От издателей русского перевода

На мировом рынке компьютерной литературы существует множество книг, предназначенных для обучения основным алгоритмам и используемых при программировании. Их довольно много, и они в значительной степени конкурируют между собой. Однако среди них есть особая книга. Это трехтомник "Искусство программирования" Д. Э. Кнута, который стоит вне всякой конкуренции, входит в золотой фонд мировой литературы по информатике и является настольной книгой практически для всех, кто связан с программированием.

Мы как издатели видим ценность книги в том, что она предназначена не столько для обучения технике программирования, сколько для обучения, если это возможно, "искусству" программирования, предлагает массу рецептов усовершенствования программ и, что самое главное, учит самостоятельно находить эти рецепты.

Ни для кого не секрет, что наши программисты являются одними из наиболее высококвалифицированных специалистов в мире. Они достойно представляют за рубежом отечественную школу программирования и информатики, которая внесла значительный вклад в формирование фундаментальных основ компьютерных наук. Для сохранения такого уровня и продвижения вперед необходимо своевременное издание на русском языке книг, отражающих основные мировые достижения в этой области. Трехтомник "Искусство программирования" Д. Э. Кнута — одна из таких книг.

Мы гордимся тем, что библиотеки программистов, преподавателей, студентов, старшеклассников и многих других пополнятся этой классической книгой и что тем самым мы внесем свой вклад в формирование более глубокого понимания основ компьютерных наук. Мы глубоко убеждены, что книга "Искусство программирования" Д. Э. Кнута способна приблизить человека к совершенству. Надеемся, наше издание на русском языке этой замечательной книги еще раз подтвердит, что истинные ценности с годами не устаревают.

– Виктор Штонда, Геннадий Петриковец, Алексей Орлович, издатели

О книге "Искусство программирования"

У каждой книги своя судьба. Одни появляются незаметно и так же незаметно исчезают в потоке времени, покрываясь пылью на полках библиотек. Другие в определенный период пользуются спросом у узкого круга специалистов, пока им на смену не приходят новые справочники. Третьи, поднимаясь над временем, оказывают мощное влияние на технологическое развитие общества. Книг, относящихся к последней категории, не так уж и много. Их выход в свет — всегда праздник. Проходят годы, изменяются технологии, но новые поколения с постоянным интересом перечитывают их страницы. Именно к таким книгам относится предлагаемый читателю многотомный труд известного американского ученого Дональда Эрвина Кнута "Искусство программирования".

Прошло почти 30 лет со времени первого издания в 1972 году в США этой книги. Она была переведена на большинство языков мира, в том числе и на русский. К настоящему времени на территории стран СНГ трехтомник Д. Э. Кнута стал библиографической редкостью. В 1998 году в США вышло третье издание "Искусства программирования". В нем сохранена последовательность изложения материала прежних версий, но значительно расширен список ссылок, в который включены свежие и наиболее важные результаты, добавлены новые упражнения и комментарии, устранены неточности. Учитывая популярность во всем мире "Искусства программирования", давно следовало ожидать появления нового переводного издания на русском языке, которое вы и держите в руках.

В чем же успех "Искусства программирования" Д. Э. Кнута?

Во-первых, эта книга — великолепное учебное пособие по составлению и анализу компьютерных алгоритмов. Ее разделы могут быть включены во многие университетские курсы по технологиям программирования, теории алгоритмов, дискретной математике. Книгу могут изучать и школьники старших классов, знакомые с основами программирования. В качестве основного языка записи алгоритмов автор выбрал язык машинных команд гипотетического универсального компьютера MIX. Это позволяет строить оптимальные программы с учетом особенностей вычислительных машин. Перенести MIX-программы на реальные ЭВМ или переписать их на языках высокого уровня не составляет особого труда. Логика работы программ почти всегда поясняется простыми блок-схемами.

Во-вторых, тщательно подобранный материал, вошедший в книгу, включает в себя основные фундаментальные классы алгоритмов, которые в том или ином виде наиболее часто встречаются в практике программирования.

В-третьих, немаловажным фактором успеха книги Д. Э. Кнута является энциклопедичность изложения. Профессор Кнут отличается уникальной способностью отслеживать проблему от исторических предпосылок ее зарождения до современного состояния. Многочисленные ссылки на работы старых мастеров (вплоть до времен античности), заключенные в современный контекст, создают у читателя особое чувство причастности к историческому развитию научных идей и методов.

В-четвертых, следует отметить мастерство изложения. Книга рассчитана на широкий круг читателей — от начинающих студентов до программистов-профессионалов. Каждому будет интересно изучать компьютерные алгоритмы на своем уровне. Материал

самодостаточен. Для понимания сути методов не требуется знания особых разделов математики или специальных технологий программирования. Прослеживается определенная "музыкальная" композиция сюжетного построения (дома у Д. Э. Кнута есть небольшой орган, на котором он играет).

Список составляющих успеха "Искусства программирования" можно легко продолжить.

Автор этих строк прослушал курс "Искусство программирования" в изложении профессора Кнута в 1976–1977 годах во время стажировки в Станфордском университете. Тогда формировалась алгоритмическая основа технологий программирования, у истоков которой стоял Д. Э. Кнут. Было много обсуждений, семинаров, творческих замыслов.

Значительные книги всегда связаны с судьбой автора. Дональд Эрвин Кнут начал работу над "Искусством программирования" в 1962 году. Продолжает ее и сейчас. У него много планов. Впереди новые тома "Искусства программирования", которых с нетерпением ждут читатели.

