МИНСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ИНГУШСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

КАФЕДРА «Информационных систем и техналогий»

И Н Ф О Р М А Т И К А

Учебное пособие для студентов, обучающихся по специальности «ИСиТ»

Магас
2022

ПРЕДИСЛОВИЕ

Место образования в жизни современного общества во многом определяется возрастающей ролью знаний, информации, что отразилось в концепциях информационного общества, становлении информационной цивилизации, информатизации образования. Благодаря быстрому развитию компьютерных технологий люди имеют доступ к самой разнообразной информации в любой точке планеты, обмениваются информацией, общаются в режиме реального времени. Для свободной ориентации в информационных потоках современный специалист любого профиля должен уметь получать, обрабатывать и использовать информацию с помощью компьютеров, телекоммуникаций и других средств связи. Информация становится движущей силой технического, экономического, культурно-коммуникативного, социального развития мира и человека, отличающегося проектно-ориентированным интеллектом, способностью к позитивной коммуникации и социальной ответственности перед собой, обществом, природой и культурной средой.

Настоящее учебное пособие составлено в соответствии с учебной программой по информатике и охватывает следующие основные разделы: информатика и информация, системы счисления, основы программирования.

Раздел 1. ИНФОРМАТИКА И ИНФОРМАЦИЯ

1.1. Информатика как наука

Информатика – фундаментальная научная дисциплина, которая изучает информационные процессы, происходящие в системах различной природы, и возможность их автоматизации [2, с. 7].

Термин информатика возник во Франции в 60-х годах. Так называли область, занимающуюся автоматизированной обработкой информации с помощью ЭВМ. Информатика – informatique, в переводе с французского, означает «информационная

автоматика или автоматизированная переработка информации». Термин образован путем слияния слов information (информация) и automatique (автоматика). Следует обратить внимание на то, что не говорят «автоматическая обработка», а говорят именно «автоматизированная», так как полностью автоматическим процесс обработки информации, то есть без участия человека, быть не может. Иногда информатику называют computer science (наука о компьютерной технике).

Позже, когда информатика стала самостоятельной областью человеческой деятельности, термин приобрел новый смысл. Сегодня он используется не только для отображения достижений компьютерной техники, но и для обозначения процессов, связанных с передачей и обработкой информации.

Информатика в широком смысле представляет собой единство разнообразных отраслей науки, техники и производства, связанных с переработкой информации главным образом (но не единственным!) с помощью компьютеров и телекоммуникационных средств связи во всех сферах человеческой деятельности.

В узком смысле информатика – наука, состоящая из трех взаимосвязанных частей – технических средств (hardware), программных средств(software), алгоритмических средств (brainware).

Главная функция информатики заключается в разработке методов и средств преобразования информации и их использовании в организации технологического процесса переработки информации.

Можно выделить следующие задачи информатики [3, с. 37]:

- исследование информационных процессов любой природы;
- разработка информационной техники и создание новейшей технологии переработки информации на базе полученных результатов исследования информационных процессов;
- решение научных и инженерных проблем создания, внедрения и обеспечения эффективного использования компьютерной техники и технологии во всех сферах общественной жизни.

Комплекс индустрии информатики станет ведущим в информационном обществе. Тенденция ко все большей информированности в обществе в существенной степени зависит от прогресса информатики как единства науки, техники и производства.

«Информатика служит прежде всего для формирования определенного мировоззрения в информационной сфере и освоения информационной культуры, то

есть умения целенаправленно работать с информацией, профессионально используя для ее получения, обработки и передачи компьютерную информационную технологию и соответствующие ей технические и программные средства» [3, с. 6].

До недавнего времени большинство специалистов по компьютерной технике и информационным технологиям склонялось к мнению, что информатика – это дисциплина практического применения, так как обучаемый учится работать на персональном компьютере и изучает наиболее распространенные программные средства. Но сегодня информатика стала уверенно отходить от автоматики. Технический аспект науки определился в отдельное направление – computer science. Появились новые разделы информатики, связанные с информационным управлением, информационной безопасностью, формализацией информации и т.д.

Основной целью изучения информатики должно быть достижение уровня общей грамотности в области информатики и информационных технологий. Эта грамотность должна встать в один ряд с такими компонентами результатов общего образования как математическая грамотность, языковая грамотность, естественнонаучная и т.д.

Изучение информатики должно сформировать информационно грамотную личность, способную эффективно работать с информацией. Такой человек должен:

- понимать, каких сведений не хватает для решения поставленных задач;
- представлять, где и как найти недостающие сведения, в том числе сопоставляя сведения из разных источников или анализируя известные данные;
- анализировать полученные сведения и проверять их на истинность, непредвзятость, актуальность;
- усваивать, осмыслять, понимать найденные сведения, сопоставляя их с известными ранее, переводить сведения в знания;
- фиксировать полученные сведения, а также результат их анализа и осмысления и сохранять для последующего использования;
- излагать свои знания, как в индивидуальной беседе, так и публично, включая выступления перед разными категориями слушателей с учетом этих категорий;
- создавать печатные и электронные издания разных видов и жанров для донесения читателю своей точки зрения;
- применять полученные сведения в своей научной деятельности, в том числе для принятия решений;
- использовать современные технические средства при выполнении перечисленных видов работы с информацией;

- уметь выполнять данные виды работы с информацией как индивидуально, так и в составе группы.

Кроме того, информатика, как учебная дисциплина в педвузе, должна продолжать формировать информационную культуру будущего учителя. Здесь следует отметить, что информационная культура учителя является частью его общей культуры, ее гуманистической и технологической составляющих.

Цели обучения информатике сформулировал К.К. Колин:

- формирование новой информационной культуры российского общества, которую должна составлять совокупность профессиональных, социальных и этических норм поведения людей в новой, высокоавтоматизированной информационной среде обитания в XXI веке;
- формирование целостного миропонимания и современного научного мировоззрения, которые должны быть основаны на признании единства основных информационных процессов в природе и обществе, а также понимании ведущей роли информации в эволюционных процессах и обеспечении жизнедеятельности природных и социальных систем;
- подготовка высокообразованных людей и высококвалифицированных специалистов, способных к профессиональному росту и профессиональной мобильности в условиях информатизации общества и развития наукоемких технологий [5].

Ядро современной информатики образуют четыре составные части [1, с. 24], каждая из которых может рассматриваться как относительно самостоятельная научная дисциплина.

Теоретическая информатика – часть информатики, занимающаяся изучением структуры и общих свойств информации и информационных процессов, разработкой общих принципов построения информационной техники и технологии. Она основана на использовании математических методов и включает в себя такие основные математические разделы, как теория алгоритмов и автоматов, теория информации и теория кодирования, теория формальных языков и грамматик, исследование операций и др.

Средства информатизации (технические и программные) – раздел, занимающийся изучением общих принципов построения вычислительных устройств и систем обработки и передачи данных, а также вопросов, связанных с разработкой систем программного обеспечения.

Информационные системы и технологии – раздел информатики, связанный с решением вопросов по анализу потоков информации, их оптимизации, структурировании в различных сложных системах, разработкой принципов реализации в данных системах информационных процессов.

Изучением закономерностей и форм движения информации в обществе, возникающих в современном обществе информационных, психологических, социально-экономических проблем и методов их решения занимается новое направление исследований в области информатики – *социальная информатика.*

Информатика – очень широкая сфера научных знаний, возникшая на стыке нескольких фундаментальных и прикладных дисциплин. К фундаментальным принято относить те науки, основные понятия которых носят общенаучный характер, используются во многих других науках и видах деятельности.

Как комплексная научная дисциплина информатика связана с:

- философией и психологией – через учение об информации и теорию познания;
- математикой – через теорию математического моделирования, дискретную математику, математическую логику и теорию алгоритмов;
- лингвистикой – через учение о формальных языках и о знаковых системах;
- кибернетикой – через теорию информации и теорию управления;
- физикой и химией, электроникой и радиотехникой – через «материальную» часть компьютера и информационных систем.

Роль информатики в развитии общества чрезвычайно велика. Она является научным фундаментом процесса информатизации общества. С ней связано прогрессивное увеличение возможностей компьютерной техники, развитие информационных сетей, создание новых информационных технологий, которые приводят к значительным изменениям во всех сферах общества: в производстве, науке, образовании, медицине и т. д. [1].

