

ВЫСШЕЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАНИЕ

А. В. МОГИЛЕВ, Н. И. ПАК, Е. К. ХЕННЕР

ИНФОРМАТИКА

Под редакцией **Е. К. ХЕННЕРА**

Допущено

*Министерством образования и науки Российской Федерации
в качестве учебного пособия для студентов высших учебных заведений,
обучающихся по педагогическим специальностям*

8-е издание, стереотипное



Москва
Издательский центр «Академия»
2012

УДК 519.72(075.8)

ББК 32.81я73

М742

Р е ц е н з е н т ы:

доктор технических наук, директор Института информатизации образования

Министерства образования РФ, профессор Я.А. Ваграменко;

доктор педагогических наук, проректор Омского государственного

педагогического университета, профессор М.П. Лапчик

Могилев А. В.

М742 Информатика : учеб. пособие для студ. пед. вузов / А. В. Могилев, Н. И. Пак, Е. К. Хеннер ; под ред. Е. К. Хеннера. — 8-е изд., стер. — М. : Издательский центр «Академия», 2012. — 848 с.

ISBN 978-5-7695-9120-4

Содержатся обширные сведения по теоретическим основам информатики, программному обеспечению, языкам и методам программирования, вычислительной технике, информационным системам, компьютерным сетям и телекоммуникациям, компьютерному моделированию и социальной информатике.

Для студентов высших педагогических учебных заведений, обучающихся по специальности «Информатика». Может быть полезно студентам университетов и учителям информатики.

УДК 519.72(075.8)

ББК 32.81я73

*Оригинал-макет данного издания является собственностью
Издательского центра «Академия», и его воспроизведение любым способом без согласия
правообладателя запрещается*

© Могилев А. В., Пак Н. И., Хеннер Е. К., 2003

© Образовательно-издательский центр «Академия», 2004

© Оформление. Издательский центр «Академия», 2004

ISBN 978-5-7695-9120-4

ПРЕДИСЛОВИЕ*

Около трех с половиной десятилетий минуло с тех пор, как в педагогических вузах введено в качестве учебной дисциплины программирование для ЭВМ.

За все это время в педагогических вузах кафедры программирования и вычислительной математики, как и образованные позднее на их основе кафедры информатики и вычислительной техники, не были избалованы своевременным появлением специальных учебных изданий. При колоссальной скорости изменений в самом предмете, всегда существенно превышавшей скорость центральных издательских механизмов, специально ориентированные на программы педагогических вузов книги выходили не чаще, чем раз в десятилетие, — едва ли несоразмерно скорости смены поколений ЭВМ. Если к этому добавить непреходящую особенность книг о компьютерах и программах к ним устаревать еще в типографии, то можно с грустью утверждать, что нормальных учебников по Computer Science для педагогических вузов пока — увы! — так и не было. И можно ли считать утешением тот безрадостный факт, что уровень компьютерного обеспечения большинства педагогических вузов обычно с трудом «дотягивал» до содержания полуусталых учебных пособий?

В наши дни многое изменилось. Во-первых, на месте узкого учебного курса программирования в учебных планах педагогических вузов давно уже обосновался куда более обширный и многоаспектный учебный предмет «Информатика». Да и положение с источниками, конечно, стало другим — полки книжных магазинов сегодня ломятся от изданий по информатике.

Однако преподавателю (а более всего студенту) специальная учебная книга, содержание и направленность которой отвечают заданному учебному плану и программе (или, как сейчас говорят, стандарту образования), все-таки очень нужна. Сделать такую книгу непросто: с одной стороны — пугающая широта и вечная необустроенность предмета информатики, а с другой — динамичность, неуспокоенность самих стандартов педагогического образования, равно как и школьных стандартов, которые тоже именно сейчас находятся в состоянии активного переосмыслиния. Эти взаимосвязанные предпосылки достаточно долгое время оставляли мало надежд на то, что найдутся отважные авторы (преисполненные профессионального долга, а вовсе не амбициозного желания прославиться), которые возьмутся за такую работу и сделают ее. Те, кто держит в руках эту книгу, могут убедиться, что за дело взялись наиболее авторитетные и хорошо известные в кругу педагогов-информатиков учёные. Мне же приятно осознавать, что с этими замечательными людьми и специалистами нас не один год связывало Учебно-методическое объединение педагогических учебных заведений России по информатизации образования, в котором и родилась инициатива написания этого учебного пособия.

М. П. Лапчик,

доктор педагогических наук, профессор,
действительный член
Академии информатизации образования

* Данное предисловие написано проф. М. П. Лапчиком к 1-му изданию (1999 г.).

ОТ АВТОРОВ

Уважаемый читатель!

Эта книга прежде всего для тех, кто избрал областью профессиональной деятельности преподавание информатики. Такой выбор налагает двойную ответственность: необходимо знать все аспекты данного предмета на достаточно высоком уровне и уметь рассказать о нем, научить других.

Авторы надеются, что учебное пособие будет полезно не только студентам педагогических вузов и «классических» университетов, избравшим обучение информатике сферой своей будущей профессиональной деятельности, но и практикующим учителям информатики для самообразования и повышения квалификации.

Эта книга может быть полезна и многим другим, изучающим информатику. В первом приближении, всех тех, кого интересует информатика, можно разбить на три категории. Представители первой, самой многочисленной категории, используют несколько широко распространенных информационных технологий: обработку текстов и графической информации, работу в сетях и т.д., причем им не надо знать, как все это происходит. Таких можно назвать «принципиальными пользователями» (без малейших намерений обидеть, у большинства из них просто иная сфера профессиональных интересов). Для них наша книга содержит слишком много теоретических сведений и недостаточно указаний о «кнопочной» стороне дела, за которой мы часто отсылаем читателей к специальной литературе.

Вторая категория состоит из тех, кто интересуется не только реализацией информационных технологий, но и ответами на вопросы, «как это делается и почему именно так». Хочется верить, что на большую часть таких вопросов данный учебник дает ответ. Наконец, третья категория — будущие профессионалы, которые намереваются работать в одной из сфер информатики, — разработчики программного обеспечения (трансляторов, издательских систем, экспертных систем и т. п.), сетевики, администраторы крупных информационных систем и т. д. Для них по направлению избранной специализации скорее всего нужны более глубокие знания, а данное учебное пособие может стать книгой для первого ознакомления и общего обзора, ибо трудно профессионально углубиться во все сферы информатики в равной мере.

Современная информатика очень велика по объему и очень динамична. Если изучаемые в вузах курсы математики, лингвистики, химии и большинства других наук практически не изменяются на протяжении многих лет будущей профессиональной деятельности сегодняшнего студента, то в информатике это полностью оформленвшееся ядро сравнительно невелико. Как же тогда быть с ее изучением?

Прежде всего следует определиться, что такое информатика. В понимании многих людей это есть совокупность приемов и методов работы с компьютерами. На самом деле это не так: компьютеры являются лишь техническим средством, с помощью которого информатика реализует свой прикладной пользовательский аспект, правда, средством столь сложным и интересным, что оно способно поглотить массу внимания не только специалистов в области компьютерных технологий, но и непрофессионалов.

Информатика — комплекс научно-практических дисциплин, изучающих все аспекты получения, хранения, преобразования, передачи и использования информации. Более детальное обсуждение содержания информатики, отражение разных точек зрения проводится во введении. Однако из какого бы определения ни исходить, все согласны с тем, что у современной информатики есть два взаимодополняющих аспекта — научный и технологический. Первый является более устоявшимся, второй — весьма мобильным, хотя и в технологической части информатики есть вполне сформировавшееся ядро, которое мало подвержено изменениям.

Приведем примеры, опираясь на имеющийся у читателей опыт. Так, существует большое количество алгоритмических языков программирования, и допустим, что человеку, умеющему работать с Бейсиком или Паскалем, приходится браться за Си. Новая система обозначений, дополнительные возможности — на некоторое время это может полностью поглотить внимание, но постепенно приходит понимание: главное — навыки к алгоритмизации и структурированию данных, и если они есть, то кодирование алгоритмов на другом языке — дело не самое сложное. Или: исчерпаны возможности привычного текстового редактора (или он просто вышел из моды — так тоже бывает) и нужно переходить на новый. Если человек понимает принципы работы программ такого рода, имеет устойчивые навыки работы с одной из них, то освоить другую, даже с большими возможностями, обычно несложно. Подтверждается известная истинка: образование — это то, что остается, когда детали знаний забываются.

Итак, главное при изучении информатики — освоить фундаментальные понятия каждой из ее областей, ориентироваться в их взаимосвязи, приобрести навыки практической работы с важнейшими техническими и программными средствами. Добавим, что от того, кто хочет стать учителем, требуется в определенном смысле больше, чем, скажем, от инженера-разработчика ЭВМ или программиста: преподаватель должен знать практически все разделы информатики, владеть ее техническими средствами на уровне высококвалифицированного пользователя и уметь передавать свои знания и навыки другим людям.

Есть еще один, чрезвычайно важный, аспект подготовки будущего бакалавра или специалиста — научиться самостоятельно осваивать новые знания и навыки. Никакой учебник не может содержать всех необходимых сведений — тем более что в вашу подготовку будут входить, кроме базовых дисциплин, спецкурсы, выполнение курсовых и выпускных работ. Для того чтобы вам в этом помочь, приводятся справочные материалы, ссылки на книги, в которых обсуждаемые вопросы излагаются более детально. Если вы привыкнете пользоваться этим справочным аппаратом, дополнительной литературой, то никакие перемены в содержании курсов, которые вы сами будете в будущем преподавать, не страшны.

Данное учебное пособие охватывает основные разделы современной информатики. Чрезвычайно важна теоретическая база, закладываемая в гл. 1, — этоочно устоявшееся ядро нашей науки, которому не грозят существенные перемены.

Гл. 2 посвящена необычному вопросу о программном обеспечении ЭВМ. Ее цель — ознакомление с основными классами базового и прикладного программного обеспечения, фундаментальными принципами, заложенными в их основу. Разумеется, в ходе изучения этих вопросов предполагается выработка практических навыков пользования типичными программами каждого класса.

Гл. 3 специально названа не «языки программирования», а «языки и методы программирования». Будущий преподаватель не может себе позволить ограничиться глубоким изучением одного-двух языков, какими бы мощными и популярными они ни были; ему необходим широкий кругозор в этих вопросах. Различные мето-

дологии («парадигмы») программирования, описанные в этой главе, отражают различные стили, способы человеческого мышления.

Без отчетливого понимания основ функционирования вычислительной техники (гл. 4) нельзя всерьез освоить многие другие разделы информатики. Наиболее глубоким уровнем этого понимания, который от вас требуется, является уровень архитектуры ЭВМ, но и он весьма глубок и предполагает проникновение в новую понятийную область. Это невозможно без ознакомления с программированием на уровне команд микропроцессора. Необходимо также понимать принципы функционирования и уметь работать с многочисленными внешними устройствами ЭВМ.

Компьютерные сети и телекоммуникации — весьма популярный сегодня раздел прикладной информатики. Осознав возможность передачи текстовой, графической и иной информации на любые расстояния в кратчайшее время, получив экономическую возможность вовлечь в этот процесс миллионы людей, человечество буквально рванулось в мир компьютерных сетей. Этим вопросам посвящена гл. 5.

В гл. 6 рассказывается об информационных системах и базах данных, без которых невозможно представить себе современную информатику. Базы и банки данных, информационно-поисковые системы и автоматизированные управляющие системы, равно как и многие другие виды информационных систем, обсуждаются в этой главе. Много внимания уделяется, в частности, информационным системам в образовании.

Мощнейшей информационной технологией, которой полностью посвящена гл. 7, является компьютерное моделирование — интегративная дисциплина, включающая выходы в самые различные науки. Моделирование является важным методом человеческого познания, в котором компьютеры выступают как мощное техническое средство.

Гл. 8, посвященная социальной информатике, в предыдущих изданиях учебника отсутствовала. Ее появление связано со стремительно растущими социальными проявлениями процессов информатизации и с тем, что соответствующая проблематика реально вошла в предметную область современной информатики.

Курс информатики, который охватывает данное учебное пособие, должен включать в себя большое число практических занятий и лабораторных работ. В ходе их выполнения вы приобретете навыки в пользовании ЭВМ, в работе со многими программными продуктами, в создании относительно несложных программ, освоитесь в современных информационных технологиях. В качестве дополнительного руководства для практического освоении курса авторы рекомендуют созданный ими практикум (А. В. Могилев, Н. И. Пак, Е. К. Хеннер. Практикум по информатике. — М.: Изд. центр «Академия», 2001).

Важную роль при изучении информатики играют смежные дисциплины предметной подготовки — в первую очередь, математика, физика. В некоторых их разделах будут существенно углублены сведения, непосредственно примыкающие к информатике, по математической логике и теории алгоритмов, по электронике и физическим основам функционирования ЭВМ и др.

Последовательность изучения курса информатики может отличаться от той, в которой расположен материал в данной книге. Информатика не столь формализована, как, например, математика, и многие разделы информатики при изучении можно менять местами. Поэтому, встретив в книге новое для вас понятие, которое не объяснено, поищите разъяснение в другом разделе или справочнике, толковом словаре, которые хорошо бы всегда иметь на столе.

При отборе материала основой послужили важнейшие нормативные документы — Государственные образовательные стандарты (ГОС), определяющие требования к обязательному минимуму содержания и уровню подготовки бакалавров

образования по направлению «Физико-математическое образование», профилю «Информатика» и специалистов — учителей информатики (тем более, что авторы данного пособия участвовали в разработке предметного блока стандарта подготовки учителя информатики). Хотя стандарты периодически пересматриваются, их основа сохраняется.

При подготовке данного (3-го) издания авторы внесли в текст и структуру книги существенные изменения. Эти изменения продиктованы как опытом использования данного учебника в различных вузах России, так и быстро меняющимся содержанием некоторых (прежде всего технологических) разделов информатики и соответствующей образовательной области.

Основными читателями этой книги, как и предполагалось при ее создании, стали студенты педагогических вузов и преподаватели информатики — как школьные, так и вузовские. Вместе с тем, первые два издания нашли немало читателей и среди студентов других категорий, в том числе классических университетов и технических вузов. Внесенные в данное издание изменения и дополнения приближают учебное пособие к обновленным ГОС подготовки учителей информатики и в то же время позволяют расширить круг его пользователей.