— Профессор Анатолий Анисимов

От редактора перевода

Со времени первого издания книги "Искусство программирования" Д. Э. Кнута прошло около 25 лет. Тем не менее книга не только не устарела, но по-прежнему остается основным руководством по искусству программирования, книгой, по которой учатся понимать суть и особенности этого искусства.

За эти годы на английском языке вышло уже третье издание 1-го и 2-го томов, а также второе издание 3-го тома. Автор внес в них значительные изменения и существенные дополнения. Достаточно сказать, что число упражнений практически удвоилось, а многие упражнения, включенные в предыдущие издания (особенно ответы к ним), модифицированы. Существенно дополнены и переделаны многие главы и разделы, исправлены неточности и опечатки, добавлены многочисленные новые ссылки на литературу, использованы теоретические результаты последних лет.

Значительно преобразилась глава 3, особенно разделы 3.5 и 3.6, а также разделы 4.5.2, 4.7, 5.1.4, 5.3, 5.4.9, 6.2.2, 6.4, 6.5 и др.

Естественно, возникла необходимость в новом издании книги.

Перевод выполнен по третьему изданию 1-го и 2-го томов и второму изданию 3-го тома. Кроме того, учтены дополнения и исправления, любезно предоставленные автором.

При переводе мы старались сохранить стиль автора, обозначения и манеру изложения материала. В большинстве случаев использовались термины, принятые в научной литературе на русском языке. При необходимости приводились английские эквиваленты. По многим причинам, в частности из-за сложности некоторых разделов, читать книгу "Искусство программирования" далеко непросто. Одной из причин, которые затрудняют понимание книги, является манера изложения автора; привыкнув к ней, можно существенно облегчить чтение.

Из-за обилия материала (часто мало связанного между собой) невозможно построить книгу так, чтобы различные понятия и определения вводились сразу же при первом упоминании о них. Поэтому в главе 1 могут обсуждаться без ссылок понятия, строгие определения которых приводятся в 3-м томе. Именно поэтому так велика роль предметного указателя, без которого понимание книги было бы существенно затруднено. Надеемся, что читатель не будет удивлен, найдя ссылки на главы 7, 8 и последующие не вошедшие в предлагаемые три тома главы. Мы вместе с автором надеемся, что очень скоро они будут опубликованы и, безусловно, сразу же появятся в русском переводе в качестве продолжения этого издания.

Следует также обратить внимание на далеко не всегда стандартные обозначения, которыми пользуется автор. Так же, как и определения, эти обозначения могут появиться в 1-м томе, а вводиться во 2-м. Поэтому без указателя обозначений пользоваться книгой было бы чрезвычайно трудно. Хочу также обратить внимание на запись `[A]`, где `A` — некоторое высказывание. Эта запись встречается в формулах, а иногда и в тексте, и обозначает величину, равную индикатору `A`.

— Профессор Ю. В. Козаченко

ПРЕДИСЛОВИЕ

Уважаемые читатели!
Вы держите в руках книгу,
издать которую Вы просили нас в тысячах писем.
Нам пришлось потратить годы на то,
чтобы самым тщательным образом проверить
и перепроверить бесконечное множество рецептов
и отобрать для вас самые лучшие, самые интересные,
самые совершенные.
Теперь без тени сомнения мы можем сказать,
что если вы будете следовать инструкциям,
то каждое блюдо будет получаться у вас таким же, как и у нас,
даже если раньше вы никогда не занимались
приготовлением пищи.

— *Поваренная книга Мак-Колла* (1963)

Процесс подготовки программ для цифрового компьютера — это очень увлекательное занятие. И дело не только в том, что оно оправдывает себя с экономической и научной точек зрения; оно может вызвать также эстетические переживания, подобные тем, которые испытывают творческие личности при написании музыки или стихов. Вы держите в руках первый том многотомного издания, цель которого — дать читателю разнообразные знания и умения, из которых и состоит ремесло программиста.

Последующие главы *не* являются введением в компьютерное программирование; предполагается, что вы уже имеете некоторый опыт в этой области. На самом деле предъявляемые к читателю требования очень просты; тем не менее начинающему программисту потребуются время и практика, чтобы понять, что собой представляет цифровой компьютер. Итак, читатель должен иметь

a) некоторое представление о том, как работает цифровой компьютер с хранимой программой; при этом необязательно разбираться в электронике, главное — понимать, каким образом команды можно сохранять в памяти компьютера, и затем последовательно их выполнять;

b) способность ставить задачу с помощью четких и определенных терминов, понятных компьютеру (у компьютеров нет разума, присущего человеку, поэтому они делают в точности то, что им приказывают, не больше и не меньше; именно этот факт обычно труднее всего уяснить начинающим пользователям);

c) знание самых простых компьютерных методов, таких как организация циклов (повторное выполнение некоторого набора команд), а также использование подпрограмм и переменных с индексами;

d) знание распространенных компьютерных терминов, например "память", "регистры", "биты", "плавающая точка", "переполнение", "программное обеспечение"; большинство терминов, которые не определены в тексте, поясняются в алфавитном указателе в конце каждого тома.

Эти четыре условия, вероятно, можно объединить в одном требовании: читатель должен иметь опыт написания и отладки по меньшей мере четырех программ хотя бы для одного компьютера.