Литература

1. Акулов, О.А. Информатика: базовый курс: учеб. пособие для студентов / О.А. Акулов, Н.В. Медведев. – М.: Омега-Л, 2005. – 552 с.

2. Бешенков, С.А. Информатика. Систематический курс. Учебник для 10 класса / С.А. Бешенков, Е.А. Ракитина – М.: Лаборатория Базовых Знаний, 2001. – 432 с.

3. Информатика. Базовый курс / Симонович С.В. [и др.] – СПб: Питер, 2005. – 640 с.

4. Информатика: Учебник / Под ред. Н.В. Макаровой. – М.: Финансы и статистика, 2005. – 768 с.

5. Колин, К.К. О структуре и содержании образовательной области «Информатика» / К.К. Колин // Информатика и образование. – 2000. – №10. – С. 2-4.

1.2. Основы теории информации

Вся жизнь человека так или иначе связана с накоплением и обработкой информации, которую он получает из окружающего мира, используя пять органов чувств – зрение, слух, вкус, обоняние и осязание. Как научная категория «информация» составляет предмет изучения для самых разных дисциплин: информатики, кибернетики, философии, физики, биологии, теории связи т. д. Несмотря на это, строгого научного определения, что же такое информация, до настоящего времени не существует, а вместо него используют понятие об информации. Понятие отличается от определений тем, что разные дисциплины в разных областях науки и техники вкладывают в него разный смысл с тем, чтобы оно в наибольшей степени соответствовало предмету и задачам конкретной дисциплины. Имеется множество определений и понятия информации, от наиболее общего философского (информация есть отражение реального мира) до наиболее частного прикладного (информация есть сведения, являющиеся объектом переработки). Вот некоторые из них [1, с. 4]:

- сообщение, осведомление о положении дел, сведения о чем-либо, передаваемые модели;
- уменьшаемая, снимаемая неопределенность в результате получения сообщений;
- передача, отражение разнообразия в любых процессах и объектах, отраженное разнообразие;
- товар, являющийся объектом купли-продажи знаний для достижения определенных целей;
- данные как результат организации символов в соответствии с установленными правилами;
- продукт взаимодействия данных и адекватных им методов;

- сведения о лицах, предметах, фактах, событиях, явлениях и процессах независимо от формы их представления.

Высшей, специфической формой отражения является сознание человека. Кроме этого, существуют и другие формы – психическая (присущая не только человеку, но и животным, несущая в себе информацию, способную влиять на их эмоциональные состояния и поведение), раздражимость (охватывающая, помимо прочего, растения и простейшие организмы, регулирующие на слабые механические, химические, тепловые контакты с окружающей средой) и самая элементарная форма – запечатление взаимодействия (присущая и неорганической природе, и элементарным частицам, т. е. материи вообще).

Так, кусок каменного угля несет в себе «отражение» событий, произошедших в далекие времена, т. е. обладает свойством информативности. Деловое письмо с предложениями сотрудничества информативно, так как отражает серьезные намерения определенного учреждения или ведомства. Команда приступить к конкретным действиям (запуску ракеты, отправки груза, производству вычислений и т. п.) содержит информацию о подготовленности соответствующих служб и своевременности предпринимаемых шагов.

Информация не является ни материей, ни энергией. В отличие от них она может возникать и исчезать. В указанных примерах информация в куске каменного угля или делового письма может исчезнуть, если исчезнет ее носитель, например, сгорит.

Особенность информации заключается в том, что проявляется она только при взаимодействии объектов, причем обмен информацией может совершаться не вообще между любыми объектами, а только между теми из них, которые представляют собой организованную структуру (систему). Элементами этой системы могут быть не только люди: обмен информацией может происходить в животном и растительном мире, между живой и неживой природой, людьми и устройствами. Так, информация, заключенная в куске каменного угля проявляется лишь при взаимодействии с человеком, а растение, получая информацию о свете, днем раскрывает свои лепестки, а ночью закрывает их.

Понятие «информация» обычно предполагает наличие двух объектов – «источника» информации и «приемника» (потребителя, адресата) информация.

Информация передается от источника к приемнику в материально-энергетической форме в виде сигналов (например, электрических, световых, звуковых и т. д.), распространяющихся в определенной среде.

Сигнал – физический процесс (явление), несущий сообщение (информацию) о событии или состоянии объекта наблюдения.

Информация может поступать непрерывно или дискретно, т.е. в виде последовательности отдельных сигналов. Соответственно различают непрерывную и дискретную информацию.

Информация – специфический атрибут реального мира, представляющий собой его объективное отражение в виде совокупности сигналов и проявляющийся при взаимодействии с «приемником» информации, позволяющим выделять, регистрировать эти сигналы из окружающего мира и по тому или иному критерию их идентифицировать.

Из этого определения следует, что:

- информация объективна, так как это свойство материи – отражение;
- информация проявляется в виде сигналов и лишь при взаимодействии объектов;
- одна и та же информация различными получателями может быть интерпретирована по-разному, в зависимости от «настройки» «приемника».

Человек воспринимает сигналы посредством органов чувств, которые «идентифицируются» мозгом. Поэтому, например, при наблюдении одного и того же объекта человек с лучшим зрением может получить больше информации об объекте, чем тот, у которого зрение хуже. В то же время при одинаковой остроте зрения, в случае, например, прочтения текста на иностранном языке, человек, не владеющий этим языком, вообще не получит никакой информации, так как его мозг не сможет ее «идентифицировать». Приемники информации в технике воспринимают сигналы с помощью различной измерительной и регистрирующей аппаратуры. При этом приемник, обладающий большей чувствительностью при регистрации сигналов и более совершенными алгоритмами их обработки, позволяет получить большие объемы информации.

Информация имеет определенные функции и этапы обращения в обществе. Основными из них являются:

- *познавательная*, цель которой – получение новой информации. Функция реализуется в основном через такие этапы обращения информации, как:

  - ее синтез (производство),

  - представление,

  - хранение (передача во времени),

- восприятие (потребление);

- *коммуникативная* – функция общения людей, реализуемая через такие этапы обращения информации, как:

- передача (в пространстве),

- распределение;

- *управленческая,* цель которой – формирование целесообразного поведения управляемой системы, получающей информацию. Эта функция информации неразрывно связана с познавательной и коммуникативной и реализуется через все основные этапы обращения, включая обработку.

Информацию, которую получает человек, можно считать мерой уменьшения неопределенности знаний. Если некоторое сообщение приводит к уменьшению неопределенности наших знаний, то можно говорить, что такое сообщение содержит информацию [4, с. 75]. Подход к информации как мере уменьшения неопределенности знаний позволяет количественно измерять информацию, что чрезвычайно важно для информатики.

Рассмотрим конкретный пример. Пусть у нас имеется монета, которую мы бросаем на ровную поверхность. С равной вероятностью произойдет одно из двух возможных событий – монета окажется в одном из двух положений: «орел» или «решка».

Можно говорить, что события равновероятны, если при возрастающем числе опытов количества вы «орла» и «решки» постепенно сближаются. Например, если мы бросим монету 10 раз, то «орел» может выпасть 7 раз, а решка – 3 раза, если монету бросим 100 раз, то «орел» может выпасть 60 раз, а «решка» – 40 раз, если бросим монету 1000 раз, то «орел» может выпасть 530 раз, а «решка» – 470 и т.д. В итоге при очень большой серии опытов количества выпадений «орла» и «решки» практически сравняются.

Перед броском существует неопределенность наших знаний (возможны два события), и, как упадет монета, предсказать невозможно. После броска наступает полная определенность, так как мы видим (получаем зрительное сообщение), что монета в данный момент находится в определенном положении (например, «орел»). Это сообщение приводит к уменьшению неопределенности наших знаний в два раза, так как до броска мы имели два вероятных события, а после броска – только одно, то есть в два раза меньше. Говорят, что получено количество информации, равное 1 биту.

Бит является минимальной единицей измерения количества информации, а следующей по величине единицей является байт, причем

$$1 \text{ байт} = 2^3 \text{ бит} = 8 \text{ бит}$$

В информатике система образования кратных единиц измерения количества информации отличается от принятых в большинстве наук. Компьютер оперирует числами не в десятичной системе счисления, а в двоичной, поэтому в кратных единицах измерения информации используется коэффициент 2^n.