Авторы учебного пособия — заведующие кафедрами информатики Воронежского, Красноярского и Пермского педагогических университетов, много лет отдавшие подготовку учителей информатики. В этой книге — их коллективный опыт.

В работе над пособием авторам оказывали помощь их коллеги. Особо следует отметить участие профессора М. П. Лапчика (Омский госпедуниверситет) в разработке и обсуждении структуры книги, доцентов О. Н. Лучко (Омский госпедуниверситет), Е. А. Еремина (Пермский госпедуниверситет), И. Я. Злотникову (Воронежский госпедуниверситет) и А. П. Шестакова (Пермский госпедуниверситет), которые предоставили часть материала для гл. 1, 4, 5 и 7 соответственно.

Желаем успеха читателям!

ВВЕДЕНИЕ

Возникновение информатики и смежные научные области

Информатика — молодая дисциплина, изучающая вопросы, связанные с поиском, сбором, хранением, преобразованием и использованием информации в самых различных сферах человеческой деятельности, преимущественно с использованием компьютеров.

До настоящего времени толкование термина «информатика» (в том смысле как он используется в современной научной и методической литературе) еще не является установленным и общепринятым. Обратимся к истории вопроса, восходящей ко времени появления электронных вычислительных машин. Однако отметим сразу, что не создание компьютера следует считать точкой в возникновении информатики. Компьютер сам по себе является продуктом длительной истории развития вычислительных устройств (мы рассмотрим ее ниже), и его можно рассматривать как некий инструмент, средство информатики. Появление компьютеров повлияло на темпы становления информатики, ее характер и содержание. Но связывать возникновение информатики только с появлением компьютеров или, наоборот, считать появление компьютера результатом информатики было бы неверным.

После Второй мировой войны возникла и начала бурно развиваться кибернетика как наука об общих закономерностях в управлении и связи в различных системах: искусственных, биологических, социальных. Рождение кибернетики принято связывать с опубликованием (1948 г.) американским математиком Норбертом Винером ставшей знаменитой книги «Кибернетика, или управление и связь в животном и машине». В этой работе были показаны пути создания общей теории управления и заложены основы методов рассмотрения проблем управления и связи для различных систем с единой точки зрения.

Развиваясь одновременно с развитием электронно-вычислительных машин, кибернетика со временем превратилась в более общую науку о преобразовании информации. Под информацией в кибернетике (рис. В.1) понимается любая совокупность сигналов, воздействий или сведений, которые некоторая система воспринимает от окружающей среды (входная информация X), выдает в окружающую среду (выходная информация Z), а также хранит в себе (внутренняя, внутрисистемная информация Y).

В нашей стране развитие кибернетики переживало драматические периоды. Как писал академик А. И. Берг, «...в 1955—1957 гг. и даже позже в нашей литературе

были допущены грубые ошибки в оценке значения и возможностей кибернетики. Это нанесло серьезный ущерб развитию науки в нашей стране, привело к задержке в разработке многих теоретических положений и даже самих электронных машин». Достаточно сказать, что еще в философском словаре 1959 г. издания кибернетика характеризовалась как «буржуазная лжененаука». Причиной этого послужили, с одной стороны, недо-

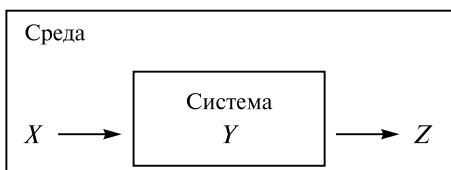


Рис. В.1. Общая схема обмена информацией между системой и внешней средой

оценка новой бурно развивающейся науки отдельными учеными «классического» направления, с другой — неумеренное пустословие тех, кто вместо активной разработки конкретных проблем кибернетики в различных областях спекулировал на полуфантастических прогнозах о безграничных возможностях кибернетики, дискредитируя тем самым эту науку.

Вскоре вслед за появлением термина «кибернетика» в мировой науке стало использоваться англоязычное «Computer Science», а чуть позже, на рубеже 1960-х и 1970-х гг., французы ввели получивший сейчас широкое распространение термин «Informatique». В русском языке наиболее раннее (примерно с середины 1960-х гг.) употребление термина «информатика» связано с узкой конкретной областью изучения структуры и общих свойств научной информации, передаваемой посредством научной (да и не только научной) литературы. Эта информационно-аналитическая деятельность, совершенно необходимая и сегодня в библиотечном деле, книгоиздании и т.д., уже давно не отражает современного понимания информатики.

Прикладные направления информатики — информационные технологии

Важной особенностью информатики является то, что она имеет широчайшие приложения, охватывающие почти все виды человеческой деятельности: производство, управление, науку, образование, проектные разработки, торговлю, финансовую сферу, медицину, криминалистику, охрану окружающей среды и др., а также быт. Главное значение здесь имеет совершенствование социального управления на основе новых информационных технологий.

Как наука информатика изучает общие закономерности, свойственные информационным процессам (в самом широком смысле этого понятия). Когда разрабатываются новые носители информации, каналы связи, приемы кодирования, визуального отображения информации и многое другое, то конкретная природа этой информации почти не имеет значение. Для разработчика системы управления базами данных (СУБД) важны общие принципы организации и эффективность поиска данных, а не то, какие конкретно данные будут затем заложены в базу многочисленными пользователями. Эти общие закономерности есть предмет информатики как науки.

Объектом приложений информатики являются самые различные науки и области практической деятельности, для которых она стала непрерывным источником самых современных технологий, называемых часто «новые информационные технологии» (НИТ). Многообразные информационные технологии, функционирующие в разных видах человеческой деятельности (управление производственным процессом, системы проектирования, финансовые операции, образование и т.п.), имея общие черты, в то же время существенно различаются между собой. Тем самым образуются и различные «предметные» информатики.

Назовем наиболее известные информационные технологии. Для обозначения ряда из них используются ставшие традиционными сокращения.

АСУ — автоматизированные системы управления — комплекс технических и программных средств, которые во взаимодействии с человеком организуют управление объектами в производстве или общественной сфере. Например, в образовании используются системы АСУ-вуз.

АСУ ТП — автоматизированные системы управления технологическими процессами. Например, такая система управляет работой станка с числовым программным управлением (ЧПУ), процессом запуска космического аппарата и т.д.

АСНИ — автоматизированная система научных исследований — программно-аппаратный комплекс, в котором научные приборы сопряжены с компьютером, вводят в него данные измерений автоматически, а компьютер производит обработку этих данных и представление их в наиболее удобной для исследователя форме.

АОС — автоматизированная обучающая система. Есть системы, помогающие учащимся осваивать новый материал, производящие контроль знаний, помогающие преподавателям готовить учебные материалы и т. д.

САПР — система автоматизированного проектирования — программно-аппаратный комплекс, который во взаимодействии с человеком (конструктором, инженером-проектировщиком, архитектором и т. д.) позволяет максимально эффективно проектировать механизмы, здания, узлы сложных агрегатов и др.

ГИС — геоинформационная система — совокупность особым образом организованной картографической информации и баз данных, хранящих сведения об объектах, отраженных на картах.

Не все такие технологии имеют общепринятые сокращенные названия. Это не значит, что они менее важны. Упомянем диагностические системы в медицине, системы организации продажи билетов, системы ведения бухгалтерско-финансовой деятельности, системы обеспечения редакционно-издательской деятельности — спектр информационных технологий чрезвычайно широк.

Структура современной информатики

В национальном докладе России на II Международном Конгрессе ЮНЕСКО «Образование и информатика» (Москва, 1996 г.) структура предметной области информатики была описана в форме таблицы, изображенной на рис. В.2.

Опишем коротко составные части «ядра» современной информатики. Каждая из этих частей может рассматриваться как относительно самостоятельная научная дисциплина; взаимоотношения между ними примерно такие же, как между алгеброй, геометрией и математическим анализом в классической математике — все они хоть и самостоятельные дисциплины, но, несомненно, части одной науки.

Фундаментальной основой информатики является, прежде всего, математика. Ее раздел — математическая логика — основа многих построений в информатике. В ряде случаев информатика опирается на лингвистику (например, конструирование языков общения пользователя с ЭВМ, создание методик распознания смысла речевых сообщений и т. д.), взаимодействует с философией (ряд проблем, связанных с фундаментальными свойствами информации) и социологией (проблемы социальной информатики), другими науками.

Теоретическая информатика использует математические методы для общего изучения процессов обработки информации и включает в себя теорию алгоритмов и теорию автоматов, теорию информации и теорию кодирования, теорию формальных языков и грамматик, исследование операций и другие разделы. Важная часть теоретической информатики — искусственный интеллект — область информатики, в которой решаются сложнейшие проблемы, находящиеся на пересечении с психологией, физиологией, лингвистикой и другими науками. Как научить компьютер мыслить подобно человеку? Основные направления разработок, относящихся к этой области, — моделирование рассуждений, компьютерная лингвистика, машинный перевод, создание экспертных систем, распознавание образов и другие. От успехов работ в области искусственного интеллекта зависит, в частности, решение такой важнейшей прикладной проблемы, как создание интеллектуальных интерфейсных систем взаимодействия человека с компьютером, благодаря

ФУНДАМЕНТАЛЬНЫЕ ОСНОВЫ ИНФОРМАТИКИ						
Теоретическая информатика		Информация как семантическое свойство материи. Информация и эволюция в живой и неживой природе. Начала общей теории информации. Методы измерения информации. Макро- и микронформация. Математические и информационные модели. Теория алгоритмов. Стохастические методы в информатике. Вычислительный эксперимент как методология научного исследования. Информация и знания. Семантические аспекты интеллектуальных процессов и информационных систем. Информационные системы искусственного интеллекта. Методы представления знаний. Познание и творчество как информационные процессы. Теория и методы разработки и проектирования информационных систем и технологий.				
Средства информатизации	Технические	Обработка, отображения и передачи данных	Системные	Персональные компьютеры. Рабочие станции. Устройства ввода/вывода и отображения информации. Аудио- и видеосистемы, системы мультимедиа. Сети ЭВМ. Средства связи и компьютерные телекоммуникационные системы. Операционные системы и среды. Системы и языки программирования. Сервисные оболочки, системы пользовательского интерфейса. Программные средства межкомпьютерной связи (системы теледоступа), вычислительные и информационные среды.		
Программные реализации технологий		Универсальных	Текстовые и графические редакторы. Системы управления базами данных. Процессоры электронных таблиц. Средства моделирования объектов, процессов, систем. Информационные языки и форматы представления данных и знаний; словари; классификаторы; тезауруссы. Средства защиты информации от разрушения и несанкционированного доступа. Издательские системы. Системы реализации технологий автоматизации расчетов, проектирования, обработки данных (учета, планирования, управления, анализа, статистики и т. д.). Системы искусственного интеллекта (базы знаний, экспертные системы, диагностические, обучающие и др.).			
Информационные технологии		Ввода/вывода, сбора, хранения, передачи и обработки данных. Подготовки текстовых и графических документов, технической документации. Интеграции и коллективного использования разнородных информационных ресурсов. Защита информации. Программирования, проектирования, моделирования, обучения, диагностики, управления (объектами, процессами, системами).				
Социальная информатика		Информационные ресурсы как фактор социально-экономического и культурного развития общества. Информационное общество — закономерности и проблемы становления и развития. Информационная инфраструктура общества. Проблемы информационной безопасности. Новые возможности развития личности в информационном обществе. Проблемы демократизации в информационном обществе и пути их решения. Информационная культура и информационная безопасность личности.				

Рис. В.2. Структура предметной области информатики

которым это взаимодействие будет походить на межчеловеческое и станет более эффективным.

Технические и программные средства информатизации — орудия реализации информатики на прикладном уровне. Сюда входят **вычислительная техника** и **техника, обеспечивающая телекоммуникации** — подраздел, в котором разрабатываются общие принципы построения вычислительных систем. Речь идет не о технических деталях и электронных схемах (это лежит за пределами информатики как таковой), а о принципиальных решениях на уровне так называемой архитектуры вычислительных (компьютерных) систем, определяющей состав, назначение, функциональные возможности и принципы взаимодействия устройств. Примеры принципиальных, ставших классическими решений в этой области — неймановская архитектура компьютеров первых поколений, шинная архитектура ЭВМ старших поколений, архитектура параллельной (многопроцессорной) обработки информации. **Программное обеспечение** — другой подраздел этого раздела информатики. В нем можно выделить несколько классов программных средств (см. рис. В.2). **Информационные системы** в рамках данной классификации также относятся к этому обширному разделу. Они связаны с решением вопросов по анализу потоков информации в различных системах, их оптимизации, структурировании, принципах хранения и поиска информации. Информационно-справочные системы, информационно-поисковые системы, глобальные системы хранения и поиска информации (включая Интернет) в последнее десятилетие XX в. привлекают внимание все большего круга пользователей. Без теоретического обоснования принципиальных решений в океане информации можно просто захлебнуться. Известным примером решения проблемы на глобальном уровне может служить гипертекстовая поисковая система WWW, а на значительно более низком уровне — справочная система, к услугам которой мы прибегаем, набрав телефонный номер 09.

Информационные технологии, отраженные в таблице (см. рис. В.2), являются достаточно универсальными. Действительно, практически каждому пользователю компьютеров приходится сталкиваться с вводом/выводом информации, подготовкой текстовых и графических документов, защитой информации (пусть на достаточно элементарном уровне). Более профессиональной деятельностью является интеграция и коллективное использование информационных ресурсов, но, строго говоря, все пользователи Интернета владеют элементами такой технологии. Некоторым проходится, в силу профессиональной необходимости, заниматься **программированием** — деятельностью, связанной с разработкой систем программного обеспечения. Основные разделы современного программирования таковы: создание системного программного обеспечения и создание прикладного программного обеспечения. Среди системного — разработка новых языков программирования и компиляторов к ним, разработка интерфейсных систем (пример — общеизвестная операционная оболочка и система Windows). Среди прикладного программного обеспечения общего назначения самые популярные — системы обработки текстов, электронные таблицы (табличные процессоры), системы управления базами данных. Наконец, в каждой области предметных приложений информатики существует множество специализированных прикладных программных систем более узкого назначения, примеры которых приведены выше. Относить ли соответствующие технологии собственно к информатике или считать их частью предметных областей — дело, скорее, договоренности.

