?

---

## Программирование

---

Перевод намерений человека в такой алгоритм, который сможет выполнить машина, сейчас чаще всего называют *программированием*, хотя при изобилии языков и прочих инструментов, доступных в наше время, этот процесс сейчас мало напоминает программирование компьютеров, каким оно было в 1950- и ранние 1960-е годы. Хотя компьютерные науки включают намного больше, чем просто написание компьютерных программ, способность решать задачи посредством разработки выполнимых алгоритмов (программ) остается фундаментальным умением для каждого специалиста по компьютерным наукам.

Аппаратные средства компьютеров способны выполнять только относительно простые алгоритмические шаги, но абстракция, обеспечиваемая компьютерными языками программирования, позволяет человеку обдумывать, находить и кодировать решения гораздо более сложных проблем. Несколько основных вопросов, которые очерчивают рамки обсуждения этой темы в книге, можно сформулировать следующим образом.

- Как создаются программы?
- Ошибки каких типов могут иметь место в программах?
- Как ошибки в программах обнаруживаются и устраняются?
- Каким может быть эффект от наличия ошибок в современных программах?
- Как программы документируются и оцениваются?

---

## Интернет

---

Глобальная сеть Интернет соединила между собой компьютеры и электронные устройства по всему миру, и это оказало громадное влияние на то, как современное технологическое общество хранит, извлекает информацию и делится ею. Коммерция, новости, развлечения и коммуникации теперь во все большей степени зависят от этой паутины соединений между компьютерными сетями меньших размеров. Наше обсуждение этой темы предусматривает не только рассмотрение методов построения и функционирования самой этой системы, но и анализ многих других аспектов жизни человеческого общества, которое теперь накрепко переплетено с глобальной сетью.

Доступность Интернета также оказала глубокое влияние на степень конфиденциальности и уровень защищенности личной информации. Киберпространство несет в себе множество опасностей. Соответственно, криптография и киберзащита имеют все возрастающую важность для нашего тесно связанного мира.

---

## Влияние на общество

---

Компьютерные науки оказали глубокое влияние не только на технологии, которые мы используем для коммуникации, работы и развлечений; их бурное развитие имело огромные социальные последствия. Прогресс в компьютерных науках размывает множество ограничений, на основании которых наше общество принимало решения в прошлом, и ставит под сомнение многие из давних принципов общества. В правоведении возникают вопросы относительно того, в какой степени возможно владение программным обеспечением, а также относительно прав и обязанностей, накладываемых этим правом собственности. В этике люди сталкиваются со множеством аспектов, бросающих вызов традиционным принципам, на которых основано поведение человека. Что касается правительств, то здесь возникают вопросы относительно допустимой степени регулирования компьютерной технологии и ее применения. В философии этот прогресс породил противоречия между наличием интеллектуального поведения и наличием самого интеллекта. А во всем обществе он вызвал острые споры о том, что собой представляют новые приложения: новые свободы или новые средства контроля.

Такие темы очень важны для тех, кто выбрал для себя карьеру в компьютерных науках или связанных с ними областях. Открытия в науке иногда вызывают противоречивые применения, вызывая серьезное недовольство в отношении сделавших их исследователей. Более того, вместо успешной карьеры иной раз можно быстро сойти с рельс и оказаться на обочине из-за совершенной этической ошибки.

Способность справляться с дилеммами, создаваемыми продвижением компьютерных технологий, также важна и для тех, кто находится вне их непосредственной сферы. И действительно, эти технологии настолько быстро пронизывают общество, что лишь очень немногие, если таковые вообще имеются, способны остаться независимыми от их воздействия.

В этой книге вам будет предоставлено техническое обоснование, необходимое для рационального подхода к дилеммам, генерируемым компьютерными науками. Однако одних технических знаний в этой области науки будет недостаточно для решения всех возникающих вопросов. Принимая это во внимание, мы включили в материал книги несколько разделов, посвященных социальным, этическим и правовым аспектам влияния компьютерных наук на общество. В них обсуждаются, например, проблемы безопасности, вопросы права собственности и ответственности за программное обеспечение, социальное воздействие технологии баз данных и последствия достижений в области искусственного интеллекта.

Более того, часто не существует окончательного правильного решения проблемы, а многие имеющие силу решения являются компромиссами между противоположными (и возможно, одинаково правомерными) представлениями. Поиск решений в подобных случаях часто требует умения выслушивать и признавать иные точки зрения, проводить рациональное обсуждение и менять собственное мнение по мере приобретения новых знаний. По этой причине в конце каждой главы этой книги приводится подборка вопросов, объединенных под заголовком “Общественные и социальные вопросы”, в которых исследуется связь между компьютерными науками и обществом. Эти вопросы не предполагают немедленных и однозначных ответов, они предназначены, прежде всего, для обдумывания и обсуждения. Во многих случаях ответ, который кажется очевидным на первый взгляд, перестанет вас удовлетворять после ознакомления с альтернативами. Если говорить коротко, назначение этих вопросов не в том, чтобы вы немедленно предоставили “правильный” ответ, а скорее в том, чтобы повысить вашу осведомленность, в том числе ваше понимание позиций различных заинтересованных сторон в той или иной проблеме, ваше понимание альтернатив и ваше понимание как краткосрочных, так и долгосрочных последствий принятия этих альтернатив.