Так, кратные байту единицы измерения количества информации вводятся следующим образом:

$$1 \text{ Кбайт} = 2^{10} \text{ байт} = 1024 \text{ байт};$$
$$1 \text{ Мбайт} = 2^{10} \text{ Кбайт} = 1024 \text{ Кбайт};$$
$$1 \text{ Гбайт} = 2^{10} \text{ Мбайт} = 1024 \text{ Мбайт}.$$

Существует несколько мер информации: синтаксическая мера, семантическая мера и прагматическая мера [2]. Синтаксическая мера – это мера количества информации, которая оперирует с обезличенной информацией, не выражающей смыслового отношения к объекту. Семантическая мера служит для измерения смыслового содержания информации, т.е. ее количества на семантическом уровне. Прагматическая мера определяет полезность информации (ценность) для достижения пользователем поставленной цели.

Для измерения информации на синтаксическом уровне вводятся два параметра: объем информации (данных) – V (объемный подход) и количество информации – I (энтропийный подход).

Объемный подход. При реализации информационных процессов информация передается в виде сообщения, представляющего собой совокупность символов какого-либо алфавита. При этом каждый новый символ в сообщении увеличивает количество информации, представленной последовательностью символов данного алфавита. Если теперь количество информации, содержащейся в сообщении из одного символа, принять за единицу, то объем информации V в любом другом сообщении будет равен количеству символов (разрядов) в этом сообщении. Так как одна и та же информация может быть представлена многими разными способами (с использованием разных алфавитов), то и единица измерения информации (данных) соответственно будет меняться.

Так, в десятичной системе счисления один разряд имеет вес, равный 10, и соответственно единицей измерения информации будет *дит* (десятичный разряд). В этом случае сообщение в виде *n*-разрядного числа имеет объем данных V = n дит.

В двоичной системе счисления один разряд имеет вес, равный 2, и соответственно единицей измерения информации будет *бит* (двоичный разряд). В этом случае сообщение в виде *n*-разрядного числа имеет объем данных V = n бит.

Создатели компьютеров отдают предпочтение именно двоичной системе счисления, потому что в техническом устройстве наиболее просто реализовать два различных состояния, например: диод открыт или закрыт, конденсатор заряжен или незаряжен и т.п. (Системы счисления рассматриваются в данном пособии в следующем разделе.)

Энтропийный подход. В теории информации и кодирования принят энтропийный подход к измерению информации. Этот подход основан на том, что факт получения информации всегда связан с уменьшением разнообразия или неопределенности (энтропии) системы. Исходя из этого, количество информации в сообщении определяется как мера уменьшения неопределенности состояния данной системы после получения сообщения. Неопределенность может быть интерпретирована в смысле того, насколько мало известно наблюдателю о данной системе. Как только наблюдатель выявил что-нибудь в физической системе, энтропия системы снизилась, так как для наблюдателя система стала более упорядоченной.

Таким образом, при энтропийном подходе под информацией понимается количественная величина исчезнувшей в ходе какого-либо процесса (испытания, измерения и т. д.) неопределенности. При этом в качестве меры неопределенности вводится энтропия H, а количество информации равно:

$$I = H_{apr} - H_{aps},$$

где H_{apr} – априорная энтропия о состоянии исследуемой системы или процесса;

H_{aps} – апостериорная энтропия.

Апостериори (от лат. *a posteriori* – из последующего) – происходящее из опыта (испытания, измерения).

Априори (от лат. *a priori* – из предшествующего) – понятие, характеризующее знание, предшествующее опыту (испытанию), и независимое от него.

В случае когда в ходе испытания имевшаяся неопределенность снята (получен конкретный результат, т. е. $H_{aps} = 0$), количество полученной информации совпадает с первоначальной энтропией $I = H_{apr}$.

Американский ученый К. Шеннон обобщил понятие меры неопределенности выбора H на случай, когда H зависит не только от числа состояний, но и от вероятностей этих состояний.

Формула Шеннона:

$$I = - \sum_{i=1}^{N} p_i \log_2 p_i ,$$

где p_i – вероятность i-го события.

При равновероятностных выборах формула Шеннона преобразуется в формулу Хартли:

$$I = \log_2 N \quad или \quad 2^I = N,$$

где N – количество равновероятных событий,

I – количество информации.

Без информации не может существовать жизнь в любой форме и не могут функционировать созданные человеком любые информационные системы. Без нее биологические и технические системы представляют груду химических элементов. Общение, коммуникации, обмен информацией присущи всем живым существам, но в особой степени – человеку. Будучи аккумулированной и обработанной с определенных позиций, информация дает новые сведения, приводит к новому знанию. Получение информации из окружающего мира, ее анализ и генерирование составляют одну из основных функций человека, отличающую его от остального животного мира.

Вопросы для самопроверки

1. Дайте определение информатике.
2. Какова общая структура информатики?
3. Определите связи информатики с другими науками.
4. Какие определения понятия «информация» вы знаете?
5. Назовите формы отражения в живой и неживой природе.
6. Какие вам известны приемники и источники информации?
7. Какие функции выполняет информация в обществе?
8. Расскажите о мерах информации.
9. В каких единицах можно измерять информацию?
10. В чем суть различных подходов к измерению информации?

Литература

1. Акулов, О.А. Информатика: базовый курс: учеб. пособие для студентов / О.А. Акулов, Н.В. Медведев. – М.: Омега-Л, 2005. – 552 с.

2. Бешенков, С.А. Информатика. Систематический курс. Учебник для 10 класса / С.А. Бешенков, Е.А. Ракитина – М.: Лаборатория Базовых Знаний, 2001. – 432 с.

3. Каймин, В.А. Информатика: учеб. – М.: ТК Велби, Изд-во Проспект, 2007. – 272 с.

4. Угринович, Н.Д. Информатика и информационные технологии. Учебник для 10-11 классов / Н.Д. Угринович. – М.: Бином. Лаборатория знаний, 2002. – 512 с.

Раздел 2. СИСТЕМЫ СЧИСЛЕНИЯ

Представление информации происходит в различных формах в процессе восприятия окружающей среды живыми организмами и человеком, в процессе обмена информацией между человеком и человеком, человеком и компьютером, компьютером и компьютером и т.д. Преобразование информации из одной формы представления (знаковой системы) в другую называется *кодированием* [4, с. 85].

Средством кодирования служит таблица соответствия знаковых систем, которая устанавливает взаимно однозначное соответствие между знаками или группами знаков двух различных знаковых систем. В процессе обмена информацией часто приходится производить операции кодирования и декодирования информации. При вводе знака алфавита в компьютер путем нажатия соответствующей клавиши на клавиатуре происходит кодирование знака, то есть преобразование его в компьютерный код. При выводе знака на экран монитора или принтер происходит обратный процесс –

декодирование, когда из компьютерного кода знак преобразуется в его графическое изображение.

В компьютере для представления информации используется двоичное кодирование, так как удалось создать надежно работающие технические устройства, которые могут со стопроцентной надежностью сохранять и распознавать не более двух состояний (цифр):

- электромагнитные реле (замкнуто/разомкнуто;
- участок поверхности магнитного носителя информации (намагничен/размагничен);
- участок поверхности лазерного диска (отражает/не отражает);
- триггер, может устойчиво находиться в одном из двух состояний, широко используется в оперативной памяти компьютера.

Поэтому все виды информации в компьютере кодируются на машинном языке, в виде логических последовательностей нулей и единиц (например, 1 – есть ток в цепи, 0 – тока в цепи нет). Цифры двоичного кода можно рассматривать как два равновероятных состояния (события). При записи двоичной цифры реализуется выбор одного из двух возможных состояний и, следовательно, она несет количество информации, равное 1 биту.

Важно, что каждая цифра машинного двоичного кода несет информацию в 1 бит. Таким образом, две цифры несут информацию в 2 бита, три цифры – в 3 бита и т.д.

Система счисления – это совокупность приемов и правил наименования и обозначения чисел, позволяющих установить взаимно однозначное соответствие между любым числом и его представлением в виде конечного числа символов. Основанием системы счисления называется количество символов, с помощью которых изображается число в данной системе счисления [1, с. 54].

В любой системе счисления выбирается алфавит, представляющий собой совокупность некоторых символов (слов или знаков), с помощью которого в результате каких-либо операций можно представить любое количество. Изображение какого-либо количества называется числом, а символы алфавита – цифрами. Символы алфавита должны быть разными и значение каждого из них должно быть известно.