Я старался писать эти книги так, чтобы они могли служить нескольким различным целям. Во-первых, они представляют собой справочное пособие, в котором сосредоточены знания из нескольких важных областей науки. Во-вторых, они могут использоваться в качестве пособий для самообразования и учебников по программированию или информатике для университетов. В связи с этим я включил в текст большое количество упражнений и предоставил ответы на большинство из них. Кроме того, я попытался сосредоточить внимание на фактах, вместо того чтобы "лить воду" и заниматься общими рассуждениями.

Этот трехтомник предназначен для всех, кто серьезно интересуется компьютерами, а не только для профессионалов. В сущности, одна из моих главных целей состояла в том, чтобы сделать методы программирования более доступными для специалистов из других областей. Как правило, эти специалисты получают большие преимущества, используя компьютеры, но не могут позволить себе тратить время на поиски необходимой информации, крупицы которой разбросаны по множеству технических журналов.

Тему этих книг можно сформулировать следующим образом: "Нечисленный анализ". Компьютеры обычно ассоциируются с решением численных задач, таких как нахождение корней уравнения, численное интерполирование, интегрирование и т. д. Но в этом трехтомнике подобные темы не рассматриваются (за исключением случаев, когда это необходимо сделать по ходу изложения). Численное компьютерное программирование — необычайно интересная и бурно развивающаяся область; на эту тему написано очень много хороших книг. Но с 60-х годов компьютеры все чаще и чаще применяются для решения проблем, в которых числа играют второстепенную роль. Теперь на первый план выходит способность компьютера принимать решения, а не просто выполнять арифметические операции. При решении нечисленных задач иногда требуется выполнять операции сложения и вычитания, но потребность в умножении и делении возникает довольно редко. Но, конечно, даже тот, кто в основном занимается численным компьютерным программированием, только выиграет от изучения нечисленных методов, так как они лежат и в основе числовых программ.

Результаты исследований в области нечисленного анализа разбросаны по многим техническим журналам. Моя цель состояла в том, чтобы извлечь из этого огромного объема информации только фундаментальные методы, которые можно применять в разнотипных ситуациях программирования. Я попытался обобщить выбранную информацию, чтобы получить то, что в большей или меньшей степени можно назвать "теорией", а также показать, как применять эту теорию при решении различных практических задач.

Конечно, "нечисленный анализ" — крайне неудачное название для данной области науки. Оно неудачно прежде всего потому, что содержит только отрицание

другого понятия; гораздо лучше было бы выбрать более содержательный термин, не имеющий приставки "не". Название "обработка информации" охватывает более широкую область, чем рассматриваемый здесь материал, а "методы программирования" — более узкую. Я считаю, что для темы, освещаемой в данных книгах, самым подходящим является название *анализ алгоритмов*, которое можно расшифровать как "теория свойств некоторых компьютерных алгоритмов".

Полный набор книг, озаглавленный как *Искусство программирования*, имеет следующую основную структуру.

Том 1. Основные алгоритмы

Глава 1. Основные понятия
Глава 2. Информационные структуры

Том 2. Получисленные алгоритмы

Глава 3. Случайные числа
Глава 4. Арифметика

Том 3. Сортировка и поиск

Глава 5. Сортировка
Глава 6. Поиск

Том 4. Комбинаторные алгоритмы

Глава 7. Комбинаторный поиск
Глава 8. Рекурсия

Том 5. Синтаксические алгоритмы

Глава 9. Лексикографический поиск
Глава 10. Синтаксический анализ

В томе 4 рассматривается очень большая тема, поэтому на самом деле он состоит из трех отдельных книг (томов 4A, 4B и 4C). Планируется также выпуск двух дополнительных томов по более специализированным темам: том 6, *Теория языков* (глава 11), и том 7, *Компиляторы* (глава 12).

Я приступил к этой работе в 1962 году с намерением написать единственную книгу, содержащую все перечисленные главы, но вскоре понял, что необходимо глубоко рассматривать выбранные темы, а не просто "скользить по поверхности". В результате получился текст такого объема, что материала каждой главы оказалось более чем достаточно для изучения в течение одного университетского семестра. И стало ясно, что необходимо разбить материал на несколько отдельных томов. Я знаю, что книга, содержащая только одну-две главы, выглядит довольно странно, но решил сохранить первоначальную нумерацию глав, чтобы упростить перекрестные ссылки. Планируется выпуск сокращенного варианта томов 1–5, который будет служить более общим справочником и/или учебником для студентов; в нем будет содержаться основная часть материала данных томов, а более специальная информация будет опущена. В сокращенном издании будет сохранена такая же нумерация глав, как и в полном.

Том 1 можно рассматривать как "пересечение" полного набора глав, в том смысле, что он содержит основные сведения, которые используются во всех остальных

книгах. С другой стороны, тома 2–5 можно читать независимо один от другого. Том 1 — это не только справочник, который необходимо использовать как пособие при чтении других томов; он может служить также университетским учебником либо пособием для самообразования по теме *структуры данных* (основное внимание следует уделить главе 2) или *дискретная математика* (основное внимание следует уделить разделам 1.1, 1.2, 1.3.3 и 2.3.4), или *программирование на языке машинных команд* (основное внимание следует уделить разделам 1.3 и 1.4).

Эти главы написаны с другой точки зрения, чем та, которая используется в самых современных книгах по программированию, т. е. я не пытался научить читателя пользоваться чужим программным обеспечением. Вместо этого я стремился научить читателя писать собственные программы более высокого качества.