Философы предоставили множество подходов к этическим аспектам в поисках фундаментальных теорий, которые позволяют вывести принципы принятия руководящих решений и правил поведения.

Этика характера, иначе называемая этикой добродетели, была разработана Платоном и Аристотелем, утверждавшими, что “хорошее поведение” не является результатом применения идентифицируемых правил, но является лишь естественным следствием “хорошего характера”. В противоположность этому другие этические учения, такие как этика последствий, этика служебная, и

этика соблюдения обязательств, предполагают, что человек для решения этической дилеммы должен дать ответ на вопрос “Какими будут последствия?”, “Каков мой служебный долг?”, “Какие у меня обязательства?” соответственно, тогда как этика добродетели предлагает решать дилеммы поиском ответа на вопрос “Кем я хотел бы стать?” Следовательно, хорошее поведение достигается посредством развития хорошего характера, что обычно является результатом здорового воспитания и развития добродетельных привычек.

Именно этика добродетели положена в основу того подхода, который обычно принимается при “обучении” этике профессионалов в различных областях. Вместо представления определенных этических теорий, этот подход заключается в рассмотрении конкретных примеров, в которых разбирается определенное множество этических вопросов в той области, в которой эти лица являются экспертами. В результате обсуждения всех конкретных “за” и “против” в этих примерах, обучающиеся становятся более осведомленными, проницательными и восприимчивыми к опасностям, которые могут скрываться в их профессиональной жизни, и, таким образом, укрепляют свой характер. Это как раз тот стиль, в котором представлены вопросы в отношении социальных аспектов компьютерных наук в конце каждой главы этой книги.

## СОЦИАЛЬНЫЕ И ОБЩЕСТВЕННЫЕ ВОПРОСЫ

Следующие вопросы приводятся для того, чтобы помочь вам разобраться в некоторых этических, общественных и юридических аспектах использования вычислительной техники. Задача не сводится к тому, чтобы просто дать ответ на предложенные вопросы. Вы должны также понять, почему вы ответили именно так, а не иначе, и насколько ваши суждения по различным вопросам согласуются друг с другом.

1. Предположение, что нынешнее общество *отличается* от того общества, которое могло бы быть без компьютерной революции, как правило, принимается большинством. Является ли наше общество *лучше* или хуже некомпьютеризованного? Могли бы вы дать иной ответ на этот вопрос, если бы занимали другое положение в обществе?
2. Возможно ли участие в жизни современного техногенного общества без понимания основ этой технологии? Например, обязаны ли члены демократического государства, чье решение на выборах часто определяет, как технология будет поддерживаться и использоваться, осмыслить эту технологию? Зависит ли ваш ответ от того, какая именно технология имеется в виду? Например, будет ли ваш ответ на вопрос о ядерной технологии таким же, как и при рассмотрении компьютерной технологии?

3. Используя наличные деньги, люди обычно имеют право проводить финансовые операции без дополнительной платы за обслуживание. Однако в процессе автоматизации экономики финансовые учреждения сделали платными услуги за доступ к своим автоматизированным системам. Верно ли утверждение, что эти доплаты несправедливо ограничивают доступ частных лиц к экономике? Предположим, что работодатель платит своим работникам только чеками, а все финансовые учреждения ввели оплату за выдачу наличных денег по чеку или за помещение их на счет. Справедливо ли это по отношению к работникам? А как быть в том случае, если работодатель настаивает на оплате труда только посредством помещения суммы на счет в банке?
4. Если говорить об интерактивном телевидении, то в каких пределах можно разрешить компании получать от детей информацию относительно состояния домашних дел (возможно, через интерактивные игры). Например, можно ли разрешать компаниям получать от ребенка информацию о покупках, сделанных им или его родителями? А как насчет информации о самом ребенке?
5. В какой мере правительственные органы должны регулировать развитие компьютерных технологий и их применение? Например, можно еще раз вернуться к проблемам, упомянутым в вопросах 3 и 4. Чем можно оправдать введение правительственного регулирования?
6. В какой степени на наших внуков повлияют решения, принятые нами относительно технологии в целом и компьютерной технологии в частности?
7. По мере развития технологии наша образовательная система вынуждена постоянно пересматривать уровень абстракции, установленный в отношении подачи отдельных тем. Многие вопросы касаются того, действительно ли необходимо какое-либо умение или можно позволить студентам пользоваться только абстрактным инструментом? Студентов, изучающих тригонометрию, уже не учат определять значения тригонометрических функций с помощью таблиц. Вместо этого они используют калькуляторы как абстрактное средство определения этих значений. Некоторые утверждают, что письменное деление чисел в столбик также обеспечивает определенную абстракцию. Какие еще есть дисциплины, характеризующиеся подобными разногласиями? Может ли наличие автоматической проверки правописания устранить необходимость в грамотном письме? Может ли случиться так, что с использованием видеотехники когда-нибудь отпадет потребность в чтении?
8. Предполагается, что концепция публичных библиотек построена в основном на утверждении, что все граждане демократического общества