В современном мире наиболее распространенной является десятичная система счисления, происхождение которой связано с пальцевым счетом. Она возникла в Индии и в XIII веке была перенесена арабами. Поэтому десятичную систему счисления стали

называть арабской, а используемые для записи чисел цифры, которыми мы теперь пользуемся, – 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, - арабскими.

С давних времен для подсчетов и вычислений применялись различные системы счисления, например, на Древнем Востоке довольно широко была распространена двенадцатеричная система. Многие предметы (ножи, вилки, тарелки и т. д.) и сейчас считают дюжинами. Число месяцев в году – двенадцать. Эта система счисления сохранилась в английской системе мер (например, 1 фут = 12 дюймов) и в денежной системе (1 шиллинг = 12 пенсов). В Древнем Вавилоне существовала весьма сложная шестидесятеричная система. Она, как и двенадцатеричная в какой-то мере сохранилась и до наших дней (например, в системе измерения времени: 1 ч = 60 мин, 1 мин = 60 с, аналогично в системе измерения углов: 1^o = 60 мин, 1 мин = 60 с).

Первые цифры (знаки для обозначения цифр) появились у египтян и вавилонян. У ряда народов (древние греки, сирийцы, финикияне) цифрами служат буквы алфавита. Аналогичная система до XVI века применялась и в России. В Средние века в Европе пользовались системой римских цифр, которую сейчас часто применяют для обозначения глав, частей, разделов в различного рода документах, книгах, для обозначения месяцев и т. д.

Все системы счисления можно разделить на позиционные и непозиционные.

Непозиционные системы счисления – система, в которой символы, обозначающие то или иное количество, не меняют своего значения в зависимости от своего местоположения (позиции) в изображении числа.

Запись числа A в непозиционной системе счисления D может быть представлена выражением:

$$A_D = D_1 + D_2 + \ldots + D_N = \sum_{i=1}^{N} D_i$$

где A_D – запись числа A в системе счисления D; D_i – символы системы.

Непозиционной системой счисления является самая простая система с одним символом (палочкой). Для изображения какого-либо числа в этой системе надо записать количество палочек, равное данному числу. Например, запись числа 12 в такой системе счисления будет иметь вид: 111111111111, где каждая «палочка» обозначена символом 1. Эта системе не эффективна, так как форма записи очень громоздка.

К непозиционной системе счисления относится и римская, символы алфавита которой и обозначаемой ими количество представлены ниже.

Римские цифры	I	V	X	L	C	D	M
Значение (обозначаемое количество)	1	5	10	50	100	500	1000

Запись чисел в этой системе счисления осуществляется по следующим правилам:

1) если цифра слева меньше, чем цифра справа, то левая цифра вычитается из правой (IV: 1 < 5, следовательно, 5 - 1 = 4, XL: 10 < 50, следовательно, 50 – 10 = 40);

2) если цифра справа меньше или равна цифре слева, то эти цифры складываются (VI: 5 + 1 = 6, VIII: 5 + 1 + 1 + 1 = 8, XX: 10 + 10 = 20).

Так, число 1964 в римской систем счисления имеет вид MCMLXIV (M – 1000, CM – 900, LX – 60, IV – 4), здесь «девятьсот» получается посредством вычитания из «тысячи» числа «сто», «шестьдесят» - посредством сложения «пятидесяти» и «десяти», «четыре» - посредством вычитания из «пяти» «единицы».

В общем случае непозиционные системы счисления характеризуется сложными способами записи чисел и правилами выполнения арифметических операций. В настоящее время все наиболее распространенные системы счисления относятся к разряду позиционных.

Позиционной системой счисления называют систему счисления, в которой значения цифры определяется ее местоположением (позицией) в изображении числа.

Упорядоченный набор символов (цифр) $\{a_0, a_1, \ldots, a_n\}$, используемый для представления любых чисел в заданной позиционной системе счисления, называют ее *алфавитом*, число символов (цифр) алфавита $p = n + 1$ - ее *основанием*, а саму систему счисления называют *p-ичной. Основание* позиционной системы счисления – количество различных цифр, используемых для изображения чисел в данной системе счисления.

Самой привычной системой для нас является десятичная система счисления. Ее алфавит – $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, а основание p = 10, т. е. в этой системе для записи любых чисел используется только 10 различных символов (цифр). Эти цифры введены для обозначения первых 10 последовательных чисел, а все последующие числа, начиная с 10 и т. д., обозначаются уже без использования новых цифр. Десятичная система счисления основана на том, что десять единиц каждого разряда объединяются в одну единицу соседнего старшего разряда, поэтому каждый разряд имеет вес, равный степени

10.Следовательно, значение одной и той же цифры определяется ее местоположением в изображении числа, характеризуемым степенью числа 10.

В позиционной системе счисления число (записанное знаками A_n, A_{n-1}, A_{n-2}, …) может быть представлено как сумма степеней основания системы счисления (обозначим его B), умноженных на коэффициенты, которыми являются A_n, A_{n-1} ...:

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

Например:

$$45,18_{10} = 4*10^1 + 5*10^0 + 1*10^{-1} + 8*10^{-2}$$

Десятичный индекс внизу числа указывает основание системы счисления.

При работе с компьютерами приходится параллельно использовать несколько позиционных систем счисления (чаще всего двоичную, восьмеричную, десятичную и шестнадцатеричную).

Десятичная	Двоичная	Восьмеричная	Шестнадцатеричная
0	0	0	0
1	1	1	1
2	**10**	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	**7**	7
8	1000	**10**	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	**10**

Арифметические действия над числами в любой позиционной системе счисления производятся по тем же правилам, что и в десятичной системе.

В основе сложения чисел в двоичной системе счисления лежит таблица сложения одноразрядных двоичных чисел:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$

В основе вычитания лежит таблица вычитания одноразрядных двоичных чисел. При вычитании из меньшего числа (0) большего (1) производится заем из старшего разряда. В таблице заем обозначен 1 с чертой:

$$0 - 0 = 0$$

$$0 - 1 = {}^{-}11$$

$$1 - 1 = 1$$

$$1 - 1 = 0$$

В основе умножения лежит таблица умножения одноразрядных двоичных чисел:

$$0 * 0 = 0$$

$$0 * 1 = 0$$

$$1 * 0 = 0$$

$$1 * 1 = 1$$

Операция деления выполняется по алгоритму, подобному алгоритму выполнения деления в десятичной системе счисления. Например, разделим число 110_2 на 11_2 :

$$\begin{array}{l|l} 110_2 & 11_2 \\ \underline{11} & 10_2 \\ 0 & \end{array}$$

Аналогично можно выполнять арифметические действия в восьмеричной и шестнадцатеричной системах счисления. Необходимо только помнить, что величина переноса в следующий разряд при сложении и заем из старшего разряда при вычитании определяется величиной основания системы счисления.

Для проведения арифметических операций над числами, выраженными в различных системах счисления, необходимо перевести их в одну и ту же систему. При переводе чисел из одной системы счисления в другую, следует придерживаться следующих правил:

1. Перевод чисел в **двоичную** систему счисления:

1.1. Из *восьмеричной* системы счисления:

Нужно каждую цифру восьмеричного числа записать триадой (тройкой) цифр двоичной системы счисления.

Например: $274_8 = 010\ 111\ 100_2$

1.2. Из *десятичной* системы счисления:

Нужно делить число нацело на 2, пока последнее полученное неполное частное не станет равным нулю, и переписать остатки в обратном порядке.

Например: 81 : 2 = 40 (1)

40 : 2 = 20 (0)

20 : 2 = 10 (0)

10 : 2 = 5 (0)

5 : 2 = 2 (1)

2 : 2 = 1 (0)

1 : 2 = 0 (1)

Ответ: $81_{10} = 1010001_2$

Для перевода дробной части (или числа, у которого «0» целых) умножаем ее на 2, целая часть произведения – первая цифра числа в двоичной системе; затем, отбрасывая у результата целую часть, вновь умножаем на 2 и т.д. Следует заметить, что конечная десятичная дробь при этом вполне может стать бесконечной (периодической) двоичной. Например: 0,73 · 2 = 1,46 (целая часть 1);

0,46 · 2 = 0,92 (целая часть 0);

0,92 · 2 = 1,84 (целая часть 1);

0,84 · 2 = 1,68 (целая часть 1) и т.д.; в итоге

$0{,}73_{10} = 0{,}1011\ldots_2$

1.3. Из *шестнадцатеричной* системы счисления:

Нужно каждую цифру шестнадцатеричного числа записать тетрадой (четверкой) чисел двоичной системы счисления.