Социальная информатика начала оформляться в самостоятельную область информатики сравнительно недавно, и этот процесс отнюдь не завершен. Социальные последствия информатизации всех сфер общественной жизни (экономики, поли-

тиki, культуры, образования и т.д.) столь велики и системны, что многие исследователи считают, что мир переходит в новую фазу развития — так называемое информационное общество.

Место информатики в системе наук

Рассмотрим место науки информатики в традиционно сложившейся системе наук (технических, естественных, гуманитарных и т.д.). В частности, это может помочь определить место образовательной области «информатика» в системе образования на различных ее ступенях.

Академик Б. Н. Наумов определял информатику «как естественную науку, изучающую общие свойства информации, процессы, методы и средства ее обработки (сбор, хранение, преобразование, перемещение, выдача)».

Как отмечал академик А. П. Ершов, в современных условиях термин «информатика» «вводится в русский язык в новом и куда более широком значении — как название фундаментальной естественной науки, изучающей процессы передачи и обработки информации. При таком толковании информатика оказывается более непосредственно связанной с философскими и общенаучными категориями, проясняется и ее место в кругу «традиционных» академических научных дисциплин».

Однако к началу XXI в. информатика все же еще «не состоялась» как фундаментальная наука. До настоящего времени не открыты научные законы, связанные с информацией, нет даже общепринятого определения базовых понятий информации и информационного процесса. Само слово «обработка...» в определении информатики говорит само за себя — это область технологическая, инженерная. Информатика остается областью общественной практики, все-таки сильно связанной с компьютерными системами. Ряд ученых подчеркивают, что информатика имеет характерные черты и других групп наук — технических и гуманитарных (или общественных).

Черты технической науки придают информатике ее аспекты, связанные с созданием и функционированием машинных систем обработки информации. Так, академик А. А. Дородницын определяет состав информатики как «три неразрывно и существенно связанные части: технические средства, программные и алгоритмические». Указанное обстоятельство нашло, в частности, свое отражение в первоначальном наименовании школьного предмета «Основы информатики и вычислительной техники», хотя в настоящее время более признанным становится название «Информатика» (но, по-прежнему, этот предмет включает в себя разделы, связанные с изучением технических, программных и алгоритмических средств). И, на-



Рис. В.3. К вопросу о месте информатики в системе наук

конец, науке информатике присущи некоторые черты гуманитарной (общественной) науки, что обусловлено ее вкладом в развитие и совершенствование социальной сферы.

Попытку определить, что же такое современная информатика, сделал в 1978 г. Международный конгресс по информатике. Цитируем: «Понятие информатики охватывает области, связанные с разработкой, созданием, использованием и материально-техническим обслуживанием систем обработки информации, включая машины, оборудование, математическое обеспечение, организационные аспекты, а также комплекс промышленного, коммерческого, административного и социального воздействия».

Таким образом, информатика является комплексной, междисциплинарной отраслью научного знания, как это изображено на рис. В.3.

Контрольные вопросы

1. Что общего и в чем различие информатики и кибернетики?
2. Какие определения информатики вы знаете?
3. Какова общая структура современной информатики?
4. Какие существуют наиболее известные информационные технологии?
5. Какое место занимает информатика в системе наук?

Глава 1

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ

Практически в каждой науке есть фундамент, без которого ее прикладные аспекты лишены основ. Для математики такой фундамент составляют теория множеств, теория чисел, математическая логика и некоторые другие разделы; для физики — это основные законы классической и квантовой механики, статистической физики, релятивистской теории; для химии — периодический закон, его теоретические основы и т.д.

Конечно можно научиться считать и пользоваться калькулятором, даже не подозревая о существовании указанных выше разделов математики, делать химические опыты без понимания существа химических законов, но при этом не следует думать, что знаешь математику или химию. Примерно то же с информатикой: можно изучить несколько популярных программ и даже приобрести некоторые умения, но это отнюдь не вся информатика, точнее, даже не самая главная и интересная ее часть.

Теоретические основы информатики — пока не вполне сложившийся, еще не устоявшийся раздел науки. Он возникает на наших глазах, что делает его особенно интересным — нечасто мы наблюдаем и даже можем участвовать в рождении новой науки! Как и теоретические разделы других наук, теоретическая информатика формируется в значительной мере под влиянием потребностей обучения информатике.

Теоретическая информатика — наука математизированная. Она складывается из ряда разделов математики, которые прежде казались мало связанными друг с другом: теории автоматов и теории алгоритмов, математической логики, теории формальных языков и грамматик, реляционной алгебры, теории информации и др.

Теоретическая информатика старается методами точного анализа ответить на основные вопросы, возникающие при работе с информацией, например на вопрос о количестве информации, сосредоточенной в той или иной информационной системе, наиболее рациональной организации таких систем для хранения и поиска информации, а также о существовании и свойствах алгоритмов преобразования информации.

Конструкторы устройств хранения данных проявляют чудеса изобретательности, увеличивая объем и плотность хранения данных на дисках, но в основе этой деятельности лежат теория информации и теория кодирования.

Для решения прикладных задач существуют замечательные программы, но для того чтобы грамотно поставить прикладную задачу, привести ее к виду, который подвластен компьютеру, надо знать основы информационного и математического моделирования и т.д. Только освоив эти разделы информатики, можно считать себя специалистом в этой науке.

Другое дело — с какой глубиной осваивать; многие разделы теоретической информатики достаточно сложны и требуют основательной математической подготовки. В ряде случаев с ними можно только ознакомиться с целью составить о них отчетливое представление.

1.1. Информация, ее виды и свойства

1.1.1. Различные уровни представлений об информации

Понятие «**информация**» является одним из фундаментальных в современной науке вообще и базовым для изучаемой нами информатики. Информацию наряду с веществом и энергией рассматривают в качестве важнейшей сущности мира, в котором мы живем. Однако если задаться целью формально определить понятие «**информация**», то сделать это будет чрезвычайно сложно. Аналогичными «неопределяемыми» понятиями, например в математике, являются точка или прямая. Так, можно сделать некоторые утверждения, связанные с этими математическими понятиями, но определить их с помощью более элементарных понятий нельзя.

В простейшем бытовом понимании с термином «**информация**» обычно ассоциируются некоторые сведения, данные, знания и т. п. Информация передается в виде **сообщений**, определяющих ее форму и представление. Примерами сообщений являются: музыкальное произведение; телепередача; команды регулировщика на перекрестке; текст, распечатанный на принтере; данные, полученные в результате работы составленной вами программы, и т. д. При этом предполагается, что имеются **источник информации** и **получатель информации**.

Сообщение от источника к получателю передается посредством какой-нибудь среды, являющейся **каналом связи** (рис. 1.1). Так, при передаче речевого сообщения в качестве такого канала связи можно рассматривать воздух, в котором распространяются звуковые волны, а в случае передачи письменного сообщения (например текста, распечатанного на принтере) каналом сообщения можно считать лист бумаги, на котором напечатан текст.

Человеку свойственно **субъективное** восприятие информации через некоторый набор ее свойств: важность, достоверность, своевременность, доступность и т. д. В этом смысле одно и то же сообщение, передаваемое от источника к получателю, может передавать информацию в разной степени. Так, например, вы хотите сообщить о неисправности компьютера. Для инженера из группы технического обслуживания сообщение «компьютер сломался» явно содержит больше информации, чем для вахтера. Но, в свою очередь, для инженера сообщение «не включается дисплей» содержит информации больше, чем первое, поскольку в большей степени снимает неопределенность, связанную с причиной неисправности компьютера.

Использование терминов «больше информации» или «меньше информации» подразумевает некую возможность ее **измерения** (или хотя бы количественного соотнесения). При субъективном восприятии измерение информации возможно лишь в виде установления некоторой порядковой шкалы для оценки «больше» — «меньше», да и то субъективной, поскольку на свете немало людей, для которых, например, оба сообщения, использованных ранее в качестве примера, вообще не несут никакой информации. Это становится невозможным при введении **объективных** характеристик, важнейшей из которых является количество. Однако при объективном измерении количества информации следует заведомо отрешиться от восприятия ее с точки зрения субъективных свойств, примеры которых были перечислены. Более того, не исключено, что не всякая информация будет иметь объек-



Рис. 1.1. Схема передачи информации

тивно измеряемое количество — все зависит от того, как будут введены единицы измерения. Не исключено и то, что при разных способах введения единиц измерения информация, содержащаяся в двух допускающих измерение сообщениях, будет по-разному соотноситься.

1.1.2. Непрерывная и дискретная информация

Чтобы сообщение было передано от источника к получателю, необходима некоторая материальная субстанция — **носитель** информации. Сообщение, передаваемое с помощью носителя, назовем сигналом. В общем случае **сигнал** — это изменяющийся во времени физический процесс. Такой процесс может содержать различные характеристики (например, при передаче электрических сигналов могут изменяться напряжение и сила тока). Та из характеристик, которая используется для представления сообщений, называется **параметром сигнала**.

В случае, когда параметр сигнала принимает последовательное во времени конечное число значений (при этом все они могут быть пронумерованы), сигнал называется **дискретным**, а сообщение, передаваемое с помощью таких сигналов — **дискретным сообщением**. Информация, передаваемая источником, в этом случае также называется дискретной. Если же источник вырабатывает непрерывное сообщение (соответственно параметр сигнала — непрерывная функция от времени), соответствующая информация называется **непрерывной**. Пример дискретного сообщения — процесс чтения книги, информация в которой представлена текстом, т. е. дискретной последовательностью отдельных значков (букв). Примером непрерывного сообщения служит человеческая речь, передаваемая модулированной звуковой волной; параметром сигнала в этом случае является давление, создаваемое этой волной в точке нахождения приемника — человеческого уха.

Непрерывное сообщение может быть представлено непрерывной функцией $f(t)$, заданной на некотором отрезке $[a, b]$ (рис. 1.2, *a*). Непрерывное сообщение можно

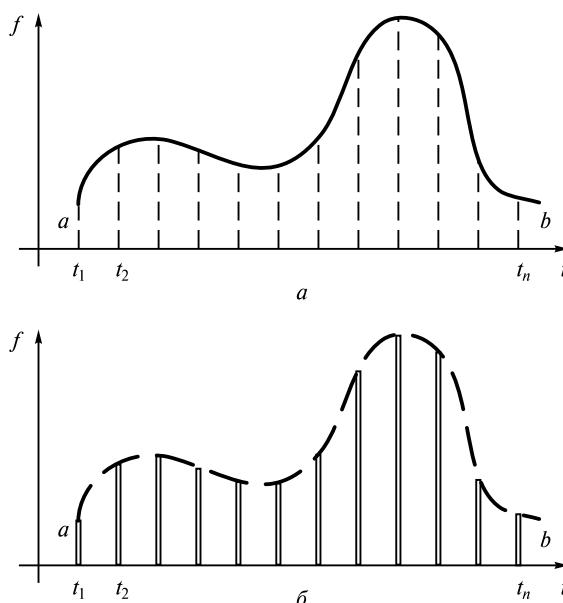


Рис. 1.2. Процедура дискретизации непрерывного сообщения

преобразовать в дискретное (такая процедура называется **дискретизацией**). Для этого из бесконечного множества значений этой функции (параметра сигнала) выбирается их определенное число, которое приближенно может характеризовать остальные значения. Один из способов такого выбора состоит в следующем. Область определения функции разбивается точками t_1, t_2, \dots, t_n на промежутки равной длины Δt (равномерная дискретизация), и непрерывная функция заменяется импульсной с некоторой короткой и одинаковой продолжительностью импульсов, амплитуда каждого из которых принимается постоянной и равной значению функции в указанных точках (рис. 1.2, б). Полученная таким образом импульсная функция является дискретным представлением непрерывной функции, точность которого можно неограниченно улучшать (по крайней мере теоретически) путем уменьшения длин отрезков разбиения области значений аргумента.

Таким образом, любое сообщение может быть представлено как дискретное, иначе говоря, последовательностью знаков некоторого алфавита.

Возможность дискретизации непрерывного сигнала с любой желаемой точностью (для возрастания точности достаточно уменьшить шаг) принципиально важна с точки зрения информатики. Компьютер — цифровая машина, т. е. внутреннее представление информации в нем дискретно. Дискретизация входной информации (если она непрерывна) позволяет сделать ее пригодной для компьютерной обработки.

Вопрос о способах дискретизации непрерывно представленной информации и восстановления такой информации по ее дискретному представлению (так называемым «отсчетам») возник в технике электросвязи существенно раньше создания ЭВМ (в связи с разработкой устройств цифровой временной коммутации для телефонных станций). Теоретическую базу для его решения (в случае равномерной дискретизации) заложил в 1933 г. советский ученый В. А. Котельников, сформулировавший ставшую впоследствии классической теорему, носящую его имя.

Теорема Котельникова: если непрерывный сигнал имеет спектр, ограниченный сверху частотой f_{\max} , то этот сигнал полностью определяется последовательностью своих значений в моменты времени, отстоящие друг от друга на интервал

$$\Delta t = \frac{1}{2f_{\max}}.$$

Например, сигнал звукового сопровождения в телевизионном канале ограничен сверху частотой 12 кГц. Если выбрать интервал дискретизации для этого сигнала не более чем $\Delta t = \frac{1}{2 \cdot 12 \cdot 10^3} \approx 4,2 \cdot 10^{-5}$ с, то непрерывный сигнал может (теоретически) быть восстановлен после дискретизации точно. Примечание «теоретически» означает, что конкретные технические устройства могут иметь дополнительные ограничения, которые здесь обсуждаться не будут. Кроме того, реальные непрерывные сигналы, подлежащие дискретизации и последующему восстановлению, имеют, как правило, неограниченные по частоте спектры, хотя высокочастотная составляющая быстро стремится к нулю с ростом частоты. Такие сигналы могут быть восстановлены по своим дискретным отсчетам лишь приближенно. При дискретизации в таких случаях вводят некоторую граничную частоту, за пределами которой высокочастотными составляющими просто пренебрегают. Так, при передаче телефонного сигнала такую частоту часто берут равной 3,4 кГц, что соответствует интервалу дискретизации $\Delta t = \frac{1}{2 \cdot 3,4 \cdot 10^3} \approx 1,5 \cdot 10^{-4}$ с.