Моя первоначальная цель заключалась в том, чтобы познакомить читателей с передовыми научными исследованиями в каждой из рассматриваемых областей знания. Но очень сложно постоянно быть в курсе дел отрасли, которая является экономически выгодной; бурный рост компьютерной науки сделал невозможным осуществление моей мечты. Образно говоря, я очутился на берегу безбрежного океана, содержащего десятки тысяч маленьких результатов, которые были получены десятками тысяч талантливых людей по всему миру. Поэтому мне пришлось поставить перед собой новую цель — сосредоточиться на "классических" методах, которые останутся актуальными в течение многих десятилетий, и описать их как можно лучше по мере моих возможностей. В частности, я попытался проследить историю каждой темы и заложить прочный фундамент ее дальнейшего развития. Я старался использовать точную терминологию, согласованную с той, которая применяется в современных публикациях, и попытался сообщить обо всех известных идеях последовательного программирования, выделяющихся простотой и изяществом формулировок.

Теперь несколько слов о математическом содержании данного многотомного издания. Материал излагается так, что он вполне доступен даже лицам со средним образованием; более сложные фрагменты они смогут просмотреть или просто опустить. В то же время те, кто имеют склонность к математике, смогут изучить интересные математические методы, связанные с дискретной математикой. Подобная двойственность представления информации была достигнута, с одной стороны, за счет присвоения рейтинга каждому упражнению (чтобы читатель смог отличать сложные с математической точки зрения упражнения от простых), а с другой стороны, благодаря такой организации разделов, при которой главные математические результаты сформулированы *перед* доказательствами. Доказательства предлагается либо провести самостоятельно в качестве упражнений (ответы на которые приводятся в отдельном разделе), либо найти в конце раздела.

Читатель, который, главным образом, интересуется программированием, а не математикой, может прекратить чтение раздела, как только математический материал станет слишком сложным для восприятия. С другой стороны, читатель-математик найдет для себя очень много интереснейших фактов. Многие математические публикации на тему программирования были ошибочными, поэтому одна из целей данной книги — предоставить читателю правильное математическое обоснование предмета изложения. И так как я считаю себя математиком, моя прямая

обязанность — правильно (насколько смогу) изложить материал с математической точки зрения.

Для чтения большей части математического материала вполне достаточно знания элементарной математики, так как почти вся остальная теория разрабатывается здесь же. Но иногда у меня возникает необходимость в более глубоких теоремах теории комплексного переменного, теории вероятностей, теории чисел и т. д. В подобных случаях я ссылаюсь на книги, содержащие подробное изложение данных тем.

Самое сложное решение, которое мне пришлось принять при подготовке этих книг, касалось способа представления различных методов. Преимущества блок-схем и пошаговых описаний алгоритмов хорошо известны; эти вопросы обсуждаются в статье "Computer-Drawn Flowcharts" в журнале ACM *Communications*, Vol. 6 (September 1963), pages 555–563. Для описания любого компьютерного алгоритма необходим также формальный и точный язык. Поэтому мне нужно было решить, какой язык использовать: алгебраический, такой как ALGOL или FORTRAN, либо машинно-ориентированный. Вероятно, многие сегодняшние компьютерные специалисты не согласятся с моим решением использовать машинно-ориентированный язык, но я убедился, что это был правильный выбор. На то существуют следующие причины.

a) На программиста большое влияние оказывает язык, на котором написаны программы. В настоящее время превалирует тенденция к выбору самых простых, а не самых оптимальных для компьютера конструкций языка. А программист, который знает машинно-ориентированный язык, стремится использовать более эффективные методы и таким образом создает более совершенные программы.

b) Все нужные нам программы, написанные на машинно-ориентированном языке, за редким исключением будут иметь небольшой размер. А это значит, что при наличии компьютера, обладающего минимальной вычислительной мощностью, проблем с использованием таких программ у нас не возникнет.

c) Языки высокого уровня не подходят для обсуждения важных деталей, имеющих отношение к низкому уровню, таких как связь сопрограмм, генерирование случайных чисел, арифметика высокой точности и многие другие проблемы, связанные с эффективным использованием памяти.

d) Тот, кто серьезно интересуется компьютерами, должен хорошо знать машинный язык, так как он лежит в основе работы компьютера.

e) Некоторое знание машинного языка необходимо в любом случае, чтобы разобраться в выходных данных программ, приведенных во многих примерах.

f) Новые алгебраические языки входят и выходят из моды приблизительно каждые пять лет, в то время как я пытаюсь говорить о "вечных истинах".

С другой стороны, я признаю, что писать программы на языках высокого уровня и отлаживать эти программы значительно проще. В сущности, начиная с 1970 года, я сам редко использовал машинный язык низкого уровня для собственных программ, так как современные компьютеры обладают большим объемом памяти и высоким быстродействием. Но для решения многих проблем, рассматриваемых в данной

книге, наибольшее значение имеет искусство программирования. Например, некоторые комбинаторные вычисления нужно повторять триллионы раз, и мы сэкономим приблизительно 11,6 дней работы за счет того, что сократим время вычислений во внутреннем цикле всего на одну микросекунду. Аналогично имеет смысл приложить дополнительные усилия для написания программы, которая будет использоваться много раз в течение каждого дня на множестве компьютеров, тем более что написать эту программу нужно только один раз.