Например: $9C5_{16} = 1001\ 1100\ 0101_2$

2. Перевод чисел в **восьмеричную** систему счисления:

2.1. Из *двоичной* системы счисления:

Нужно разбить число влево и вправо от запятой на триады (тройки) цифр и каждую из них представить восьмеричным числом.

Например: 110111, $101_2 = 110\ \ 111\ \ ,\ \ 101_2 = 67{,}5_8$

2.2. Из *десятичной* системы счисления:

Нужно делить число нацело на 8, пока последнее полученное неполное частное не станет равным нулю, и переписать остатки в обратном порядке.

Например: 265 : 8 = 33 (1)

33 : 8 = 4 (1)

4 : 8 = 0 (4)

Ответ: $265_{10} = 411_8$

2.3. Из *шестнадцатеричной* системы счисления:

Нужно сначала представить число в двоичной системе счисления, а затем в восьмеричной.

Например: C6, $8_{16} = 1100\ 0110$, $1000_2 = 011\ 000\ 110$, $100_2 = 306{,}4_8$

3. Перевод чисел из любой системы счисления в **десятичную** систему счисления:

Нужно представить число в виде суммы произведений коэффициентов и степеней основания системы счисления.

6 5 4 3 2 1 0

Например: $1000111_2 = 1\ 0\ 0\ 0\ 1\ 1\ 1_2 = 1*2^6 + 0*2^5 + 0*2^4 + 0*2^3 + 1*2^2 + 1*2^1 + 1*2^0 = 32 + 0 + 0 + 0 + 4 + 2 + 1 = 35_{10}$

4. Перевод чисел в **шестнадцатеричную** систему счисления:

4.1. Из *двоичной* системы счисления:

Нужно разбить число влево и вправо от запятой на тетрады (четверки) цифр и каждую из них представить шестнадцатеричным числом.

Например: 1111011, $111_2 = 0111\ 1011$, $1110_2 = 7B{,}E_{16}$

4.2. Из *десятичной* системы счисления:

Нужно делить число нацело на 16, пока последнее полученное неполное частное не станет равным нулю, и переписать остатки в обратном порядке.

Например: 1756 : 16 = 109 (12 = C)

109 : 16 = 6 (13 = D)

6 : 16 = 0 (6)

Ответ: $1756_{10} = 6DC_{16}$

4.3. Из *восьмеричной* системы счисления:

Нужно сначала представить число в двоичной системе счисления, а затем в шестнадцатеричной.

Например: 672, $5_8 = 110\ 111\ 010$, $101_2 = 0001\ 1011\ 1010$, $1010_2 = 1BA{,}A_{16}$

Итак, сформулируем общие правила перевода чисел из одной системы счисления в другую:

1. При переводе чисел из десятичной системы счисления в систему с основанием p *> 1 исходное число делится нацело на* p*, после чего запоминается остаток от деления. Полученное частное вновь делится на* p*, остаток запоминается. Процедура продолжается до тех пор, пока частное не станет равным нулю. Остатки от деления, записанные в системе счисления с основанием* p *справа налево, образуют результат.*

2. При переводе чисел из системы счисления с основанием p *в десятичную систему счисления необходимо пронумеровать разряды исходного числа справа налево, начиная с нулевого. Затем вычислить сумму произведений значений разрядов на основание системы счисления в степени, равной соответствующему номеру разряда.*

3. Перевод в двоичную систему восьмеричных и шестнадцатеричных чисел может быть произведен простой заменой каждой цифры исходного числа на соответствующее ей двоичное число.

4. Для перевода двоичных чисел в восьмеричную (шестнадцатеричную) систему счисления необходимо разбить цифры исходного числа на группы по три (четыре) справа налево. При этом последняя группа может содержать меньше трех (четырех) цифр. Затем каждой группе цифр ставится в соответствие ее восьмеричный (шестнадцатеричный) эквивалент.

Вопросы для самопроверки

1. Что такое система счисления?
2. В чем отличие позиционной системы счисления от непозиционной?
3. Что называется основанием системы счисления?
4. Как выполняются арифметические действия в позиционных системах счисления?
5. Сформулируйте правила выполнения переводов чисел из одной системы счисления в другую.

Литература

1. Акулов, О.А. Информатика: базовый курс: учеб. пособие для студентов / О.А. Акулов, Н.В. Медведев. – М.: Омега-Л, 2005. – 552 с.

2. Бешенков, С.А. Информатика. Систематический курс. Учебник для 10 класса / С.А. Бешенков, Е.А. Ракитина – М.: Лаборатория Базовых Знаний, 2001. – 432 с.

3. Могилев, А.В. Информатика: учеб. пособие для студ. высш. пед. учеб. заведений / А.В. Могилев, Е.К. Хеннер, Н.И. Пак; под ред. А.В. Могилева. – М.: Издательский центр «Академия», 2006. – 336 с.

4. Угринович, Н.Д. Информатика и информационные технологии. Учебник для 10-11 классов / Н.Д. Угринович. – М.: Бином. Лаборатория знаний, 2002. – 512 с.

Раздел 3. ОСНОВЫ ПРОГРАММИРОВАНИЯ

3.1. Понятие алгоритма

Понятие алгоритма является одним из основных понятий современной информатики. Термин «алгоритм» (алгорифм) происходит от имени среднеазиатского ученого IX века аль-Хорезми, который разработал правила выполнения четырех арифметических действий в десятичной системе счисления.

Вплоть до 30-х годов прошлого столетия понятие алгоритма носило сугубо интуитивный характер и имело скорее методологическое, чем математическое значение. Общей теории алгоритмов фактически не существовало, а под алгоритмом понимали *конечную совокупность точно сформулированных правил, которые позволяли решать те или иные классы задач* Основными свойствами такого «интуитивного» понятия алгоритма являются [1, с. 177]:

1. Массовость алгоритма. Подразумевается, что алгоритм позволяет решать не одну конкретную задачу, а некоторый класс задач данного типа. В простейшем случае массовость обеспечивает возможность изменения исходных данных в определенных пределах.

2. Детерминированность алгоритма. Процесс применения правил к исходным данным (путь решения задачи) однозначно определен.

3. Результативность алгоритма. На каждом шаге процесса применения правил известно, что считать результатом этого процесса, а сам процесс должен прекратиться за конечное число шагов.

В течение длительного времени пока дело касалось задач, имеющих решение, математиков устраивало такое определение алгоритма. Совсем другое дело, когда задача или класс задач могут и не иметь решения. В этом случае требуется строго формализованное понятие алгоритма, чтобы иметь возможность доказать его отсутствие.

В 20-х годах XX века задача такого определения понятия алгоритма стала одной из центральных математических проблем. Решение ее было получено в середине 30-х годов в работах известных математиков Д. Гильберта, К. Геделя, А. Черча, С. Клини, Э. Поста и А. Тьюринга в двух эквивалентных формулировках: на основе особого класса функций, получивших название рекурсивных функций, и на основе абстрактных автоматов.

Таким образом, первоначально теория алгоритмов возникла в связи с внутренними потребностями теоретической математики. Математическая логика, основания математики, алгебра, геометрия и анализ остаются и сегодня одной из основных областей приложения теории алгоритмов. Кроме того, теория алгоритмов оказалась тесно связанной и с рядом областей лингвистики, экономики, физиологии мозга и психологии, философии и естествознания. Примером одной из задач этой области может служить описание алгоритмов, реализуемых человеком в процессе умственной деятельности.

Вместе с тем в 40-х годах прошлого века в связи с созданием электронных вычислительных и управляющих машин возникла область теории алгоритмов, тесно взаимодействующая с информатикой. Появление ЭВМ способствовало развитию разделов этой теории, имеющих ярко выраженную прикладную направленность.