Существенно, что теорема Котельникова не просто «теорема о возможности», но носит конструктивный характер. Восстановление непрерывного сигнала по его дискретному образу реализуется путем суммирования ряда

$$y(t) = \frac{1}{4\pi f_{\max}} \sum_{n=-\infty}^{\infty} y(n\Delta t) \frac{\sin(2\pi f_{\max}(1 - n\Delta t))}{2\pi f_{\max}(1 - n\Delta t)}.$$

В завершение отметим, что существуют и принципиально другие, нецифровые, вычислительные машины — аналоговые ЭВМ. Они используются обычно для решения задач специального характера и широкой публике практически неизвестны. Эти ЭВМ в принципе не нуждаются в дискретизации входной информации, так как ее внутреннее представление у них непрерывно. В этом случае все наоборот — если внешняя информация дискретна, то ее «перед употреблением» необходимо преобразовать в непрерывную.

1.1.3. Единицы количества информации: вероятностный и объемный подходы

Определить понятие «количество информации» довольно сложно. В решении этой проблемы существуют два основных подхода. Исторически они возникли почти одновременно. В конце 1940 гг. один из основоположников кибернетики американский математик Клод Шеннон развел вероятностный подход к измерению количества информации, а работы по созданию ЭВМ привели к «объемному» подходу.

Вероятностный подход

Рассмотрим в качестве примера опыт, связанный с бросанием правильной игральной кости, имеющей N граней (наиболее распространенным является случай шестигранной кости: $N = 6$). Результатом данного опыта может быть выпадение грани с одним из следующих знаков: 1, 2, ..., N .

Введем в рассмотрение численную величину, измеряющую неопределенность — **энтропию** (обозначим ее H). Величины N и H связаны между собой некоторой функциональной зависимостью:

$$H = f(N), \quad (1.1)$$

а сама функция f является возрастающей, неотрицательной и определенной (в рассматриваемом нами примере) для $N = 1, 2, \dots, 6$.

Рассмотрим процедуру бросания кости более подробно:

1) готовимся бросить кость; исход опыта неизвестен, т.е. имеется некоторая неопределенность; обозначим ее $H1$;

2) кость брошена; информация об исходе данного опыта получена; обозначим количество этой информации через I ;

3) обозначим неопределенность данного опыта после его осуществления через $H2$.

За количество информации, которое получено в ходе осуществления опыта, примем разность неопределенностей «до» и «после» опыта:

$$I = H1 - H2.$$

Очевидно, что в случае, когда получен конкретный результат, имевшаяся неопределенность снята ($H2 = 0$), и, таким образом, количество полученной информации совпадает с первоначальной энтропией. Иначе говоря, неопределенность, заключенная в опыте, совпадает с информацией об исходе этого опыта.

Следующим важным моментом является определение вида функции f в формуле (1.1). Если варьировать число граней N и число бросаний кости (обозначим эту величину через M), то общее число исходов (векторов длины M , состоящих из знаков 1, 2, ..., N) будет равно N в степени M :

$$X = N^M. \quad (1.2)$$

Так, в случае двух бросаний кости с шестью гранями имеем $X = 6^2 = 36$. Фактически каждый исход X есть некоторая пара (X_1, X_2) , где X_1 и X_2 — соответственно исходы первого и второго бросаний (X — общее число таких пар).

Ситуацию с бросанием кости M раз можно рассматривать как некую сложную систему, состоящую из независимых друг от друга подсистем — «однократных бросаний кости». Энтропия такой системы в M раз больше, чем энтропия одной системы (так называемый «принцип аддитивности энтропии»):

$$f(6^M) = Mf(6).$$

Данную формулу можно распространить и на случай любого N :

$$f(N^M) = Mf(N). \quad (1.3)$$

Прологарифмируем левую и правую части формулы (1.2): $\ln X = M \ln N$, $M = \ln X / \ln N$.

Подставляем полученное для M значение в формулу (1.3):

$$f(X) = \frac{\ln X}{\ln N} f(N).$$

Обозначив через K положительную константу, получим $f(X) = K \ln X$, или, с учетом (1.1), $H = K \ln N$. Обычно принимают $K = 1/\ln 2$. Таким образом,

$$H = \log_2 N. \quad (1.4)$$

Выражение (1.4) — формула Хартли.

Важным при введении какой-либо величины является вопрос о том, что принимать за единицу ее измерения. Очевидно, H будет равно единице при $N = 2$. Иначе говоря, в качестве единицы принимается количество информации, связанное с проведением опыта, состоящего в получении одного из двух равновероятных исходов (примером такого опыта может служить бросание монеты, при котором возможны два исхода: «орел» и «решка»). Такая единица количества информации называется «бит».

Все N исходов рассмотренного выше опыта являются равновероятными и поэтому можно считать, что на «долю» каждого исхода приходится одна N -я часть общей неопределенности опыта: $(\log_2 N)/N$. При этом вероятность i -го исхода P_i равняется, очевидно, $1/N$.

Таким образом,

$$H = \sum_{i=1}^N P_i \log_2 \left(\frac{1}{P_i} \right). \quad (1.5)$$

Формула (1.5) принимается за меру энтропии и в случае, когда вероятности различных исходов опыта **неравновероятны** (т. е. P_i могут быть различны). Формула (1.5) называется **формулой Шеннона**.

В качестве примера определим количество информации, связанное с появлением каждого символа в сообщениях, записанных на русском языке. Будем считать, что русский алфавит состоит из 33 букв и знака «пробел» для разделения слов. По формуле (1.4)

Частотность букв русского языка

i	Символ	$P(i)$	i	Символ	$P(i)$	i	Символ	$P(i)$
1	Пробел	0,175	13	К	0,028	25	Ч	0,012
2	О	0,090	14	М	0,026	26	Й	0,010
3	Е	0,072	15	Д	0,025	27	Х	0,009
4	Ё	0,072	16	П	0,023	28	Ж	0,007
5	А	0,062	17	У	0,021	29	Ю	0,006
6	И	0,062	18	Я	0,018	30	Ш	0,006
7	Т	0,053	19	Ы	0,016	31	Ц	0,004
8	Н	0,053	20	З	0,016	32	Щ	0,003
9	С	0,045	21	Ь	0,014	33	Э	0,003
10	Р	0,040	22	Ъ	0,014	34	Ф	0,002
11	В	0,038	23	Б	0,014			
12	Л	0,035	24	Г	0,012			

$$H = \log_2 34 \approx 5 \text{ бит.}$$

Однако в словах русского языка (равно как и в словах других языков) различные буквы встречаются неодинаково часто. В табл. 1.1 приведены вероятности частоты употребления различных знаков русского алфавита, полученные на основе анализа очень больших по объему текстов.

Воспользуемся для подсчета H формулой (1.5): $H \approx 4,72$ бит. Полученное значение H , как и можно было предположить, меньше вычисленного ранее. Величина H , вычисляемая по формуле (1.4), является максимальным количеством информации, которое могло бы приходиться на один знак.

Аналогичные подсчеты H можно провести и для других языков, например использующих латинский алфавит, — английского, немецкого, французского и др. (26 различных букв и «пробел»). По формуле (1.4) получим

$$H = \log_2 27 \approx 4,76 \text{ бит.}$$

Как и в случае русского языка, частота появления тех или иных знаков неодинакова. Если расположить все буквы данных языков в порядке убывания вероятностей, то получим такие последовательности:

английский язык: «пробел», Е, Т, А, О, Н, Р, ...

немецкий язык: «пробел», Е, Н, И, С, Т, Р, ...

французский язык: «пробел», Е, С, А, Н, И, Т, ...

Рассмотрим алфавит, состоящий из двух знаков: 0 и 1. Если считать, что со знаками 0 и 1 в двоичном алфавите связаны одинаковые вероятности их появления ($P(0) = P(1) = 0,5$), то количество информации на один знак при двоичном кодировании

$$H = \log_2 2 = 1 \text{ бит.}$$

Таким образом, количество информации (в битах), заключенное в двоичном слове, равно числу двоичных знаков в нем.

Объемный подход

В двоичной системе счисления знаки 0 и 1 будем называть **битами** (от английского выражения *Binary digits* — двоичные цифры). Отметим, что создатели компьютеров отдают предпочтение именно двоичной системе счисления потому, что в техническом устройстве наиболее просто реализовать два противоположных физических состояния. В компьютере бит является наименьшей возможной единицей информации.

Объем информации, записанной двоичными знаками в памяти компьютера или на внешнем носителе информации, подсчитывается просто по количеству требуемых для такой записи двоичных символов. При этом, в частности, невозможно нецелое число битов (в отличие от вероятностного подхода).

Для удобства использования введены и более крупные, чем бит, единицы количества информации. Так, двоичное слово из восьми знаков содержит один **байт** информации, 1024 байта образуют **килобайт** (Кбайт), 1024 килобайта — **мегабайт** (Мбайт), а 1024 мегабайта — **гигабайт** (Гбайт).

Между вероятностным и объемным количеством информации соотношение неоднозначное. Далеко не всякий текст, записанный двоичными символами, допускает измерение объема информации в кибернетическом смысле, но заведомо допускает его измерение в объемном смысле. Далее, если некоторое сообщение допускает измеримость количества информации в обоих смыслах, то они не обязательно совпадают, при этом кибернетическое количество информации не может быть больше объемного.

В дальнейшем тексте данного учебника практически всегда количество информации понимается в объемном смысле.

1.1.4. Информация: более широкий взгляд

Как ни важно измерение информации, нельзя сводить к нему все связанные с этим понятием проблемы. При анализе информации социального (широким смысле) происхождения на первый план могут выступить такие ее свойства, как истинность, своевременность, ценность, полнота и т. д. Их невозможно оценить ни уменьшением неопределенности (вероятностный подход), ни числом символов (объемный подход). Обращение к качественной стороне информации породило иные подходы к ее оценке. При **аксиологическом** подходе стремятся исходить из ценности, практической значимости информации, т. е. качественных характеристик, значимых в социальной системе. При **семантическом** подходе информация рассматривается с точки зрения как формы, так и содержания. При этом информацию связывают с **тезаурусом**, т. е. полнотой систематизированного набора данных о предмете информации. Отметим, что эти подходы не исключают количественного анализа, но он становится существенно сложнее и должен базироваться на современных методах математической статистики.

Понятие информации нельзя считать лишь техническим, междисциплинарным и даже наддисциплинарным термином. Информация — это фундаментальная философская категория. Дискуссии ученых о философских аспектах информации надежно показали несводимость информации ни к одной из этих категорий. Концепции и толкования, возникающие на пути догматических подходов, оказываются слишком частными, односторонними, не охватывающими всего объема этого понятия.

Попытки рассмотреть категорию информации с позиций основного вопроса философии привели к возникновению двух противостоящих концепций — так

называемых **атрибутивной** и **функциональной**. «Атрибутисты» квалифицируют информацию как свойство всех материальных объектов, т. е. как атрибут материи. «Функционалисты» связывают информацию лишь с функционированием сложных, самоорганизующихся систем. Оба подхода, скорее всего, неполны. Дело в том, что природа сознания по сути своей является информационной, т. е. сознание — менее общее понятие по отношению к категории «информация». Нельзя признать корректными попытки сведения более общего понятия к менее общему. Таким образом, информация и информационные процессы, если иметь в виду решение основного вопроса философии, опосредуют материальное и духовное, т. е. вместо классической постановки этого вопроса получается два новых: о соотношении материи и информации и о соотношении информации и сознания (духа).

Можно попытаться дать философское определение информации с помощью указания на связь определяемого понятия с категориями **отражения** и **активности**. Информация есть содержание образа, формируемого в процессе отражения. Активность входит в это определение в виде представления о формировании некоторого образа в процессе отражения некоторого субъект-объектного отношения. При этом не требуется указания на связь информации с материей, поскольку как субъект, так и объект процесса отражения могут принадлежать как к материальной, так и к духовной сфере социальной жизни. Однако существенно подчеркнуть, что материалистическое решение основного вопроса философии требует признания необходимости существования материальной среды — носителя информации в процессе такого отражения. Итак, информацию следует трактовать как имманентный (неотъемлемо присущий) атрибут материи, необходимый момент ее самодвижения и саморазвития. Эта категория приобретает особое значение применительно к высшим формам движения материи — биологической и социальной.

Данное выше определение охватывает важнейшие характеристики информации. Оно не противоречит тем знаниям, которые накоплены по этой проблематике, а наоборот, является выражением наиболее значимых.

Современная практика психологии, социологии, информатики диктует необходимость перехода к информационной трактовке сознания. Такая трактовка оказывается чрезвычайно плодотворной и позволяет, например, рассмотреть с общих позиций индивидуальное и общественное сознание. Генетически индивидуальное и общественное сознание неразрывны и в то же время общественное сознание не есть простая сумма индивидуальных, поскольку оно включает информационные потоки и процессы между индивидуальными сознаниями.

В социальном плане человеческая деятельность предстает как взаимодействие реальных человеческих коммуникаций с предметами материального мира. Поступившая извне к человеку информация является отпечатком, снимком сущностных сил природы или другого человека. Таким образом, с единых методологических позиций может быть рассмотрена деятельность индивидуального и общественного сознания, экономическая, политическая, образовательная деятельность различных субъектов социальной системы.

Определение информации как философской категории не только затрагивает физические аспекты существования информации, но и фиксирует ее социальную значимость.

Одной из важнейших черт функционирования современного общества выступает его информационная оснащенность. В ходе своего развития человеческое общество прошло через пять информационных революций. Первая из них была связана с появлением языка, вторая — письменности, третья — книгопечатания, четвертая — телесвязи, и, наконец, пятая — компьютеров (а также магнитных и

оптических носителей хранения информации). Каждый раз новые информационные технологии поднимали информированность общества на несколько порядков, радикально меняя объем и глубину знания, а вместе с этим и уровень культуры в целом.

Одна из целей философского анализа понятия информации — указать место информационных технологий в развитии форм движения материи, в прогрессе человечества и, в том числе, в развитии разума как высшей отражательной способности материи. На протяжении десятков тысяч лет сфера разума развивалась исключительно через общественную форму сознания. С появлением компьютеров начались разработки систем искусственного интеллекта, идущих по пути моделирования общих интеллектуальных функций индивидуального сознания.

1.1.5. Информация и физический мир

Известно большое количество работ, посвященных физической трактовке информации. Эти работы в значительной мере построены на основе аналогии формулы Больцмана, описывающей энтропию статистической системы материальных частиц, и формулы Хартли.