А если принять решение использовать машинно-ориентированный язык, то какому языку следует отдать предпочтение? Я мог бы выбрать язык для конкретной машины X, но тогда те, кто используют другой компьютер, подумают, что данная книга написана только в расчете на обладателей компьютера X. Более того, машина X, вероятно, имеет много характерных особенностей, для которых совершенно неприменим материал данной книги, но все же его необходимо изложить. И наконец, через два года фирма — производитель машины X выпустит машину $X+1$ или $10X$, и компьютер X больше никого не будет интересовать.

Чтобы решить эту проблему, я попытался разработать "идеальный" компьютер с очень простыми правилами работы (изучить которые можно, скажем, всего за час) и очень похожий на реальные машины. Для студента нет никакой причины избегать изучения характеристик различных компьютеров; после изучения одного языка все остальные будут усваиваться гораздо легче. Кроме того, серьезный программист должен быть готов к тому, что в ходе работы ему придется сталкиваться с различными машинными языками. Поэтому остается только один недостаток использования вымышленной машины — сложность запуска написанных для нее программ. К счастью, на самом деле это не проблема, так как много добровольцев предложили свои услуги по написанию имитаторов гипотетической машины. Такие имитаторы идеальны для учебных целей, и работать с ними даже проще, чем с реальным компьютером.

Я старался ссылаться на самые лучшие старые статьи по каждой теме, а также упоминать новые работы. Ссылаясь на литературные источники, я использовал стандартные сокращения для названий периодических изданий, за исключением наиболее часто цитируемых журналов, для которых применялись следующие сокращения.

CACM — Communications of the Association for Computing Machinery

JACM — Journal of the Association for Computing Machinery

Comp. J. — The Computer Journal (British Computer Society)

Math. Comp. — Mathematics of Computation

AMM — American Mathematical Monthly

SICOMP — SIAM Journal on Computing

FOCS — IEEE Symposium on Foundations of Computer Science

SODA — ACM-SIAM Symposium on Discrete Algorithms

STOC — ACM Symposium on Theory of Computing

Crelle — Journal für die reine und angewandte Mathematik

Например, "*CACM* **6** (1963), 555–563" означает ссылку на журнал, упомянутый в одном из предыдущих абзацев этого предисловия. Сокращение "*CMath*" я использовал также для обозначения книги *Конкретная математика*, на которую есть ссылка во введении к разделу 1.2.

Бóльшая часть технического материала этих книг приходится на упражнения. Если идея нетривиального упражнения принадлежала не мне, то я старался упомянуть ее автора. Ссылки на литературу обычно даются в тексте раздела либо в ответе к упражнению. Но во многих случаях в основе упражнений лежат неопубликованные материалы, на которые нельзя дать ссылки.

В течение долгих лет работы над этими книгами мне помогали многие люди, которым я благодарен от всей души. Прежде всего, я хочу выразить благодарность моей жене Джилл) за ее бесконечное терпение, за подготовку некоторых иллюстраций и за постоянную помощь во всем. Я признателен также Роберту В. Флойду (Floyd Robert W.) за то, что в 60-х годах он посвятил столько времени работе над улучшением и углублением данного материала. Тысячи других людей также оказали мне неоценимую помощь. Чтобы просто перечислить их имена, понадобилась бы еще одна такая книга! Многие из них любезно разрешили мне использовать их старые неопубликованные работы. Мои исследования в Калифорнийском технологическом институте и Станфордском университете щедро финансировались Национальным научным фондом (National Science Foundation) и департаментом морских исследований (Office of Naval Research). Издательство Addison-Wesley всегда оказывало мне всемерную помощь и поддержку с того самого времени, когда в 1962 году я приступил к работе над проектом. Мне кажется, что для всех этих людей лучшая благодарность — данная публикация. Она показывает, что их вклад привел к появлению книг, в которых, я надеюсь, мне удалось написать то, чего они ожидали.

Предисловие к третьему изданию

Потратив десять лет на разработку систем компьютерного набора METAFONT и TEX, я теперь могу осуществить свою мечту — применить эти системы для набора книги *Искусство программирования*. Наконец-то мне удалось внести полный текст этой книги в персональный компьютер и таким образом получить ее электронную версию, что позволит в дальнейшем вносить любые изменения в технологию печати и отображения на экране. Такой способ работы дал мне возможность сделать буквально тысячи улучшений; я добился того, о чем так долго мечтал.

В этом новом издании я смог проверить каждое слово в тексте, стараясь сохранить юношеский задор моих первоначальных исследований и в то же время внести бóльшую зрелость суждений. Были добавлены десятки новых упражнений, а на десятки старых даны новые или улучшенные ответы.

Таким образом, работа над книгой *Искусство программирования* продолжается. Именно поэтому некоторые части данной книги начинаются пиктограммой "В процессе построения" (это своеобразное извинение за то, что приведены не самые новые данные). Мои папки переполнены важными материалами, которые я планирую включить в окончательное, славное четвертое издание тома 1; оно выйдет,

вероятно, через 15 лет. Но сначала я должен закончить тома 4 и 5. Я хочу, чтобы они были опубликованы сразу же, как только будут готовы к печати.