В общем случае при составлении алгоритма конкретной задачи актуальное значение имеет такое представление алгоритма, которое позволяет наиболее быстро

реализовать его механизированным путем, и в частности с помощью ЭВМ. При этом для решения задачи с помощью ЭВМ ее необходимо запрограммировать, т.е. представить алгоритм решения задачи в виде последовательности команд, которые может выполнять машина. Однако процесс записи алгоритма в виде последовательности машинных команд очень длительный и трудоемкий. Его также можно автоматизировать, если использовать для записи алгоритмов *алгоритмические языки*, представляющие собой набор символов и терминов, связанных синтаксической структурой. С их помощью можно по определенным правилам описывать алгоритмы решения задач. Алгоритмы, записанные в алгоритмическом языке, автоматически самой ЭВМ с помощью специальной программы-транслятора могут быть переведены в машинные программы для конкретной ЭВМ.

Итак, *алгоритм – конечный набор правил или команд (указаний), позволяющий исполнителю решать любую конкретную задачу из некоторого класса однотипных задач.*

Исполнителем может человек, группа людей, станок, компьютер и др. С учетом особенностей исполнителя составленный алгоритм может быть представлен различными способами: с помощью графического или словесного описания, в виде таблицы, последовательностью формул, записанных на алгоритмическом языке (языке программирования) и т.д.

Алгоритм, записанный на «понятном» компьютеру языке программирования, называется *программой*.

К основным типам алгоритмических структур относят [2, с. 150-156]:

- линейный алгоритм;
- алгоритмическая структура «ветвление»;
- алгоритмическая структура «выбор»;
- алгоритмическая структура «цикл».

Алгоритм, в котором команды выполняются последовательно одна за другой, называется линейным алгоритмом.

В алгоритмической структуре «ветвление» та или иная серия команд выполняется в зависимости от истинности условия.

В алгоритмической структуре «выбор» выполняется одна из нескольких последовательностей команд при истинности соответствующего условия.

В алгоритмической структуре «цикл» серия команд (тело цикла) выполняется многократно.

Литература

1. Акулов, О.А. Информатика: базовый курс: учеб. пособие для студентов / О.А. Акулов, Н.В. Медведев. – М.: Омега-Л, 2005. – 552 с.

2. Угринович, Н.Д. Информатика и информационные технологии. Учебник для 10-11 классов / Н.Д. Угринович. – М.: Бином. Лаборатория знаний, 2002. – 512 с.

3.2. Языки программирования

Под *языком программирования* будем понимать совокупность средств и правил представления алгоритма в виде, приемлемом для компьютера. Существует разделение всех языков на две большие группы – языки высокого и низкого уровней.

Языком самого высокого уровня считается человеческий язык, и когда компьютер станет его легко понимать, то он вплотную приблизится к человеку. Языком самого низкого уровня считается язык так называемых машинных кодов. Все остальные алгоритмические языки лежат где-то посредине.

С помощью языков низкого уровня создаются очень эффективные и компактные программы, так как разработчик получает доступ ко всем возможностям процессора. С другой стороны, при этом требуется очень хорошо понимать устройство компьютера, затрудняется отладка больших приложений, а результирующая программа не может быть перенесена на компьютер с другим типом процессора. Подобные языки обычно применяют для написания небольших системных приложений, драйверов устройств, модулей стыковки с нестандартным оборудованием, когда важнейшими требованиями становятся компактность, быстродействие и возможность прямого доступа к аппаратным ресурсам.

Особенности конкретных компьютерных архитектур в языках высокого уровня не учитываются, поэтому создаваемые программы на уровне исходных текстов легко переносимы на другие платформы, для которых создан транслятор этого языка. Разрабатывать программы на языках высокого уровня с помощью понятных и мощных команд значительно проще, а ошибок при создании программ допускается гораздо меньше.

Языки программирования принято делить на пять поколений [1, с. 572]. В первое поколение входят языки, созданные в начале 50-х годов, когда первые

компьютеры только появились на свет. Это был первый язык ассемблера, созданный по принципу «одна инструкция – одна строка». Программы представляли собой длинные логические последовательности нулей и единиц.

Расцвет второго поколения языков программирования пришелся на конец 50-х – начало 60-х годов. Тогда был разработан символический ассемблер, в котором появилось понятие переменной. Он стал первым полноценным языком программирования. Благодаря его возникновению заметно выросли скорость разработки и надежность программ.

Появление третьего поколения языков программирования принято относить к 60-м годам. В это время родились универсальные языки высокого уровня, с их помощью удается решать задачи из любых областей. Такие качества новых языков, как относительная простота, независимость от конкретного компьютера и возможность использования мощных синтаксических конструкций, позволили резко повысить производительность труда программистов. Понятная большинству пользователей структура этих языков привела к написанию небольших программ (как правило, инженерного или экономического характера) значительное число специалистов из некомпьютерных областей. Подавляющее большинство языков этого поколения успешно применяется и сегодня.

С начала 70-х годов по настоящее время продолжается период языков четвертого поколения. Эти языки предназначены для реализации крупных проектов, повышения их надежности и скорости создания. Они обычно ориентированы на специализированные области применения, где хороших результатов можно добиться, используя не универсальные, а проблемно-ориентированные языки, оперирующие понятиями узкой предметной области. Как правило, в эти языки встраиваются мощные операторы, позволяющие одной строкой описать такую функциональность, для реализации которой на языках младших поколений потребовались бы тысячи строк исходного кода.

Рождение языков пятого поколения произошло в середине 90-х годов. К ним относятся также системы автоматического создания прикладных программ с помощью визуальных средств разработки, без знания программирования. Главная идея, которая закладывается в эти языки, – возможность автоматического формирования результирующего текста на универсальных языках программирования. Инструкции же вводятся в компьютер в максимально наглядном виде с помощью методов, наиболее удобных для человека, не знакомого с программированием.

Примерами языков высокого уровня, широко используемых во всем мире являются FORTRAN (Фортран), COBOL (Кобол), Algol (Алгол), Pascal (Паскаль), Basic (Бейсик), C (Си), C++ (Си++), Java (Джава, Ява).

Первое место по популярности в мире занимает язык Basic. Разработан первый Basic в 1964 г. сотрудниками Дартмутского колледжа Дж. Кемени и Т. Курцем. Интересно происхождение названия языка. В 19 веке один английский миссионер выделил из английского языка триста наиболее употребительных слов, назвал их Basic English и стал обучать туземцев. Опыт оказался весьма успешным, и контакты с аборигенами значительно упростились. Создатели языка Basic стремились достигнуть того же эффекта – облегчить понимание между «туземцами» – начинающими программистами, и компьютерами.

Идея оказалась удачной, и на десятилетия язык Basic стал основным в деле вовлечения в программирование новых и новых адептов. Большое достоинство Бейсика, из-за которого его изучение продолжается и поныне – это возможность создавать диалоговые программы. За прошедшие годы было создано несколько версий языка – GW-Basic, MSX-Basic, Turbo Basic, Quick Basic, и, наконец, мощный Visual Basic.

Программа, подготовленная на языке программирования, проходит этап трансляции, когда происходит преобразование исходного кода программы в объектный код, который далее пригоден к обработке редактором связей. Редактор связей – специальная программа, обеспечивающая построение загрузочного модуля, пригодного к выполнению.

Трансляция может выполняться с использованием средств компиляторов или интерпретаторов. Компиляторы транслируют всю программу, но без ее выполнения. Интерпретаторы, в отличие от компиляторов, выполняют пооператорную обработку и выполнение программы.

Существуют специальные программы, предназначенные для трассировки и анализа выполнения других программ, так называемые отладчики. Лучшие отладчики позволяют осуществить трассировку (отслеживание выполнения программы в пооператорном варианте), идентификацию места и вида ошибок в программе, «наблюдение» за изменением значений переменных, выражений и т. п. Для отладки и тестирования правильности работы программ создается база данных контрольного примера.

Системы программирования включают:

- компилятор;
- интегрированную среду разработчика программ;
- отладчик;
- средства оптимизации кода программ;
- набор библиотек (возможно с исходными текстами программ);
- редактор связей;
- сервисные средства (утилиты) для работы с библиотеками, текстовыми и двоичными файлами;
- справочные системы;
- документатор исходного кода программы;
- систему поддержки и управления проектом программного комплекса.

Инструментальная среда пользователя представлена специальными средствами, встроенными в пакеты прикладных программ, такими, как:

- библиотека функций, процедур, объектов и методов обработки;
- макрокоманды;
- клавишные макросы;
- языковые макросы;
- программные модули-вставки;
- конструкторы экранных форм и отчетов;
- генераторы приложений;
- языки запросов высокого уровня;
- языки манипулирования данными;
- конструкторы меню и многое другое.