Заметим, что при всех выводах формулы Больцмана явно или неявно предполагается, что макроскопическое состояние системы, к которому относится функция энтропии, реализуется на микроскопическом уровне как сочетание механических состояний очень большого числа частиц (молекул), образующих систему. В задачах кодирования и передачи информации, для решения которых Хартли и Шенноном была развита вероятностная мера информации, использовалось очень узкое техническое понимание информации, почти не имеющее отношения к полному объему этого понятия. Таким образом, большинство рассуждений, использующих термодинамические свойства энтропии применительно к информации нашей реальности, носят спекулятивный характер. В частности, являются необоснованными использование понятия «энтропия» для систем с конечным и небольшим числом состояний, а также попытки расширительного методологического толкования результатов теории вне довольно примитивных механических моделей, для которых они были получены.

Информацию следует считать особым видом ресурса, при этом имеется в виду толкование «ресурса» как запаса неких знаний материальных предметов или энергетических, структурных или каких-либо других характеристик предмета. В отличие от ресурсов, связанных с материальными предметами, информационные ресурсы являются неистощимыми и предполагают существенно иные методы воспроизведения и обновления, чем материальные ресурсы.

Рассмотрим некоторый набор свойств информации:

- запоминаемость;
- передаваемость;
- воспроизводимость;
- преобразуемость;
- стираемость.

Запоминаемость — одно из самых важных свойств. Запоминаемую информацию будем называть макроскопической (имея в виду пространственные масштабы запоминающей ячейки и время запоминания). Именно с макроскопической информацией мы имеем дело в реальной практике.

Передаваемость информации с помощью каналов связи (в том числе с помехами) хорошо исследована в рамках теории информации К. Шеннона. В данном случае имеется в виду несколько иной аспект — способность информации к копиро-

ванию, т.е. к тому, что она может быть «запомнена» другой макроскопической системой и при этом останется тождественной самой себе. Очевидно, что количество информации не должно возрастать при копировании.

Воспроизведимость информации тесно связана с ее передаваемостью и не является ее независимым базовым свойством. Если передаваемость означает, что не следует считать существенными пространственные отношения между частями системы, между которыми передается информация, то воспроизведимость характеризует неиссякаемость и неистощимость информации, т.е. что при копировании информация остается тождественной самой себе.

Фундаментальное свойство информации — **преобразуемость**. Оно означает, что информация может менять способ и форму своего существования. Копируемость есть разновидность преобразования информации, при котором ее количество не меняется. В общем случае количество информации в процессах преобразования меняется, но возрастать не может.

Свойство **стираемости** информации также не является независимым. Оно связано с таким преобразованием информации (передачей), при котором ее количество уменьшается и становится равным нулю.

Данных свойств информации недостаточно для формирования ее меры, так как они относятся к физическому уровню информационных процессов.

Подводя итог сказанному, отметим, что предпринимаются (но отнюдь не завершены) усилия ученых, представляющих самые разные области знания, построить единую теорию, которая призвана формализовать понятие информации и информационного процесса, описать превращения информации в процессах самой разной природы. Движение информации есть сущность процессов управления, которые суть проявление имманентной активности материи, ее способности к самодвижению. С момента возникновения кибернетики управление рассматривается применительно ко всем формам движения материи, а не только к высшим (биологической и социальной). Многие проявления движения в неживых — искусственных (технических) и естественных (природных) — системах также обладают общими признаками управления, хотя их исследуют в химии, физике, механике в энергетической, а не в информационной системе представлений. Информационные аспекты в таких системах составляют предмет новой междисциплинарной науки — синергетики.

Высшей формой информации, проявляющейся в управлении в социальных системах, являются знания. Это наддисциплинарное понятие, широко используемое в педагогике и исследованиях по искусственноному интеллекту, также претендует на роль важнейшей философской категории. В философском плане познание следует рассматривать как один из функциональных аспектов управления. Такой подход открывает путь к системному пониманию генезиса процессов познания, его основ и перспектив.

Контрольные вопросы

1. Какая форма представления информации — непрерывная или дискретная — приемлема для компьютеров и почему?
2. В чем состоит процедура дискретизации непрерывной информации?
3. Как определяется понятие энтропии?
4. Каким образом определяется единица количества информации при кибернетическом подходе?
5. Каковы особенности определения количества информации, связанной с появлением различных знаков в сообщениях?
6. Перечислите свойства информации.

1.2. Кодирование информации

1.2.1. Абстрактный алфавит

Информация передается в виде сообщений. Дискретная информация записывается с помощью некоторого конечного набора знаков, которые будем называть **буквами**, не вкладывая в это слово привычного ограниченного значения (типа «русские буквы» или «латинские буквы»). Буква в данном расширенном понимании — любой из знаков, которые некоторым соглашением установлены для общения. Например, при привычной передаче сообщений на русском языке такими знаками будут русские буквы — прописные и строчные, знаки препинания, пробел; если в тексте есть числа — то и цифры. Вообще буквой будем называть элемент некоторого конечного множества (набора) отличных друг от друга знаков. Множество знаков, в котором определен их порядок, назовем алфавитом (общезвестен порядок знаков в русском алфавите: А, Б, ..., Я).

Рассмотрим некоторые примеры алфавитов.

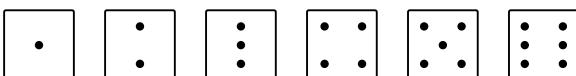
1. Алфавит прописных русских букв:

А Б В Г Д Е Ё Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я

2. Алфавит клавиатурных символов ПЭВМ IBM (русифицированная клавиатура):

'	1	2	3	4	5	6	7	8	9	0	-	=
~	!	@	#	\$	%	^	&	*	()	_	+	
q	w	e	r	t	y	u	i	o	p	[]		
Q	W	E	R	T	Y	U	I	O	P	{ }		
a	s	d	f	g	h	j	k	l	;	:	"	
A	S	D	F	G	H	J	K	L	:	«»		
z	x	c	v	b	n	m	,	.	/			
Z	X	C	V	B	N	M	<	>	?			
й	ц	у	к	е	н	г	ш	щ	з	х	ъ	
Й	Ц	У	К	Е	Н	Г	Ш	Щ	З	Х	Ъ	
ф	ы	в	а	п	р	о	л	д	ж	э		
Ф	Ы	В	А	П	Р	О	Л	Д	Ж	Э		
я	ч	с	м	и	т	ь	б	ю				
Я	Ч	С	М	И	Т	Ь	Б	Ю				

3. Алфавит знаков правильной шестигранной игральной кости:



4. Алфавит арабских цифр:

0 1 2 3 4 5 6 7 8 9

5. Алфавит шестнадцатеричных цифр:

0 1 2 3 4 5 6 7 8 9 A B C D E F

Этот пример, в частности, показывает, что знаки одного алфавита могут образовываться из знаков других алфавитов.

6. Алфавит двоичных цифр:

0 1

Алфавит 6 является одним из примеров так называемых «двоичных» алфавитов, т.е. алфавитов, состоящих из любых двух знаков. Другими примерами являются двоичные алфавиты: «точка», «тире» (· —); «плюс», «минус» (+ —).

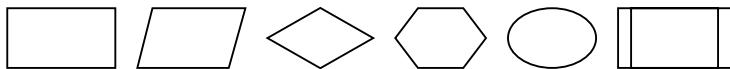
7. Алфавит прописных латинских букв:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

8. Алфавит римской системы счисления:

I V X L C D M

9. Алфавит языка блок-схем изображения алгоритмов:



10. Алфавит языка программирования Паскаль (см. гл. 3).

1.2.2. Кодирование и декодирование

В канале связи сообщение, составленное из символов (букв) одного алфавита, может преобразовываться в сообщение из символов (букв) другого алфавита. Правило, описывающее однозначное соответствие букв алфавитов при таком преобразовании, называют **кодом**. Саму процедуру преобразования сообщения называют перекодировкой. Подобное преобразование сообщения может осуществляться в момент поступления сообщения от источника в канал связи (**кодирование**) и в момент приема сообщения получателем (**декодирование**). Устройства, обеспечивающие кодирование и декодирование, будем называть соответственно **кодировщиком** и **декодировщиком**. На рис. 1.3 приведена схема, иллюстрирующая процесс передачи сообщения в случае перекодировки, а также воздействия помех.

Рассмотрим некоторые примеры кодов.

1. Азбука Морзе в русском варианте (алфавиту, составленному из алфавита русских заглавных букв и алфавита арабских цифр, ставится в соответствие алфавит Морзе):

А .—	К —.—	Ф .—..	0 — — — —
Б —... .	Л .—..	Х	1 .— — —
В .—. .	М ——	Ц —.—.	2 .. — —
Г —— .	Н —.	Ч — — — .	3 ... —
Д —.. .	О — — —	Ш — — — —	4
Е .	П .— — .	Щ — — .—	5
Ж ...— .	Р .—.	Ъ, Ъ —.—	6 —....
З .— ..	С ...	Ы — .— —	7 — — ...
И ..	Т —	Э ..— ..	8 — — — ..
Й .— — —	У ...	Ю ..— —	9 — — — — .
		Я .—.—	

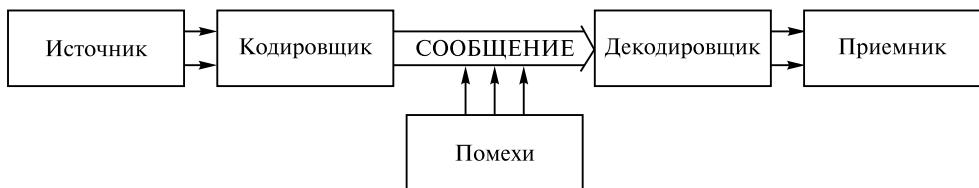


Рис. 1.3. Процесс передачи сообщения от источника к приемнику

2. Код Трисиме (знакам латинского алфавита ставятся в соответствие комбинации из трех знаков: 1, 2, 3):

A 111	D 121	G 131	J 211	M 221	P 231	S 311	V 321	Y 331
B 112	E 122	H 132	K 212	N 222	Q 232	T 312	W 322	Z 332
C 113	F 123	I 133	L 213	O 223	R 233	U 313	X 323	.

Код Трисиме является примером так называемого равномерного кода (такого, в котором все кодовые комбинации содержат одинаковое число знаков — в данном случае три). Пример неравномерного кода — азбука Морзе.

Кодирование чисел знаками различных систем счисления будет рассмотрено далее.

1.2.3. Кодирование и декодирование числовой информации, системы счисления

Общие понятия о позиционных и непозиционных системах счисления

Система счисления — принятый способ записи чисел и сопоставления этим записям реальных значений. Все системы счисления можно разделить на два класса: **позиционные** и **непозиционные**. Для записи чисел в различных системах счисления используется некоторое количество отличных друг от друга знаков. Число таких знаков в позиционной системе счисления называется **основанием системы счисления**. В табл. 1.2 приведены наименования некоторых позиционных систем счисления и перечень знаков (цифр), из которых в них образуются числа.

В позиционной системе счисления число может быть представлено в виде суммы произведений коэффициентов на степени основания системы счисления:

$$A_n A_{n-1} A_{n-2} \dots A_1 A_0, A_{-1} A_{-2} \dots = \\ = A_n * B^n + A_{n-1} * B^{n-1} + \dots + A_1 * B^1 + A_0 * B^0 + A_{-1} * B^{-1} + A_{-2} * B^{-2} + \dots$$

Знак «запятая» отделяет целую часть числа от дробной; знак «звездочка» здесь и далее используется для обозначения операции умножения; B — основание системы счисления. Таким образом, значение каждого знака в числе зависит от позиций, которую занимает знак в записи числа. Именно поэтому такие системы счисления называют позиционными.

Таблица 1.2

Некоторые системы счисления

Основание	Система счисления	Знаки
2	Двоичная	0, 1
3	Троичная	0, 1, 2
4	Четверичная	0, 1, 2, 3
5	Пятеричная	0, 1, 2, 3, 4
8	Восьмеричная	0, 1, 2, 3, 4, 5, 6, 7
10	Десятичная	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
12	Двенадцатеричная	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B
16	Шестнадцатеричная	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Например (индекс внизу указывает основание системы счисления):

$$23,43_{(10)} = 2 * 10^1 + 3 * 10^0 + 4 * 10^{-1} + 3 * 10^{-2}$$

(в данном примере знак «3» в одном случае означает число единиц, а в другом — число сотых долей единицы);

$$692_{(10)} = 6 * 10^2 + 9 * 10^1 + 2$$

(«шестьсот девяносто два» с формальной точки зрения представляется в виде: «шесть умножить на десять в степени два, плюс девять умножить на десять в степени один, плюс два»);

$$1101_{(2)} = 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0;$$

$$112_{(3)} = 1 * 3^2 + 1 * 3^1 + 2 * 3^0;$$

$$341,5_{(8)} = 3 * 8^2 + 4 * 8^1 + 1 * 8^0 + 5 * 8^{-1};$$

$$A1F,4_{(16)} = A * 16^2 + 1 * 16^1 + F * 16^0 + 4 * 16^{-1}.$$

При работе с компьютерами приходится параллельно использовать несколько позиционных систем счисления (чаще всего двоичную, десятичную и шестнадцатеричную), поэтому большое практическое значение имеют процедуры перевода чисел из одной системы счисления в другую. Заметим, что во всех приведенных выше примерах результат является десятичным числом, и, таким образом, способ перевода чисел из любой позиционной системы счисления в десятичную уже продемонстрирован.

Чтобы перевести целую часть числа из десятичной системы в систему с основанием В, необходимо разделить ее на В. Остаток даст младший разряд числа. Полученное при этом частное необходимо вновь разделить на В — остаток даст следующий разряд числа и т.д. Для перевода дробной части ее необходимо умножить на В. Целая часть полученного произведения будет первым (после запятой, отделяющей целую часть от дробной) знаком. Дробную же часть произведения необходимо вновь умножить на В. Целая часть полученного числа будет следующим знаком и т.д.

Отметим, что кроме рассмотренных выше позиционных систем счисления существуют такие, в которых значение знака не зависит от того места, которое он занимает в числе. Такие системы счисления называются непозиционными. Наиболее известным примером непозиционной системы является римская. В этой системе используется 7 знаков (I, V, X, L, C, D, M), которые соответствуют следующим величинам:

$$I \ (1) \quad V \ (5) \quad X \ (10) \quad L \ (50) \quad C \ (100) \quad D \ (500) \quad M \ (1000)$$

Например: III (три), LIX (пятьдесят девять), DLV (пятьсот пятьдесят пять).