Бóльшая часть тяжелой работы по подготовке этого нового издания была выполнена Филлис Винклер (Phyllis Winkler) и Сильвио Леви (Silvio Levy), которые профессионально набирали и редактировали текст второго издания, а также Джеффри Олдхэмом (Jeffrey Oldham), который конвертировал почти все оригинальные иллюстрации в формат METAPOST. Я исправил все ошибки, которые бдительные читатели (Бэрри) обнаружили во втором издании (а также ошибки, которые, увы, не заметил никто), и постарался избежать появления новых ошибок в новом материале. Тем не менее я допускаю, что некоторые огрехи все же остались, и хотел бы исправить их как можно скорее. Поэтому за каждую опечатку*, а также ошибку, относящуюся к сути излагаемого материала или к приведенным историческим сведениям, я охотно заплачу $2,56 тому, кто первым ее найдет. На Web-странице, адрес которой приведен на обороте титульной страницы, содержится текущий список всех ошибок, о которых мне сообщили**.

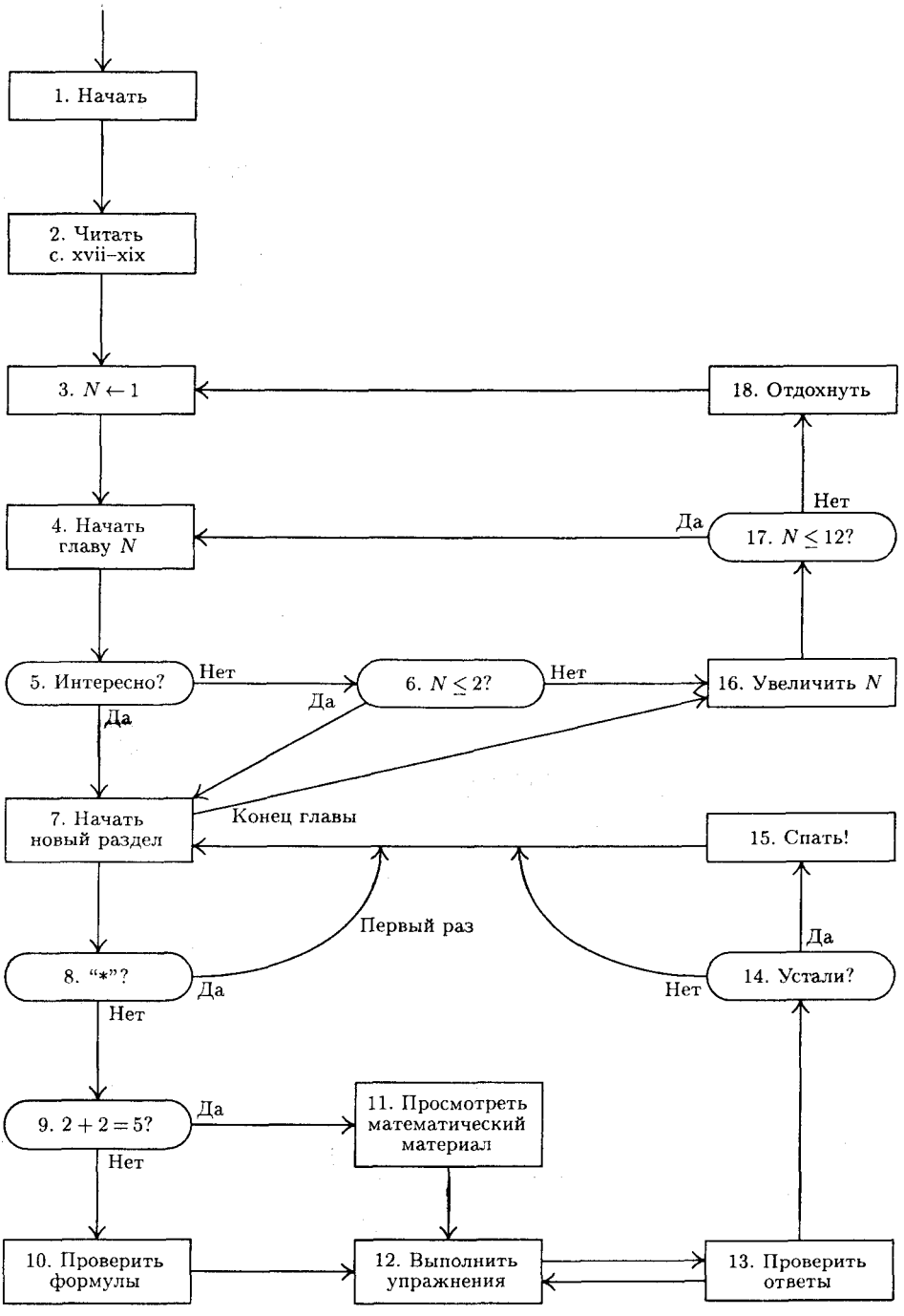
Станфорд, Калифорния
Апрель 1997

За последние двадцать лет мир изменился.

— Билл Гейтс (Bill Gates) (1995)

* Имеется в виду оригинал настоящего издания. — *Прим. ред.*

** Ошибки, известные на момент подготовки русского издания, были исправлены. — *Прим. ред.*



Блок-схема процедуры чтения книг этой серии.