Дальнейшим развитием локальных средств разработки программ, которые объединяют набор средств для комплексного их применения на всех технологических этапах создания программ, являются интегрированные программные среды разработчиков. Основное назначение инструментария данного вида – повышение производительности труда программистов, автоматизация создания кодов программ, обеспечивающих интерфейс пользователя графического типа, разработка приложений для архитектуры клиент-сервер, запросов и отчетов.

Литература

1. Информатика. Базовый курс / Симонович С.В. [и др.] – СПб: Питер, 2005. – 640 с.

2. Сафронов И.К. Бейсик в задачах и примерах / И.К. Сафронов. – СПб.: БХВ-Петербург, 2003. – 224 с.

3. Информатика: Учебник / Под ред. Н.В. Макаровой. – М.: Финансы и статистика, 2005. – 768 с.

3.3. Программирование на языке Quick Basic

Рассмотрим подробно язык программирования Quick Basic.

Алфавит данного языка содержит в себе следующие символы:

- Заглавные буквы латинского алфавита. При наборе программы, впрочем, нет нужды следить за тем, чтобы буквы были заглавными. Интерпретатор сам изменит строчные буквы на заглавные.
- Арабские цифры.
- Разделители: , (запятая), ; (точка с запятой), . (точка), : (двоеточие), ` (апостроф), “ (кавычки), ( ) (скобки), символ <Пробел>.
- Знаки арифметических операций: + (сложение), - (вычитание), * (умножение), / (деление), ^ (возведение в степень).
- Знаки операций отношений: > (больше), < (меньше), = (равно), <> (не равно), >= (больше либо равно), <= (меньше либо равно).

Для создания программ довольно часто требуются переменные, то есть такие области оперативной памяти, которые имеют имя, данное программистом, и значения, которые могут меняться. Имя переменной в ходе выполнения программы постоянно, а значение может меняться многократно. Существуют ограничения на имена переменных:

- имя переменной должно состоять не боле чем из сорока символов;
- в качестве символов можно использовать только латинские буквы и цифры;
- имя переменной не может начинаться с цифры;
- запрещены в именах символы точки, запятой, звездочки, вопросительного знака, пробела.

Переменные различаются по типу хранимой в них информации. Два наиболее крупных типа – числовой (для хранения чисел) и строковый (для хранения символов и строк). Во втором случае к имени переменной добавляется обязательный символ $.

Чтобы вычислять элементарные арифметические выражения, необходимо представить выражение в понятном для компьютера виде, а именно:

- в отличие от арифметики, выражение должно быть записано в одну строку без числителей и знаменателей;
- для записи арифметических действий допустимо использовать только перечисленные ниже знаки:

+ (сложение);

– (вычитание);

* (умножение);

/ (деление);

^ (возведение в степень);

( ) (скобки);

- недопустим пропуск знака умножения между коэффициентом и переменной, как это возможно в алгебре;
- дробная часть отделяется от целой точкой, а не запятой;
- допустимо опускать в записи десятичной дроби ноль, стоящий перед точкой (вместо 0.561 можно .561).

Чтобы компьютер вычислил выражение правильно, необходимо помнить о приоритете выполнения действий:

- сначала выполняются действия в скобках;
- далее вычисляются функции, если они есть;
- затем выполняется возведение в степень;
- потом умножение и деление;
- в последнюю очередь – сложение и вычитание.

Действия одинаковой очередности выполняются слева направо.

При решении различных задач с помощью компьютера бывает необходимо вычислить логарифм или модуль числа, синус или тангенс угла и т.п. Вычисления часто употребляемых функций осуществляются посредством подпрограмм, называемых стандартными функциями, которые заранее запрограммированы и встроены в транслятор языка.

Числовые функции

Функция	**Описание**
ABS (X)	Возвращает абсолютное значение (модуль) аргумента
ATN (X)	Арктангенс (в радианах)
CDBL (X)	Переводит числовое выражение в значение с двойной

	точностью
CINT (X)	Округление
CLNG (X)	Округление числового выражения до длинного (4 байта) целого значения
COS (X)	Косинус
CSNG (X)	Переводит числовое выражение в значение с одинарной точностью
EXP (X)	Экспонента e^x
FIX (X)	Округление выражения с плавающей запятой до его целой части
INT (X)	Возвращает наибольшее целое, меньшее либо равное числовому выражению
LOG (X)	Натуральный логарифм числового выражения
RND (X)	Случайное число одинарной точности между 0 и 1
SCN (X)	Возвращает значение знака числового выражения (1, если выражение положительное; 0, если равно 0 и – 1, если отрицательно)
SIN (X)	Синус
SQR (X)	Корень квадратный
TAN (X)	Тангенс

Приведем список основных операторов языка Quick Basic.

Операторы выбора и перехода

Оператор	**Описание**
GOTO	Безусловный переход на метку
IF … THEN … ELSE	Переход в зависимости от истинности или ложности проверяемого условия
SELECT CASE	Переход в зависимости от значения выражения

Операторы и функции для работы с файлами

Оператор, функция	**Описание**
CLOSE	Закрывает один или несколько файлов или устройств

FIELD	Отводит место под переменные в буфере файлов прямого доступа
FILEATTR	Возвращает информацию об открытом файле
GET	Считывает из файла в буфер прямого доступа или в переменную
INPUT #	Считывает данные из файла
IOCTL	Посылает управляющую строку драйверу устройства
LINE INPUT #	Считывает строку до 255 символов с клавиатуры или из файла
LOCK	Ограничивает или закрывает доступ к файлу при работе в сети
OPEN	Открывает файл или устройства
PRINT #	Записывает данные в файл
PRINT # USING	Записывает отформатированные данные в файл
PUT	Записывает содержимое переменной или буфера прямого доступа в файл
RESET	Закрывает все открытые файлы и устройства
SEEK	Устанавливает позицию файла для последующей записи или считывания
UNLOCK	Снимает ограничения, наложенные последним оператором LOCK
WRITE #	Записывает данные в последовательный файл

Переменные

Конструкция	**Описание**
CLEAR	Закрывает все файлы, освобождает буферы файлов, очищает все общие переменные, устанавливает числовые переменные и массивы в ноль, устанавливает строковые переменные в ноль и инициализирует стек. Кроме того, CLEAR может изменять размер стека
CONST	Описывает одну или несколько символьных переменных
DATA	Указывает значение данных для последующего считывания у оператора READ

INPUT	Считывает входные данные с клавиатуры или из файлов
LET	Присваивает значение выражения переменной
RANDOMIZE	Инициализирует генератор случайных чисел
READ	Считывает данные, указанные в операторе DATA
RESTORE	Восстанавливает считанные значения в операторе DATA
SWAP	Обменивает значения двух переменных

Массивы

Конструкция	**Описание**
DIM	Оператор объявления массива
ERASE	Для статических массивов каждому элементу присваивается ноль. Для стокового – определяются строки нулевой длины. Для динамического – освобождает память, используемую массивом
OPTION BASE	Устанавливает нижнюю границу индекса массива
REDIM	Описывает или изменяет размер динамического массива

Циклы

Оператор	**Описание**
DO … LOOP	Повторяет блок операторов, пока условие верно, или пока оно не станет верным
END	Заканчивает программу, процедуру или блок
FOR … NEXT	Цикл с параметром, заранее известным числом повторений
WHILE … WEND	Выполняет блок операторов, пока указанное условие верно

Подпрограммы и функции

Оператор	**Описание**
CALL	Передает управление в процедуру типа SUB
DECLARE	Описывает процедуру типа FUNCTION или SUB
DEF FN	Определяет функцию
FUNCTION	Определяет процедуру FUNCTION
GOSUB	Переходит в подпрограмму и возвращается из нее
ON GOSUB	Выполняет переход к одной из нескольких подпрограмм в

	зависимости от выражения
RETURN	Возвращает из подпрограммы в основную программу
SUB	Определяет процедуру SUB

Пример 1. В стене существует квадратное отверстие N x N см. Имеется кирпич с измерениями A, B и C. Определить, пройдет он в отверстие или нет, если подавать его можно только параллельно стенкам отверстия.