Недостатком непозиционных систем, из-за которых они представляют лишь исторический интерес, является отсутствие формальных правил записи чисел и, соответственно, арифметических действий над ними (хотя по традиции римскими числами часто пользуются при нумерации глав в книгах, веков в истории и др.).

Двоичная система счисления

Особая значимость двоичной системы счисления в информатике определяется тем, что внутреннее представление любой информации в компьютере является двоичным, т.е. описываемым наборами только из двух знаков (0 и 1).

Таблица 1.3
Таблицы сложения
и умножения в двоичной
системе

Сложение	Умножение
+ 0 1	* 0 1
0 0 1	0 0 0
1 1 10	1 0 1

Конкретизируем описанный выше способ в случае перевода чисел из десятичной системы в двоичную. Целая и дробная части переводятся порознь. Для перевода целой части (или просто целого) числа необходимо разделить ее на основание системы счисления и продолжать делить частные от деления до тех пор, пока частное не станет равным 0. Значения получившихся остатков, взятые в обратной последовательности, образуют искомое двоичное число.

Например:

Остаток	
$25 : 2 = 12$	(1);
$12 : 2 = 6$	(0);
$6 : 2 = 3$	(0);
$3 : 2 = 1$	(1);
$1 : 2 = 0$	(1).

Таким образом,

$$25_{(10)} = 11001_{(2)}.$$

Для перевода дробной части (или числа, у которого «0» целых) надо умножить ее на 2. Целая часть произведения будет первой цифрой числа в двоичной системе. Затем, отбрасывая у результата целую часть, вновь умножаем на 2 и т.д. Заметим, что конечная десятичная дробь при этом вполне может стать бесконечной (периодической) двоичной. Например:

$$\begin{aligned} 0,73 \cdot 2 &= 1,46 \text{ (целая часть 1);} \\ 0,46 \cdot 2 &= 0,92 \text{ (целая часть 0);} \\ 0,92 \cdot 2 &= 1,84 \text{ (целая часть 1);} \\ 0,84 \cdot 2 &= 1,68 \text{ (целая часть 1) и т.д.} \end{aligned}$$

В итоге

$$0,73_{(10)} = 0,1011\dots_{(2)}.$$

Над числами, записанными в любой системе счисления, можно производить различные арифметические операции. Так, для сложения и умножения двоичных чисел необходимо использовать табл. 1.3.

Заметим, что при двоичном сложении $1 + 1$ возникает перенос единицы в старший разряд — точь-в-точь как в десятичной арифметике:

$$\begin{array}{r} 1001 \\ + 11 \\ \hline 1100 \end{array} \quad \begin{array}{r} * 1001 \\ 11 \\ \hline 111 \\ + 111 \\ \hline 10101 \end{array}$$

Восьмеричная и шестнадцатеричная системы счисления

С точки зрения изучения принципов представления и обработки информации в компьютере, обсуждаемые в этом подразд. системы представляют большой интерес. Хотя компьютер «знает» только двоичную систему счисления, часто с целью

уменьшения количества записываемых на бумаге или вводимых с клавиатуры компьютера знаков бывает удобнее пользоваться восьмеричными или шестнадцатеричными числами, тем более что, как будет показано далее, процедура взаимного перевода чисел из каждой из этих систем в двоичную очень проста — гораздо проще переводов между любой из этих трех систем и десятичной.

Перевод чисел из десятичной системы счисления в восьмеричную производится (по аналогии с двоичной системой счисления) с помощью делений и умножений на 8. Например, переведем число $58,32_{(10)}$:

$$58 : 8 = 7 \text{ (2 в остатке);}$$

$$7 : 8 = 0 \text{ (7 в остатке).}$$

$$0,32 \cdot 8 = 2,56;$$

$$0,56 \cdot 8 = 4,48;$$

$$0,48 \cdot 8 = 3,84, \dots$$

Таким образом,

$$58,32_{(10)} = 72,243\dots_{(8)}$$

(из конечной дроби в одной системе может получиться бесконечная дробь в другой).

Перевод чисел из десятичной системы счисления в шестнадцатеричную производится аналогично.

С практической точки зрения представляет интерес процедура взаимного преобразования двоичных, восьмеричных и шестнадцатеричных чисел. Для этого воспользуемся представлением чисел от 0 до 15 (в десятичной системе счисления) в других системах счисления (табл. 1.4).

Для перевода целого двоичного числа в восьмеричное необходимо разбить его справа налево на группы по три цифры (самая левая группа может содержать менее трех двоичных цифр), а затем каждой группе поставить в соответствие ее восьмеричный эквивалент. Например:

$$11011001 = 11\ 011\ 001, \text{ т.е. } 11011001_{(2)} = 331_{(8)}.$$

Заметим, что группу из трех двоичных цифр часто называют «двоичной триадой».

Таблица 1.4

Соответствие чисел в различных системах счисления

Десятичная	Шестнадцатеричная	Восьмеричная	Двоичная	Десятичная	Шестнадцатеричная	Восьмеричная	Двоичная
0	0	0	0	8	8	10	1000
1	1	1	1	9	9	11	1001
2	2	2	10	10	A	12	1010
3	3	3	11	11	B	13	1011
4	4	4	100	12	C	14	1100
5	5	5	101	13	D	15	1101
6	6	6	110	14	E	16	1110
7	7	7	111	15	F	17	1111

Перевод целого двоичного числа в шестнадцатеричное производится путем разбиения данного числа на группы по четыре цифры — «двоичные тетрады»:

$$1100011011001 = 1\ 1000\ 1101\ 1001, \text{ т.е. } 1100011011001_{(2)} = 18D9_{(16)}.$$

Для перевода дробных частей двоичных чисел в восьмеричную или шестнадцатеричную системы аналогичное разбиение на триады или тетрады производится от запятой вправо (с дополнением недостающих последних цифр нулями):

$$0,1100011101_{(2)} = 0,110\ 001\ 110\ 100 = 0,6164_{(8)};$$

$$0,1100011101_{(2)} = 0,1100\ 0111\ 0100 = 0,C74_{(16)}.$$

Перевод восьмеричных (шестнадцатеричных) чисел в двоичные производится обратным путем — сопоставлением каждому знаку числа соответствующей тройки (четверки) двоичных цифр.

Преобразования чисел из двоичной в восьмеричную и шестнадцатеричную системы и наоборот столь просты (по сравнению с операциями между этими тремя системами и привычной нам десятичной) потому, что числа 8 и 16 являются целыми степенями числа 2. Этой простотой и объясняется популярность восьмеричной и шестнадцатеричной систем в вычислительной технике и программировании.

Арифметические действия с числами в восьмеричной и шестнадцатеричной системах счисления выполняются по аналогии с двоичной и десятичной системами. Для этого необходимо воспользоваться соответствующими таблицами. Для примера табл. 1.5 иллюстрирует сложение и умножение восьмеричных чисел.

Рассмотрим еще один возможный способ перевода чисел из одной позиционной системы счисления в другую — метод вычитания степеней. В этом случае из числа последовательно вычитается максимально допустимая степень требуемого основания, умноженная на максимальный возможный коэффициент, меньший основания; этот коэффициент и является значащей цифрой числа в новой системе. Например, число $114_{(10)}$:

$$114 - 2^6 = 114 - 64 = 50;$$

$$50 - 2^5 = 50 - 32 = 18;$$

$$18 - 2^4 = 2;$$

$$2 - 2^1 = 0.$$

Таблица 1.5

Таблицы сложения и умножения в восьмеричной системе

Сложение

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

Умножение

*	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	25
4	0	4	10	14	20	24	30	34
5	0	5	12	17	24	31	36	43
6	0	6	14	22	30	36	44	52
7	0	7	16	25	34	43	52	61

Таким образом, $114_{(10)} = 1110010_{(2)}$.

$$114 - 1 \cdot 8^2 = 114 - 64 = 50;$$

$$50 - 6 \cdot 8^1 = 50 - 48 = 2;$$

$$2 - 2 \cdot 8^0 = 2 - 2 = 0.$$

Итак, $114_{(10)} = 162_{(8)}$.

1.2.4. Международные системы байтового кодирования

Информатика и ее приложения интернациональны. Это связано как с объективными потребностями человечества в единых правилах и законах хранения, передачи и обработки информации, так и с тем, что в этой сфере деятельности (особенно в ее прикладной части) заметен приоритет одной страны, которая благодаря этому получает возможность «диктовать моду».

Компьютер считают универсальным преобразователем информации. Тексты на естественных языках и числа, математические и специальные символы — одним словом все, что в быту или в профессиональной деятельности может быть необходимо человеку, должно иметь возможность быть введенным в компьютер.

В силу безусловного приоритета двоичной системы счисления при внутреннем представлении информации в компьютере кодирование «внешних» символов основывается на сопоставлении каждому из них определенной группы двоичных знаков. При этом из технических соображений и из соображений удобства кодирования-декодирования следует пользоваться равномерными кодами, т. е. двоичными группами равной длины.

Попробуем подсчитать наиболее короткую длину такой комбинации с точки зрения человека, заинтересованного в использовании лишь одного естественного алфавита — скажем, английского: 26 букв следует умножить на 2 (прописные и строчные) — итого 52; 10 цифр; будем считать, 10 знаков препинания; 10 разделительных знаков (три вида скобок, пробел и др.); знаки привычных математических действий; несколько специальных символов (типа #, \$, & и др.) — итого примерно 100. Точный подсчет здесь не нужен, поскольку нам предстоит решить простейшую задачу: имея равномерный код из групп по N двоичных знаков, сколько можно образовать разных кодовых комбинаций? Ответ очевиден: $K = 2^N$. Итак, при $N = 6$ $K = 64$ — явно мало, при $N = 7$ $K = 128$ — вполне достаточно.

Однако для кодирования нескольких (хотя бы двух) естественных алфавитов (плюс все отмеченные выше знаки) и этого недостаточно. Минимально достаточное значение N в этом случае 8; имея 256 комбинаций двоичных символов, вполне можно решить указанную задачу. Поскольку 8 двоичных символов составляют 1 байт, то говорят о системах «байтового» кодирования.

Наиболее распространены две такие системы: EBCDIC (Extended Binary Coded Decimal Interchange Code) и ASCII (American Standard Cod Information Interchange). Первая исторически тяготеет к «большим» машинам, вторая чаще используется на мини- и микроЭВМ (в том числе и на персональных компьютерах). Ознакомимся подробнее именно с ASCII, созданной в 1963 г.

В своей первоначальной версии это система семибитного кодирования. Она ограничивалась одним естественным алфавитом (английским), цифрами и набором различных символов, включающих «символы пишущей машинки» (привычные знаки препинания, знаки математических действий и др.) и «управляющие символы». Примеры последних легко найти на клавиатуре компьютера: для микроЭВМ, например, DEL — знак удаления символа.

В следующей версии фирма IBM перешла на расширенную 8-битную кодировку. В ней первые 128 символов совпадают с исходными и имеют коды со старшим битом, равным нулю, а остальные коды отданы под буквы некоторых европейских языков, в основе которых лежит латиница, греческие буквы, математические символы (например знак квадратного корня) и символы псевдографики. С помощью последних можно создавать таблицы, несложные схемы и др.

Для представления букв русского языка (кириллицы) в рамках ASCII было предложено несколько версий. Первоначально был разработан стандарт под названием КОИ-7, оказавшийся по ряду причин крайне неудачным; ныне он практически не используется.

В табл. 1.6 приведена часто используемая в нашей стране модифицированная альтернативная кодировка. В левую часть входят исходные коды ASCII, в правую часть (расширение ASCII) вставлены буквы кириллицы взамен букв немецкого, французского алфавитов (не совпадающих по написанию с английскими), греческих букв, некоторых специальных символов.

Знакам алфавита ПЭВМ ставятся в соответствие шестнадцатеричные числа по правилу: первая — номер столбца, вторая — номер строки. Например: английская «A» — код 41, русская «и» — код A8.

Одним из достоинств этой системы кодировки русских букв является их естественное упорядочение, т. е. номера букв следуют друг за другом в том же порядке, в каком сами буквы стоят в русском алфавите. Это очень существенно при решении ряда задач обработки текстов, когда требуется выполнить или использовать лексикографическое упорядочение слов.

Отметим, что даже 8-битная кодировка недостаточна для кодирования всех символов, которые хотелось бы иметь в расширенном алфавите. Все препятствия

Таблица 1.6

Таблица кодов ASCII (расширенная)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	►			0	@	P	'	р	А	Р	а	■	Л	॥	р	Ё
1	☺	◀	!	1	A	Q	а	҆	Б	С	б	■	Л	҆	с	ё
2	☻	↑	"	2	B	R	b	г	В	Т	в	■	Т	҆	т	€
3	♥	!!	#	3	C	S	c	s	Г	У	г		†	҆	у	€
4	♦	¶	\$	4	D	T	d	t	Д	Ф	д	-	-	ے	Ф	Ї
5	♣	§	%	5	E	U	e	u	Е	Х	е	‡	+	ؒ	خ	ۿ
6	♠	-	&	6	F	V	f	v	Ж	Ц	ж	‡	ؒ	ؒ	ؒ	ؒ
7	•	↑	'	7	G	W	g	w	З	Ч	з	¶	¶	¶	چ	ۊ
8	▣	↑	(8	H	X	h	x	И	Ш	и	ؒ	ؒ	ؒ	ؒ	ؒ
9	○	↓)	9	I	Y	i	y	Й	Щ	й	‡	ؒ	ؒ	ؒ	ؒ
A	▣	→	*	:	J	Z	j	z	К	Ҋ	к		॥	گ	Ҋ	.
B	♂	←	+	:	K	[k	{	Л	Ы	л	¶	҆	■	ы	√
C	♀	↔	,	<	L	\	l	!	М	Ь	м	ؒ	ؒ	ؒ	ؒ	ؒ
D	♪	↔	-	=	M]	m	}	Н	Э	н	ؒ	=	ؒ	ؒ	ؒ
E	♫	▲	.	>	N	^	n	~	О	Ю	о	ؒ	ؒ	ؒ	ؒ	ؒ
F	⊗	▼	/	?	O	-	o	◊	П	Я	پ	ؒ	ؒ	ؒ	ؒ	ؒ

могут быть сняты при переходе на 16-битную кодировку Unicode, допускающую 65 536 кодовых комбинаций.