должны иметь доступ к информации. Так как все больше информации сохраняется и распространяется с помощью компьютерных технологий, является ли доступ к таким технологиям правом каждого человека? Если это так, то должны ли публичные библиотеки стать средством, обеспечивающим этот доступ?

9. Какие вопросы этического характера возникают в обществе, которое полагается на использование абстрактных инструментов? Есть ли такие случаи, когда использование товара или услуги без знания того, как это осуществляется, является неэтичным? А без знания того, как это производится? Или без понимания побочных последствий использования этого товара или услуги?
10. По мере того, как наше общество становится все более и более автоматизированным, правительствам становится проще следить за деятельностью граждан. Как вы считаете, хорошо это или плохо?
11. Какие технологии, которые Джордж Оруэлл описал в своем романе-антиутопии *1984*, теперь стали реальными? Используются ли они для тех же целей и тем же способом, как это предсказано в романе Оруэллом?
12. Если бы у вас была машина времени, в какой период истории вы хотели бы жить? Какие современные технологии вы хотели бы взять с собой? Можно было бы выбранные вами технологии взять с собой, не взяв прочие? До какой степени одну технологию можно отделить от другой? Согласуется ли протест против глобального потепления с безусловным принятием современного уровня медицинской помощи?
13. Предположим, что работа требует от вас проживания в иной культуре. Должны ли вы сохранить верность этическим нормам своей культуры или принять иную этику той культуры, в которой живете? Будет ли ваш ответ зависть от того, с чем именно связана проблема, скажем, со стилем одежды или с правами человека? Какие этические стандарты должны превалировать, если вы продолжаете жить в своей родной культуре, но ведете свой бизнес через Интернет в странах с иной культурой?
14. Стало ли общество более зависимым от компьютерных приложений в области коммерции, коммуникаций или социального общения? Например, какими могут быть последствия от долговременного прерывания работы Интернета и/или сотовой телефонной связи?
15. Большинство смартфонов способны определить свое местоположение посредством GPS. Это позволяет приложениям предоставлять вам информацию, связанную с вашим текущим местоположением (например, новости, местная погода или наличие учреждений, предприятий или организаций поблизости). Однако подобные возможности GPS могут также



позволить другим приложениям предоставлять сведения о местонахождении смартфона любым другим заинтересованным сторонам. Хорошо это или плохо? Как можно злоупотреблять информацией о местонахождении смартфона (а значит, и о вашем местоположении)?

## РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Goldstine J.J. *The Computer from Pascal to von Neumann*. — Princeton: Princeton University Press, 1972.
2. Haigh T., Priestley M., Rope C. *ENIAC in Action: Making and Remaking the Modern Computer*. — Cambridge, MA: The MIT Press, 2016.
3. Kizza J.M. *Ethical and Social Issues in the Information Age*, 3rd ed. — London: Springer-Verlag, 2007.
4. Mollenhoff C.R. *Atanasoff: Forgotten Father of the Computer*. — Ames, IA: Iowa State University Press, 1988.
5. Neumann P.G. *Computer Related Risks*. — Boston, MA: Addison-Wesley, 1995.
6. Ni L. *Smart Phone and Next Generation Mobile Computing*. — San Francisco: Morgan Kaufmann, 2006.
7. Quinn M.J. *Ethics for the Information Age*, 5th ed. — Boston, MA: AddisonWesley, 2012.
8. Randell B. *The Origins of Digital Computers*, 3rd ed. — New York: SpringerVerlag, 1982.
9. Spinello R.A., Tavani H.T. *Readings in CyberEthics*, 2nd ed. — Sudbury, MA: Jones and Bartlett, 2004.
10. Swade D. *The Difference Engine*. — New York: Viking, 2000.
11. Tavani H.T. *Ethics and Technology: Ethical Issues in an Age of Information and Communication Technology*, 4th ed. — New York: Wiley, 2012.
12. Woolley B. *The Bride of Science: Romance, Reason, and Byron's Daughter*. — New York: McGraw-Hill, 1999.