Решение. Понятно, что кирпич пройдет в отверстие только в случае, если хотя бы два его измерения меньше N. Программа будет выглядеть следующим образом:

```
INPUT «Введите сторону отверстия N»; N
1: INPUT «Введите стороны кирпича A, B и C»; A, B, C
IF A<N AND B<N OR A<N AND C<N OR B<N AND C<N THEN PRINT
 «Кирпич проходит в отверстие» ELSE PRINT «Кирпич не проходит в отверстие»
INPUT «Рассмотрим еще один кирпич? 1 – да, 0 –нет»; X
IF X=1 THEN GOTO 1 ELSE PRINT «Спасибо за работу!»
```

Объясняется данная программа следующим образом. Если A и B меньше N, или A и C меньше N, или B и C меньше N, тогда печатать «Кирпич проходит в отверстие», иначе печатать «Кирпич не проходит в отверстие». Кроме того, в данной программе применен достаточно простой, но эффективный ход, управляющий совместной работой пользователя и программы. Чтобы каждый раз не запускать программу (если пользователь хочет рассмотреть несколько вариантов исходных данных), применен запрос с клавиатуры вариантов продолжения: либо рассматриваем еще один кирпич, либо – конец программы. Там же использовано применение безусловного перехода в условном операторе.

Пример 2. Напечатать значения $y = sin\ x$ в интервале $[-30^\circ; 30^\circ]$ с шагом 5°.

Решение. Начальное, конечное значения параметра и шаг указаны в задании. Примем параметр как *x.*

```
FOR X=-30 TO 30 STEP 5
Y=SIN (X*3.14/180)
PRINT «SIN («; X; »)=»; Y
NEXT X
```

Пример 3. Вычислить значение функции:

$$y = \begin{cases} x^3, & x < -1 \\ x^2, & -1 \le x \le 1 \end{cases}$$

x^3, $x > 1$

Решение:

```
INPUT «Введите значение X»;  X
IF X < -1 OR X >= 1 THEN Y = X^3  ELSE Y = X^ 2
PRINT «Y (X) = »; Y
```

Пример 4. Вычислить сумму всех четных чисел от 1 до 100 включительно.

Решение:

```
S = 0  `Обнуление переменной, где будет накапливаться сумма`
FOR I = 2 TO 100  STEP 2
S = S + I
NEXT I
PRINT «Сумма четных чисел от 1 до 100 равна»; S
```

Задачи для самостоятельного решения

1. Осуществите запрос трех целых различных чисел с клавиатуры. Выведите на экран наибольшее и наименьшее.
2. Напишите программу, выводящую на экран степени числа 2 от 2 до 10 включительно.
3. Напишите программу для нахождения суммы пяти произвольных чисел, вводимых с клавиатуры.
4. Напишите программу вычисления среднего балла при поступлении в вуз по результатам четырех экзаменов, которые вводятся с клавиатуры.
5. Вычислить количество прожитых составителем программы дней. Учесть, что в високосном году 366 дней.
6. Вычислить, какая сумма будет лежать на вкладе в банке через 5, 10, 15, 20 лет, если положить сегодня 1000 рублей под 3 % годовых?

Литература

1. Информатика. Базовый курс / Симонович С.В. [и др.] – СПб: Питер, 2005. – 640 с.
2. Сафронов И.К. Бейсик в задачах и примерах / И.К. Сафронов. – СПб.: БХВ-Петербург, 2003. – 224 с.

3.4. Информационные массивы

Массивом называется упорядоченная совокупность элементов одного типа. Массивы бывают одномерные, двумерные и многомерные. Массив, в котором каждый элемент имеет один порядковый номер, называется одномерным. Например, список фамилий студентов вашей группы – это одномерный массив элементов символьного типа, а численные данные о среднесуточной температуре за месяц – одномерный массив элементов численного типа.

Если известно, что в программе предстоит работать с большим объемом данных, то следует этот массив в программе объявить с помощью специального оператора DIM. Например, DIM MASS(15). Это значит, что в программе определен одномерный массив с именем MASS, содержащий 15 элементов.

Массив всегда имеет:

- *имя,* которое ему дает программист;
- *тип,* который определяется именем (числовой – имя без знака $, символьный – имя со знаком $);
- *размер,* т.е. количество составляющих его элементов;
- *сквозную последовательную индексацию,* составляющих его элементов;
- *значение* каждого элемента массива.

Массив нельзя объявлять дважды, поэтому следует объявлять массивы в начальных строках программы и не возвращаться в эти строки с помощью оператора GOTO.

Ввод элементов массива может осуществляться несколькими способами:

1. DIM A(5)

   A(0)=4: A(1)=-2.5: A(2)=40: A(3)=7: A(4)=-5: A(5)=1

Этот способ ввода удобен, когда массив небольшой.

2. Ввод элементов массива с клавиатуры

```
DIM A(9)
FOR I=0 TO 9
INPUT A(I)
NEXT I
```

3. DIM B(6)

```
DATA 0, -2, 1, 5, 6, 7, 23
FOR N=0 TO 6
READ B(N)
NEXT N
```

Основные приемы при решении задач с применением массивов

Пример 1. Дан массив из 10 элементов. Найти сумму элементов массива.

Решение:

```
DIM A(9)
S=0
FOR N=0 TO 9
S=S+A(N)
NEXT N
PRINT «СУММА =»; S
```

Пример 2. Найти минимальный элемент массива и индекс этого минимального элемента.

Решение:

```
DIM L(9)
MIN=L(0) : K=0
FOR I=1 TO 9
IF MIN>L(I) THEN MIN=L(I) : K=I
NEXT I
PRINT «МИНИМУМ»; MIN
PRINT «НОМЕР МИНИМУМА»; K
```

Пример 3. Даны два массива A(9), B(9). Получить массив, каждый элемент которого равен сумме соответствующих элементов данных массивов.

Решение:

```
DIM A(9), B(9), C(9)
FOR I=0 TO 9
C(I)=A(I) +B(I) : PRINT C(I)
NEXT I
```

Двумерный массив состоит из элементов, имеющих два порядковых номера. Один номер – это номер строки, а второй номер – номер столбца. Таким образом, двумерный массив можно представить в виде матрицы, состоящей из определенного количества строк и столбцов. Например, в массиве MASS (4,5) 4 строки и 5 столбцов, то есть в данной матрице содержится 20 элементов численного типа. Массив, в котором количество строк совпадает с количеством столбцов, называют квадратной матрицей. В квадратной матрице есть главная диагональ, которая идет слева направо и сверху вниз.

Пример 1. Найти количество положительных элементов главной диагонали массива A(3,3).

```
DIM A(3, 3)
FOR I=0 TO 3
FOR J=0 TO 3
INPUT A(I, J)
NEXT J
NEXT I
K=0
FOR I=0 TO 3
FOR J=0 TO 3
IF I=J THEN IF A(I, J)>0 THEN K=K+1
NEXT J
NEXT I
PRINT «КОЛИЧЕСТВО ПОЛОЖИТЕЛЬНЫХ ЭЛЕМЕНТОВ=»; K
```

Пример 2. Вывести на печать номер столбца, содержащего нулевой элемент массива B(3,4).

```
DIM B(3, 4)
FOR I=0 TO 3
FOR J=0 TO 4
INPUT B(I, J)
NEXT J
NEXT I
FOR I=0 TO 3
FOR J=0 TO 4
IF B(I, J)=0 THEN PRINT J
NEXT J
NEXT I
```

Задачи для самостоятельного решения

1. Найти произведение элементов массива B(14), которые имеют четные номера.

2. Найти произведение элементов массива B(12), стоящих после минимального элемента.

3. Расположить элементы массива C(15) по возрастанию.

4. Найти количество положительных элементов главной диагонали массива C(5,5).

5. Вывести на печать номера элементов массива A(4,5), которые больше удвоенного произведения минимального элемента.

6. Найти произведение максимального и минимального элементов массива C(4,6).

Вопросы для самопроверки

1. Что понимается под языком программирования?
2. Какие языки программирования вам известны?
3. Что включает в себя система программирования?
4. Что понимается под инструментальной средой пользователя?
5. Из чего состоит алфавит языка Quick Basic?
6. Назовите основные операторы языка Quick Basic.
7. Что такое числовой массив?
8. Чем отличаются одномерные и двумерные массивы?

Литература

1. Информатика. Базовый курс / Симонович С.В. [и др.] – СПб: Питер, 2005. – 640 с.

2. Сафронов И.К. Бейсик в задачах и примерах / И.К. Сафронов. – СПб.: БХВ-Петербург, 2003. – 224 с.

СОДЕРЖАНИЕ