Процедура чтения книг этой серии

1. Начните читать настоящую процедуру, если вы этого еще не сделали. *Продолжайте в точном соответствии с указанными шагами.* (Общая форма этой процедуры и сопровождающей ее блок-схемы будет использоваться на протяжении всей книги.)
2. Прочтите примечания к упражнениям (с. 23–25).
3. Установите N равным 1.
4. Начните чтение главы N. *Не* читайте эпиграфы, помещенные в ее начале.
5. Вам интересен предмет этой главы? Если да, перейдите к шагу 7; если нет, перейдите к шагу 6.
6. $N \leq 2$? Если нет, перейдите к шагу 16; если да, то все-таки просмотрите главу. (В главах 1 и 2 содержится важный вводный материал, а также обзор основных методов программирования. Эти главы следует хотя бы просмотреть, чтобы ознакомиться с условными обозначениями и компьютером MIX.)
7. Начните чтение следующего раздела главы; но, если вы уже дошли до конца главы, перейдите к шагу 16.
8. Отмечен ли номер раздела символом "*"? Если да, то при первом чтении этот раздел можно пропустить (в нем рассматривается специальный вопрос, который интересен, но не имеет первостепенного значения). Вернитесь к шагу 7.
9. Есть ли у вас наклонности к математике? Если математика для вас — китайская грамота, то перейдите к шагу 11; в противном случае перейдите к шагу 10.
10. Проверьте математические выкладки, выполненные в этом разделе (и сообщите автору о замеченных ошибках). Перейдите к шагу 12.
11. Если в текущем разделе содержится много математических выкладок, то лучше не читайте их. Тем не менее вам следует ознакомиться с основными результатами раздела; они, как правило, либо приводятся в самом начале раздела, либо выделены *курсивом* в самом конце сложной части текста.
12. Выполните рекомендуемые упражнения к разделу в соответствии с указаниями, приведенными в примечаниях к упражнениям (которые вы прочитали на шаге 2).
13. Поработав над упражнениями в свое удовольствие, сравните полученные ответы с теми, которые приведены в соответствующем разделе в конце книги (если для

этих задач даны ответы). Прочтите также ответы к упражнениям, над которыми у вас не было времени поработать. (*Замечание.* В большинстве случаев имеет смысл сначала прочесть ответ к упражнению n, а затем приступить к упражнению $n + 1$, поэтому шаги 12 и 13 обычно выполняются одновременно.)

14. Вы устали? Если нет, вернитесь к шагу 7.

15. Отправляйтесь спать. А когда проснетесь, вернитесь к шагу 7.

16. Увеличьте N на 1. Если $N = 3, 5, 7, 9, 11$ или 12, то возьмите следующий том этой серии книг.

17. Если N меньше или равно 12, то вернитесь к шагу 4.

18. Поздравляю! Теперь постарайтесь убедить друзей в том, что необходимо приобрести экземпляр тома 1 и начать его читать. Сами же возвращайтесь к шагу 3.

Горе тому, кто читает только одну книгу.

— ДЖОРДЖ ГЕРБЕРТ (GEORGE HERBERT), *Jacula Prudentum*, 1144 (1640)

Единственный недостаток всех литературных произведений в том, что они слишком длинны.

— ВОВЕНАРГ (VAUVENARGUES), *Réflexions*, 628 (1746)

Книги банальны. Гениальна только жизнь.

— ТОМАС КАРЛЕЙЛЬ (THOMAS CARLYLE), *Journal* (1839)

ПРИМЕЧАНИЯ К УПРАЖНЕНИЯМ

Упражнения, приведенные в этой серии книг, предназначены как для самостоятельной проработки, так и для семинарских занятий. Очень трудно и, наверное, просто невозможно выучить предмет, только читая теорию и не применяя ее для решения конкретных задач, которые заставляют задуматься о прочитанном. Более того, мы лучше всего заучиваем то, до чего дошли самостоятельно, своим умом. Поэтому упражнения занимают важное место в данном издании. Я приложил немало усилий, чтобы сделать их как можно более информативными, а также отобрать задачи, которые были бы не только поучительны, но и позволяли читателю получить удовольствие от их решения.

Во многих книгах простые упражнения даются вперемешку с исключительно сложными. Это не всегда удобно, так как читателю хочется знать заранее, сколько времени ему придется затратить на решение задач (иначе в лучшем случае он их только просмотрит). В качестве классического примера подобной ситуации можно привести книгу Ричарда Беллмана (Richard Bellman) *Динамическое программирование* (М.: Изд-во иностр. лит., 1960). Это очень важная, новаторская работа, но у нее есть один недостаток: в конце некоторых глав в разделе "Упражнения и научные проблемы" среди серьезных, еще нерешенных проблем, попадаются простейшие вопросы. Говорят, что кто-то однажды спросил д-ра Беллмана, как отличить упражнения от научных проблем, и он ответил: "Если вы можете решить задачу, значит, это упражнение; в противном случае это научная проблема".

Совершенно очевидно, что в книге, подобной этой, должны быть приведены и сложные научные проблемы, и простейшие упражнения. Поэтому, чтобы читатель не ломал голову, пытаясь отличить одно от другого, были введены рейтинги, которые определяют степень сложности каждого упражнения. Эти рейтинги имеют следующее значение.

Рейтинг *Объяснение*

00 Чрезвычайно простое упражнение, на которое можно ответить сразу же, если прочитанный материал понят. Упражнения подобного типа почти всегда можно решить "в уме".

10 Простая задача, которая заставляет задуматься над прочитанным, но не представляет особых трудностей. На ее решение вы затратите не больше минуты; в процессе решения могут понадобиться карандаш и бумага.

20 Средняя задача, которая позволяет проверить, понял ли читатель основные положения изложенного материала. Чтобы получить исчерпывающий ответ, может понадобиться примерно 15–20 минут.