1.2.5. Теоремы Шеннона

Ранее отмечалось, что при передаче сообщений по каналам связи могут возникать помехи, способные привести к искажению принимаемых знаков. Так, например, если вы попытаетесь в ветреную погоду передать речевое сообщению человеку, находящемуся от вас на значительном расстоянии, то оно может быть сильно искажено такой помехой, как ветер. Вообще передача сообщений при наличии помех является серьезной теоретической и практической задачей. Ее значимость возрастает в связи с повсеместным внедрением компьютерных телекоммуникаций, в которых помехи неизбежны. При работе с кодированной информацией, искажаемой помехами, можно выделить следующие основные проблемы: установление самого факта того, что произошло искажение информации; выяснение того, в каком конкретно месте передаваемого текста это произошло; исправление ошибки, хотя бы с некоторой степенью достоверности.

Помехи в передачи информации — вполне обычное дело во всех сферах профессиональной деятельности и в быту. Один из примеров был уже приведен, другие примеры — разговор по телефону, в трубке которого «трещит», вождение автомобиля в тумане и т. д. Чаще всего человек вполне справляется с каждой из указанных задач, хотя и не всегда отдает себе отчет, как он это делает (т. е. не-алгоритмически, а исходя из каких-то ассоциативных связей). Известно, что естественный язык обладает большой **избыточностью** (в европейских языках — до 7 %), чем объясняется большая помехоустойчивость сообщений, составленных из знаков алфавитов таких языков. Примером, иллюстрирующим устойчивость русского языка к помехам, может служить предложение «в словах все гласно зомононо боквой о». Здесь 26 % символов «поражены», однако это не приводит к потере смысла. Таким образом, в данном случае избыточность является полезным свойством.

Избыточность могла бы быть использована и при передаче кодированных сообщений в технических системах. Например, каждый фрагмент текста (предложение) передается трижды, и верным считается та пара фрагментов, которая полностью совпада. Однако большая избыточность приводит к большим временным затратам при передаче информации и требует большого объема памяти при ее хранении. Впервые теоретическое исследование эффективного кодирования предпринял К. Шенон.

Первая теорема Шеннона декларирует возможность создания системы эффективного кодирования дискретных сообщений, у которой среднее число двоичных символов на один символ сообщения асимптотически стремится к энтропии источника сообщений (в отсутствии помех).

Задача эффективного кодирования описывается триадой:

$$X = \{x_i\} \text{ — кодирующее устройство — } B.$$

Здесь X , B — соответственно входной и выходной алфавит; под множеством x_i можно понимать любые знаки (буквы, слова, предложения); B — множество, число элементов которого в случае кодирования знаков числами определяется основанием системы счисления (например, $m = 2$). Кодирующее устройство сопоставляет каждому сообщению x_i из X кодовую комбинацию, составленную из n_i символов множества B . Ограничением данной задачи является отсутствие помех. Требуется оценить минимальную среднюю длину кодовой комбинации.

Пример к первой теореме Шеннона

N	Px_i	x_i	Код	n_i	$n_i P_i$	$P_i \log P_i$
1	0,19	x_1	10	2	0,38	-0,45522
2	0,16	x_2	001	3	0,48	-0,42301
3	0,16	x_3	011	3	0,48	-0,42301
4	0,15	x_4	100	3	0,45	-0,41054
5	0,12	x_5	101	3	0,36	-0,36706
6	0,11	x_6	111	3	0,33	-0,35028
7	0,09	x_7	1011	4	0,36	-0,31265
8	0,02	x_8	1001	4	0,08	-0,11288
$\Sigma = 1$					$\Sigma = 2,92$	$\Sigma = -2,85$

Для решения данной задачи должна быть известна вероятность P_i появления сообщения x_i , которому соответствует определенное количество символов n_i алфавита B . Тогда математическое ожидание количества символов из B определится следующим образом:

$$n_{\text{ср}} = n_i P_i \text{ (средняя величина).}$$

Этому среднему числу символов алфавита B соответствует максимальная энтропия $H_{\max} = n_{\text{ср}} \log m$. Для обеспечения передачи информации, содержащейся в сообщениях X кодовыми комбинациями из B , должно выполняться условие $H_{\max} \geq H(x)$, или $n_{\text{ср}} \log m \geq -P_i \log P_i$. В этом случае закодированное сообщение имеет избыточность $n_{\text{ср}} \geq H(x)/\log m$, $n_{\min} = H(x)/\log m$.

Коэффициент избыточности

$$K_i = (H_{\max} - H(x))/H_{\max} = (n_{\text{ср}} - n_{\min})/n_{\text{ср}}.$$

Выпишем эти значения в виде табл. 1.7. Имеем:

$$n_{\min} = H(x)/\log 2 = 2,85; \quad K_i = (2,92 - 2,85)/2,92 = 0,024,$$

т. е. код практически не имеет избыточности. Видно, что среднее число двоичных символов стремится к энтропии источника сообщений.

Вторая теорема Шеннона гласит, что при наличии помех в канале всегда можно найти такую систему кодирования, при которой сообщения будут переданы с заданной достоверностью. При наличии ограничения пропускная способность канала должна превышать производительность источника сообщений. Таким образом, вторая теорема Шеннона устанавливает принципы помехоустойчивого кодирования. Для дискретного канала с помехами теорема утверждает, что если скорость создания сообщений меньше или равна пропускной способности канала, то существует код, обеспечивающий передачу со сколь угодно малой частотой ошибок.

Доказательство теоремы основывается на следующих рассуждениях. Первоначально последовательность $X = \{x_i\}$ кодируется символами из B так, что достигается максимальная пропускная способность (канал не имеет помех). Затем в последовательность из B длины n вводится r дополнительных символов и по каналу передается новая последовательность из $n + r$ символов. Эти r дополнительных символов обеспечивают избыточность кодирования информации и позволяют противодействовать помехам. Число возможных последовательностей длины $n + r$ больше чис-

ла возможных последовательностей длины n . Множество всех последовательностей длины $n + r$ может быть разбито на n подмножеств, каждому из которых сопоставлена одна из последовательностей длины n . Воздействие помехи на последовательность из $n + r$ символов выводит ее из соответствующего подмножества с вероятностью, зависящей от r . Это позволит определить на приемной стороне канала, какому подмножеству принадлежит искаженная помехами принятая последовательность длины $n + r$, и тем самым восстановить исходную последовательность длины n .

Эта теорема не дает конкретного метода построения кода, но указывает на пределы достижимого в создании помехоустойчивых кодов, стимулирует поиск новых путей решения этой проблемы.

Контрольные вопросы

1. Как определяется алфавит?
2. Что такое код?
3. В чем отличие позиционной системы счисления от непозиционной?
4. Каковы способы перевода чисел из одной системы счисления в другую?
5. Почему при работе с компьютерами используются восьмеричная и шестнадцатеричная системы счисления?
6. Как выглядят таблицы сложения и умножения в шестнадцатеричной системе?
7. Как объяснить большую помехоустойчивость передаваемых сообщений, составленных на русском языке?
8. В чем суть теорем Шеннона?

1.3. Алгоритм и его свойства

1.3.1. Различные подходы к понятию «алгоритм»

Алгоритм — одно из фундаментальных понятий информатики. Алгоритмизация наряду с моделированием выступает в качестве общего метода информатики. К реализации определенных алгоритмов сводятся процессы управления в различных системах, что делает понятие алгоритма близким и кибернетике.

Алгоритмы являются объектом систематического исследования пограничной между математикой и информатикой научной дисциплины, примыкающей к математической логике — **теории алгоритмов**.

Особенность положения состоит в том, что при решении практических задач, предполагающих разработку алгоритмов для реализации на ЭВМ, и тем более при использовании на практике информационных технологий, можно, как правило, не опираться на высокую формализацию данного понятия. Поэтому представляется целесообразным познакомиться с алгоритмами и алгоритмизацией на основе содержательного толкования сущности понятия алгоритма и рассмотрения основных его свойств. При таком подходе алгоритмизация выступает как набор определенных практических приемов, особых специфических навыков рационального мышления в рамках заданных языковых средств. Можно провести аналогию между этим обстоятельством и рассмотренным выше подходом к измерению информации: тонкие математические построения при «кибернетическом» подходе не очень нужны при использовании гораздо более простого «объемного» подхода при практической работе с компьютером.

Само слово «алгоритм» происходит от *algorithmi* — латинской формы написания имени великого математика IX в. аль-Хорезми, который сформулировал пра-

вила выполнения арифметических действий. Первоначально под алгоритмами и понимали только правила выполнения четырех арифметических действий над многозначными числами.

1.3.2. Понятие «исполнитель алгоритма»

Понятие «исполнитель» невозможно определить с помощью какой-либо формализации. Исполнителем может быть человек, группа людей, робот, станок, компьютер, язык программирования и т.д. Важнейшим свойством, характеризующим любого из этих исполнителей, является то, что исполнитель умеет выполнять некоторые команды. Так, исполнитель-человек умеет выполнять такие команды, как «встать», «сесть», «включить компьютер» и т.д., а исполнитель — язык программирования Бейсик (Basic) — команды PRINT, END, LIST и другие аналогичные. Вся совокупность команд, которые данный исполнитель умеет выполнять, называется **системой команд исполнителя (СКИ)**.

В качестве примера (рис. 1.4) рассмотрим исполнителя-робота*, работа которого состоит в собственном перемещении по рабочему полю (квадрату произвольного размера, разделенному на клетки) и перемещении объектов, в начальный момент времени находящихся на «складе» (правая верхняя клетка).

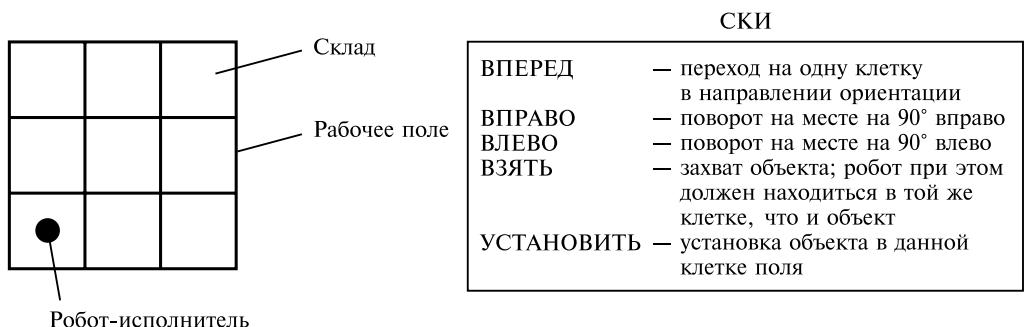


Рис. 1.4. Исполнитель-робот

Одно из принципиальных обстоятельств состоит в том, что исполнитель не вникает в смысл того, что он делает, но получает необходимый результат. В таком случае говорят, что исполнитель действует **формально**, т.е. отвлекается от содержания поставленной задачи и только строго выполняет некоторые правила, инструкции.

Это важная особенность алгоритмов. Наличие алгоритма формализует процесс решения задачи, исключает рассуждение исполнителя. Использование алгоритма дает возможность решать задачу формально, механически исполняя команды алгоритма в указанной последовательности. Целесообразность предусматриваемых алгоритмом действий обеспечивается точным анализом со стороны того, кто составляет этот алгоритм.

Введение в рассмотрение понятия «исполнитель» позволяет определить алгоритм как понятное и точное предписание исполнителю совершить последовательность действий, направленных на достижение поставленной цели. В случае исполнителя-робота мы имеем пример алгоритма «в установке», характеризующегося

* Фрагмент программно-методической системы «Учебные роботы» (Лапчик М.П., Лучко О.Н. Первые уроки алгоритмизации. — Омск: ОмГПУ, 1987).

отсутствием каких-либо величин. Наиболее же распространенными и привычными являются алгоритмы работы с величинами — числовыми, символьными, логическими и т.д.

1.3.3. Графическое представление алгоритмов

Алгоритм, составленный для некоторого исполнителя, можно представить различными способами: с помощью графического или словесного описания, в виде таблицы, последовательностью формул, записанным на алгоритмическом языке (языке программирования). Остановимся на графическом описании алгоритма, называемом **блок-схемой**. Этот способ имеет ряд преимуществ благодаря наглядности, обеспечивающей, в частности, высокую «читаемость» алгоритма и явное отображение управления в нем.

Прежде всего определим понятие блок-схемы. Блок-схема — это ориентированный граф, указывающий порядок выполнения команд алгоритма. Вершины такого графа могут быть одного из трех типов (рис. 1.5): функциональная вершина (F), имеющая один вход и один выход; предикатная вершина (P), имеющая один вход и два выхода, в этом случае функция P передает управление по одной из ветвей в зависимости от значения P (t , т.е. *true*, означает «истина», f , т.е. *false*, — «ложь»); объединяющая вершина (вершина «слияния») (U), обеспечивающая передачу управления от одного из двух входов к выходу. Иногда вместо t пишут «да» (либо знак «+»), вместо f — «нет» (либо знак «-»).