30 Задача умеренной сложности. Для ее решения может понадобиться более двух часов (а если одновременно вы смотрите телевизор, то еще больше).

40 Достаточно сложная или трудоемкая задача, которую вполне можно включить в план семинарских занятий. Предполагается, что студент должен справиться с ней, затратив не слишком много времени, и решение будет нетривиальным.

50 Научная проблема, которая (насколько известно автору в момент написания книги) пока еще не получила удовлетворительного решения, хотя найти его пытались очень многие. Если вы нашли решение подобной проблемы, то опубликуйте его; более того, автор данной книги будет очень признателен, если ему сообщат решение как можно скорее (при условии, что оно правильно).

Интерполируя по этой "логарифмической" шкале, можно понять, что означает любой промежуточный рейтинг. Например, рейтинг *17* говорит о том, что упражнение немного проще, чем задача средней сложности. Если задача с рейтингом *50* будет впоследствии решена каким-либо читателем, то в следующих изданиях данной книги и в списке ошибок, опубликованных в Internet, она может иметь рейтинг *45* (адрес Web-страницы приводится на обороте титульной страницы).

Остаток от деления рейтинга на 5 показывает, какой объем рутинной работы потребуется для решения данной задачи. Таким образом, для решения упражнения с рейтингом *24* может потребоваться больше времени, чем для упражнения с рейтингом *25*, но для последнего необходим более творческий подход.

Автор очень старался правильно присвоить рейтинги упражнениям, но тому, кто составляет задачи, трудно предвидеть, насколько сложными они окажутся для кого-то другого. К тому же одному человеку некая задача может показаться простой, а другому — сложной. Таким образом, определение рейтингов — дело достаточно субъективное и относительное. Я надеюсь, что рейтинги помогут вам получить правильное представление о степени трудности задач, но их следует воспринимать в качестве ориентира, а не в качестве абсолюта.

Эта книга написана для читателей с различным уровнем математической подготовки и научного кругозора, поэтому некоторые упражнения рассчитаны исключительно на тех, кто серьезно интересуется математикой или занимается ею профессионально. Если рейтингу предшествует буква *M*, значит, математические понятия и обоснования используются в упражнении в большей степени, чем это необходимо тому, кто интересуется в основном программированием алгоритмов. Если же упражнение отмечено буквами *HM*, то для его решения необходимо знание высшей математики в большем объеме, чем дается в настоящей книге. Но пометка *HM* совсем необязательно означает, что упражнение трудное.

Перед некоторыми упражнениями стоит стрелка "▸", которая означает, что они особенно поучительны и их очень рекомендуется выполнить. Само собой разумеется, никто не ожидает, что читатель (или студент) будет решать *все* задачи, поэтому наиболее важные из них и были выделены. Но это ни в коем случае не означает, что другие упражнения выполнять не стоит! Каждый читатель должен хотя бы попытаться решить все задачи, рейтинг которых меньше или равен *10*. Стрелки помогут выбрать задачи с более высокими рейтингами, которые следует решать в первую очередь.

К большинству упражнений приведены ответы, помещенные в отдельном разделе в конце книги. Пожалуйста, пользуйтесь ими разумно: ответ смотрите только

после того, как приложите все усилия, чтобы решить задачу самостоятельно, либо если у вас совершенно нет времени на ее решение. Ответ будет поучителен и полезен для вас только в том случае, если вы ознакомитесь с ним *после* того, как найдете свое решение или изрядно потрудитесь над задачей. Ответы к задачам излагаются очень кратко и схематично, так как предполагается, что читатель честно пытался решить задачу собственными силами. Иногда в приведенном решении дается меньше информации, чем спрашивалось, но чаще бывает наоборот. Вполне возможно, что полученный вами ответ окажется лучше того, который помещен в книге, или вы найдете ошибку в ответе. В таком случае автор был бы очень признателен, если бы вы как можно скорее подробно сообщили ему об этом; тогда в последующих изданиях книги будет опубликовано более удачное решение, а также имя его автора.

Решая задачи, вы, как правило, можете пользоваться ответами к предыдущим упражнениям, за исключением случаев, когда это будет оговорено особо. Рейтинги упражнениям присваивались в расчете именно на это, и вполне возможно, что рейтинг упражнения $n + 1$ ниже рейтинга упражнения n, даже если результат упражнения n является его частным случаем.

Условные обозначения		*00*	Простейшее (ответ дать немедленно)
		10	Простое (на одну минуту)
▶	Рекомендуется	*20*	Средней трудности (на четверть часа)
M	С математическим уклоном	*30*	Повышенной трудности
HM	Требует знания высшей математики	*40*	Высокой трудности
		50	Научная проблема

УПРАЖНЕНИЯ

▶ **1.** [*00*] Что означает рейтинг *M20*?

2. [*10*] Какое значение для читателя имеют упражнения, которые приводятся в учебниках?

3. [*14*] Докажите, что $13^3 = 2197$. Обобщите ответ. (Скучных задач, подобных этой, автор старался избегать.)

4. [*HM45*] Докажите, что если n — целое число, $n > 2$, то уравнение $x^n + y^n = z^n$ неразрешимо в целых положительных числах x, y, z.

Мы обсудили этот вопрос со всех сторон.
Перед нами факты, изложенные систематично и по порядку.

— ЭРКЮЛЬ ПУАРО (HERCULE POIROT),
Убийство в восточном экспрессе (1934)