Из данных элементарных блок-схем можно построить четыре блок-схемы (рис. 1.6), имеющих особое значение для практики алгоритмизации: **композиция**, или **следование**; **альтернатива**, или **ветвление**; **итерация**, или **цикл**, с предусловием или постусловием. Обозначения $F1$ и $F2$, используемые на рис. 1.6 представляют собой в общем случае некоторые команды для соответствующего исполнителя, P — это условие, в зависимости от истинности (t) или ложности (f) которого управление передается по одной из двух ветвей. Можно доказать, что для со-

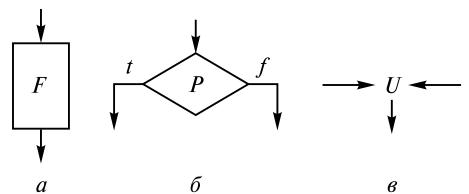


Рис. 1.5. Три типа вершин графа:
а — функциональная; б — предикатная; в — объединяющая

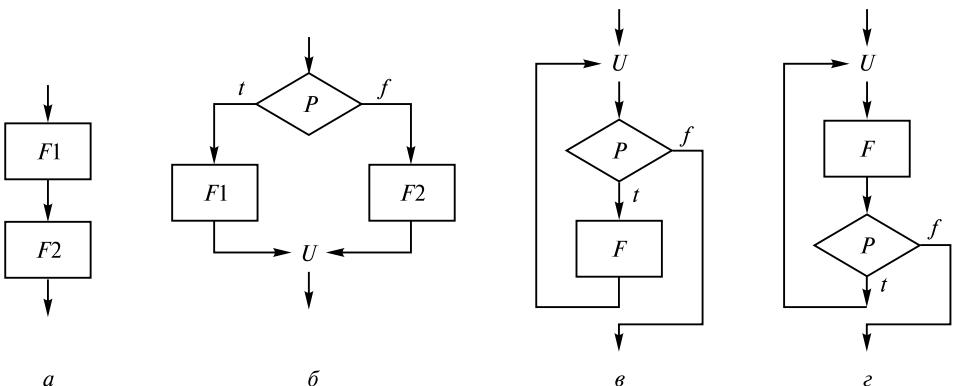


Рис. 1.6. Основные алгоритмические структуры:

а — композиция, или следование; б — альтернатива, или ветвление; в — итерация, или цикл с предусловием; г — итерация, или цикл с постусловием

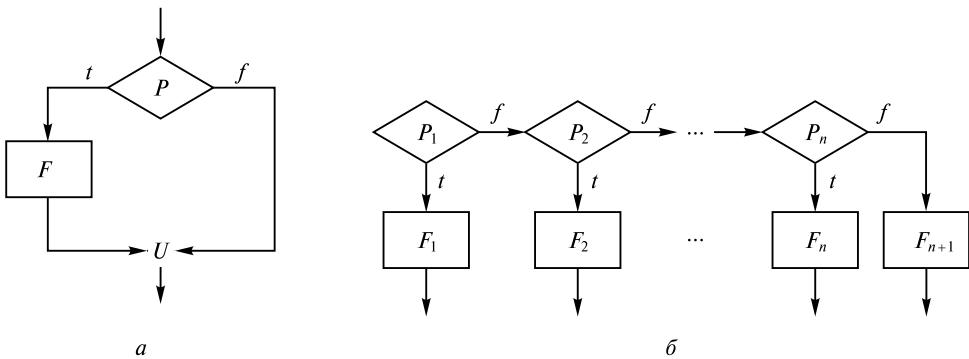


Рис. 1.7. Развитие структуры типа альтернатива:

а — неполное ветвление; *б* — структура выбор

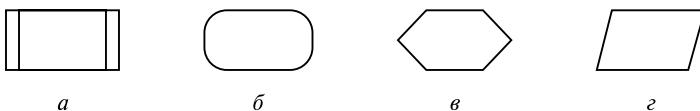


Рис. 1.8. Некоторые дополнительные конструкции для изображения блок-схем алгоритмов:
а — типовой процесс (обращение к процедуре); *б* — начало или завершение алгоритма; *в* — подготовка к циклическому процессу; *г* — ввод-вывод данных

составления любого алгоритма достаточно представлена выше четырех блок-схем, если пользоваться их последовательностями и/или суперпозициями.

Блок-схема альтернатива может иметь и сокращенную форму, в которой отсутствует ветвь F_2 (рис. 1.7, *а*). Развитием блок-схемы типа альтернатива является блок-схема выбор (рис. 1.7, *б*).

На практике при составлении блок-схем оказывается удобным использовать и другие графические знаки (некоторые из них приведены на рис. 1.8).

1.3.4. Свойства алгоритмов

Алгоритм должен быть составлен таким образом, чтобы исполнитель, в расчете на которого он создан, мог однозначно и точно следовать командам алгоритма и эффективно получать определенный результат. Это накладывает на записи алгоритмов ряд обязательных требований, суть которых вытекает, вообще говоря, из приведенного выше неформального толкования понятия алгоритма. Сформулируем эти требования в виде перечня свойств, которым должны удовлетворять алгоритмы, адресуемые заданному исполнителю.

1. Одно из первых требований, которое предъявляется к алгоритму, состоит в том, что описываемый процесс должен быть разбит на последовательность отдельных шагов. Возникающая в результате такого разбиения запись представляет собой упорядоченную совокупность четко разделенных друг от друга предписаний (директив, команд, операторов), образующих прерывную (или, как говорят, дискретную) структуру алгоритма. Только выполнив требования одного предписания, можно приступить к выполнению следующего. Дискретная структура алгоритмической записи может, например, подчеркиваться сквозной нумерацией отдельных команд алгоритма, хотя это требование не является обязательным. Рассмотренное свойство алгоритмов называют **дискретностью**.

2. Используемые на практике алгоритмы составляются с ориентацией на определенного исполнителя. Чтобы составить для него алгоритм, нужно знать, какие команды этот исполнитель может понять и выполнить, а какие — не может. Известно, что у каждого исполнителя имеется своя система команд. Очевидно, составляя запись алгоритма для определенного исполнителя, можно использовать лишь те команды, которые имеются в его СКИ. Это свойство алгоритмов называют **понятностью**.

3. Будучи понятным, алгоритм не должен содержать предписаний, смысл которых может восприниматься неоднозначно. Запись алгоритма должна быть настолько четкой, полной и продуманной в деталях, чтобы у исполнителя не могло возникнуть потребности в принятии решений, не предусмотренных составителем алгоритма. Говоря иначе, алгоритм не должен оставлять места для производа исполнителя. Кроме того, в алгоритмах недопустимы также ситуации, когда после выполнения очередной команды алгоритма исполнителю неясно, какая из команд алгоритма должна выполняться на следующем шаге.

Отмеченное свойство алгоритмов называют **определенностью** или **детерминированностью**.

4. Обязательное требование к алгоритмам — **результативность**. Смысл этого требования состоит в том, что при точном исполнении всех предписаний алгоритма процесс должен прекратиться за конечное число шагов и при этом должен получиться определенный результат. Вывод о том, что решения не существует — тоже результат.

5. Наиболее распространены алгоритмы, обеспечивающие решение не одной конкретной задачи, а некоторого класса задач данного типа. Это свойство алгоритма называют **массовостью**. В простейшем случае массовость обеспечивает возможность использования различных исходных данных.

1.3.5. Понятие алгоритмического языка

Достаточно распространенным способом представления алгоритма является его запись на **алгоритмическом языке**, представляющем в общем случае систему обозначений и правил для единообразной и точной записи алгоритмов и исполнения их. Отметим, что между понятиями «алгоритмический язык» и «языки программирования» есть различие; прежде всего, под исполнителем в алгоритмическом языке может подразумеваться не только компьютер, но и устройство для работы «в обстановке». Программа, записанная на алгоритмическом языке, не обязательно предназначена компьютеру. Практическая же реализация алгоритмического языка — отдельный вопрос в каждом конкретном случае.

Как и каждый язык, алгоритмический язык имеет свой словарь. Основу этого словаря составляют слова, употребляемые для записи команд, входящих в систему команд исполнителя того или иного алгоритма. Такие команды называются **простыми** командами. В алгоритмическом языке используют слова, смысл и способ употребления которых задан раз и навсегда. Эти слова называются **служебными**. Использование служебных слов делает запись алгоритма более наглядной, а форму представления различных алгоритмов — единообразной.

Алгоритм, записанный на алгоритмическом языке, должен иметь название. Название желательно выбирать так, чтобы было ясно, решение какой задачи описывает данный алгоритм. Для выделения названия алгоритма перед ним записывают служебное слово **АЛГ** (АЛГоритм). За названием алгоритма (обычно с новой строки) записывают его команды. Для указания начала и конца алгоритма его команды заключают в пару служебных слов **НАЧ** (НАЧало) и **КОН** (КОНец). Команды записывают последовательно.

Последовательность записи алгоритма:

```
АЛГ название алгоритма
НАЧ
    серия команд алгоритма
КОН
```

Например, алгоритм, определяющий движение исполнителя-робота (см. рис. 1.4), может иметь вид:

```
АЛГ в_склад
НАЧ
    вперед
    вправо
    вперед
    вперед
    влево
    вперед
КОН
```

При построении новых алгоритмов могут использоваться алгоритмы, составленные ранее. Алгоритмы, целиком используемые в составе других алгоритмов, называют **вспомогательными** алгоритмами. Вспомогательным может быть любой алгоритм из числа ранее составленных. Не исключается также, что вспомогательным может оказаться алгоритм, сам содержащий ссылку на вспомогательные алгоритмы.

Очень часто при составлении алгоритмов возникает необходимость использования в качестве вспомогательного одного и того же алгоритма, который к тому же может быть весьма сложным и громоздким. Было бы нерационально, начиная работу, каждый раз заново составлять и запоминать такой алгоритм для его последующего использования. Поэтому в практике широко используют так называемые встроенные (или стандартные) вспомогательные алгоритмы, т. е. такие алгоритмы, которые постоянно имеются в распоряжении исполнителя. Обращение к таким алгоритмам осуществляется так же, как и к «обычным» вспомогательным алгоритмам. У исполнителя-робота встроенным вспомогательным алгоритмом может быть перемещение в склад из любой точки рабочего поля; у исполнителя — язык программирования Бейсик — это, например, встроенный алгоритм SIN.

Алгоритм может содержать обращение к самому себе как вспомогательному, и в этом случае его называют **рекурсивным**. Если команда обращения алгоритма к самому себе находится в самом алгоритме, то такую рекурсию называют **прямой**. Возможны случаи, когда рекурсивный вызов данного алгоритма происходит из вспомогательного алгоритма, к которому в данном алгоритме имеется обращение. Такая рекурсия называется **косвенной**.

Пример прямой рекурсии:

```
АЛГ движение
НАЧ
    вперед
    вперед
    вправо
    движение
КОН
```

Алгоритмы, при исполнении которых порядок следования команд определяется в зависимости от результатов проверки некоторых условий, называют **разветвительными**.

вляющимися. Для их описания в алгоритмическом языке используют специальную составную команду — команду **ветвления**. Она соответствует блок-схеме «альтернатива» и также может иметь полную или сокращенную форму. Применительно к исполнителю-роботу условием может быть проверка нахождения робота у края рабочего поля (край/не_край); проверка наличия объекта в текущей клетке (есть/нет) и некоторые другие:

ЕСЛИ условие ТО серия1 ИНАЧЕ серия2 ВСЕ	ЕСЛИ условие ТО серия ВСЕ	ЕСЛИ край ТО вправо ИНАЧЕ вперед ВСЕ
--	--	---

Ниже приводится запись на алгоритмическом языке команды выбора (см. рис. 1.7, б), являющейся развитием команды ветвления:

ВЫБОР
 ПРИ условие 1: серия 1
 ПРИ условие 2: серия 2
 ...
 ПРИ условие N: серия N
 ИНАЧЕ серия N+1
ВСЕ

Алгоритмы, при исполнении которых отдельные команды или серии команд выполняются неоднократно, называют циклическими. Для организации циклических алгоритмов в алгоритмическом языке используют специальную составную команду цикла. Она соответствует блок-схемам типа «итерация» и может принимать следующий вид:

ПОКА условие НЦ серия КЦ	НЦ серия ДО условие КЦ
--	--

В случае составления алгоритмов работы с величинами можно рассмотреть и другие возможные алгоритмические конструкции, например цикл с параметром, или выбор. Подробно эти конструкции будут рассматриваться при знакомстве с реальными языками программирования.

В заключение приведем алгоритм, составленный для исполнителя-робота, по которому робот переносит все объекты со склада в левый нижний угол рабочего поля (поле может иметь произвольные размеры):

АЛГ перенос НАЧ в_угол3 ЕСЛИ есть ТО взять в_угол3 установить перенос ИНАЧЕ в_угол3 ВСЕ КОН	АЛГ в_угол3 НАЧ до_края вправо до_края вправо КОН	АЛГ до_края НАЧ ПОКА не_край НЦ вперед КЦ КОН
--	--	--

Контрольные вопросы

1. Каковы возможные подходы к определению понятия «алгоритм»?
2. Кто (что) может быть исполнителем алгоритма?
3. В чем особенности графического способа представления алгоритмов?
4. Каковы основные алгоритмические структуры?
5. Чем определяются свойства алгоритмов дискретность, определенность, понятность, результативность, массовость?
6. Что такое алгоритмический язык?
7. Расскажите о командах алгоритмического языка.

1.4. Формализация понятия «алгоритм»

1.4.1. Постановка проблемы

Понятие алгоритма, введенное в подразд. 1.3, можно назвать понятием алгоритма в интуитивном смысле. Оно имеет нечеткий, неформальный характер, ссылается на некоторые точно не определенные, но интуитивно понятные вещи. Например, при определении и обсуждении свойств алгоритма мы исходили из возможностей некоторого исполнителя алгоритма. Его наличие предполагалось, но ничего определенного о нем не было известно. Говоря языком математики, ни аксиоматического, ни исчерпывающего конструктивного определения исполнителя мы так и не дали.

Установленные в предыдущем подразделе свойства алгоритмов следует называть эмпирическими. Они выявлены на основе обобщения свойств алгоритмов различной природы и имеют прикладной характер. Этих свойств достаточно для практического программирования, для создания обширного круга программ для компьютеров, станков с ЧПУ, промышленных роботов и т.д. Однако как фундаментальное научное понятие алгоритм требует более обстоятельного изучения. Оно невозможно без уточнения понятия «алгоритм», более строгого его описания или, как еще говорят, без его **формализации**.

Известно несколько подходов к формализации понятия «алгоритм»:

- теория конечных и бесконечных автоматов;
- теория вычислимых (рекурсивных) функций;
- λ -исчисление Черча.

Все эти возникшие исторически независимо друг от друга подходы оказались впоследствии эквивалентными.

Главная цель формализации понятия алгоритма такова: подойти к решению проблемы алгоритмической разрешимости различных математических задач, т.е. ответить на вопрос, может ли быть построен алгоритм, приводящий к решению задачи. Мы рассмотрим постановку этой проблемы и некоторые результаты теории алгоритмической разрешимости задач, но вначале обсудим формализацию понятия алгоритма в теории автоматов на примере машин Поста, Тьюринга, а также нормальных алгоритмов Маркова, а затем — основы теории рекурсивных функций. Идеи λ -исчислений Черча реализованы в языке программирования Лисп (см. гл. 3).

Вместе с тем, формально определенный любым из известных способов алгоритм не может в практическом программировании заменить то, что мы называли алгоритмами в предыдущем подразделе. Основная причина состоит в том, что формальное определение резко сужает круг рассматриваемых задач, делая многие практически важные задачи недоступными для рассмотрения.