

№ 1761

МИСиС

Г.И. Светозарова, О.В. Андреева

Информатика

Основы программирования на языке
Турбо-Бейсик и численные методы

Учебное пособие

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

№ 1761

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ
ИНСТИТУТ СТАЛИ и СПЛАВОВ
Технологический университет



Кафедра инженерной кибернетики

Г.И. Светозарова, О.В. Андреева

Информатика

Основы программирования на языке
Турбо-Бейсик и численные методы

Учебное пособие
для студентов всех специальностей

Рекомендовано редакционно-издательским
советом института

УДК 004.42:591.6
С24

Р е ц е н з е н т
доцент *А.П. Смирнов*

Светозарова Г.И., Андреева О.В.

С24 Информатика. Основы программирования на языке Турбо-Бейсик: Учеб. пособие. – М.: МИСиС, 2003. – 127 с.

В учебном пособии приведены основные понятия программирования, описание языка Турбо-Бейсик, программная реализации типовых алгоритмов, а также рекомендации по разработке и отладке программ. Рассмотрены численные методы решения часто встречающихся инженерных задач: нахождение корней алгебраических и трансцендентных уравнений, решение систем линейных уравнений, вычисление определенного интеграла, решение дифференциальных уравнений, а также методы приближения функций.

Включены задания для самостоятельного выполнения.

Может быть использовано для самостоятельного овладения основными навыками программирования и при решении инженерных задач.

Содержание пособия соответствует программе курса «Информатика».

Пособие предназначено для студентов всех специальностей.

ОГЛАВЛЕНИЕ

Предисловие	5
1. Общее представление о компьютере. Основные понятия программирования	9
2. Основные средства языка Турбо-Бейсик	15
2.1. Структура программы	15
2.2. Переменные. Типы данных. Константы	16
2.3. Массивы	18
2.4. Выражения	19
2.5. Основные операторы Турбо-Бейсика.....	21
2.5.1. Оператор присваивания.....	21
2.5.2. Оператор перестановки Swap	22
2.5.3. Условный оператор. If-блок	22
2.5.4. Операторы цикла.....	25
2.5.5. Операторы ввода – вывода.....	28
3. Реализация простейших алгоритмов на языке Турбо-Бейсик	34
3.1. Организация циклов	34
3.2. Организация разветвлений. Разветвления в цикле	42
3.3. Составление программ для обработки потока данных.....	45
4. Типовые алгоритмы обработки массивов.....	49
4.1. Алгоритмы обработки одномерных массивов	49
4.2. Работа с матрицами.....	57
5. Графика	68
6. Процедуры и функции.....	74
7. Использование внешних устройств.....	82
7.1. Файлы данных	82
7.2. Программные файлы	84
8. Некоторые практические рекомендации по разработке и отладке программ.....	87
9. Решение инженерных задач с использованием компьютера. Численные методы.....	91
9.1. Необходимость использования численных методов. Влияние ошибок на результаты численных решений	91
9.2. Решение алгебраических и трансцендентных уравнений	92
9.2.1. Метод половинного деления	93
9.2.2. Метод итераций.....	94
9.2.3. Метод Ньютона	96

9.3. Вычисление определенных интегралов	99
9.3.1. Метод трапеций.....	100
9.3.2. Метод Симпсона	101
9.4. Решение систем линейных уравнений	104
9.5. Интерполирование	107
9.6. Аппроксимация	110
9.7. Решение обыкновенных дифференциальных уравнений. Метод Эйлера	116
9.8. Обеспечение необходимой точности	121
Библиографический список	124
Приложение. Встроенные функции. Логические операции	125

ПРЕДИСЛОВИЕ

В настоящее время решение любой инженерной, научной или технической задачи немыслимо без использования компьютера. Современные компьютеры имеют мощное программное обеспечение, включающее пакеты программ (приложения), позволяющие решать задачи из различных областей без трудоемкой разработки программ, обеспечивая, таким образом, доступ к этой мощной технике и непрофессиональным пользователям.

Однако, чтобы грамотно использовать эти возможности, необходимо иметь понятие об основах программирования, способах представления данных, ограничениях, связанных с особенностями вычислительного процесса, осуществляемого компьютером, вычислительных методов, к которым приходится прибегать, чтобы получить решение задачи, и т.п. Поэтому в курсе «Информатика» перед использованием приложений Windows изучаются основы алгоритмизации и программирования и простейшие численные методы в том минимальном объеме, который позволяет сформировать представление об используемых в этой области подходах, дает возможность грамотно обратиться к специалистам в сложных случаях, когда использование стандартных средств невозможно, или внести изменения в уже разработанные программы, приспособив их к своей задаче.

При изучении основ алгоритмизации с применением структурного подхода и метода пошаговой детализации формируются также навыки, которые с успехом можно использовать и для решения сложных инженерных задач, представляя исходную задачу в виде последовательности более простых задач, что может быть полезным в будущей работе.

В качестве языка программирования используется Турбо-Бейсик, который не перегружен излишней формализацией, но в то же время находится в ряду современных алгоритмических языков, содержит конструкции для реализации типовых структур алгоритма и более мощные средства: процедуры и функции, графические средства, файлы данных и программные файлы с возможностью подключения к основной программе и т.д. Использование Турбо-Бейсика позволяет изучение программирования сразу начать с решения практических задач, не затрачивая при этом много времени на изучение синтаксиса, осознание того, что такое «тип переменной» и т.п. слож-

ных понятий, которые формируются постепенно на протяжении всего времени обучения.

Изучив основы программирования, нетрудно освоить и любой другой (более строгий и мощный) язык.

Книга состоит из двух частей: первая (гл. 1–8) посвящена основам алгоритмизации и программирования, а вторая (гл. 9) – вычислительным методам решения типовых задач, часто встречающихся в инженерной практике, и особенностям их программной реализации. При этом рассматриваются наиболее простые и доступные для понимания методы с тем, чтобы дать представление о подходах к решению реальных инженерных задач, что позволит грамотно пользоваться пакетами прикладных программ или в нестандартных случаях обращаться к специалистам.

Изложение сопровождается примерами программ, демонстрирующих основные средства или приемы программирования. Каждая глава содержит вопросы для самопроверки и задания для самостоятельного выполнения, работа над которыми будет способствовать усвоению изучаемого материала.

Для понимания основ программирования необходимо с самого начала изучения курса иметь практику составления программ и их реализации на компьютере. Это требует знания и учета многих аспектов решения задач с использованием компьютера, а именно: основных понятий программирования, таких, как переменная, присваивание, цикл и т.п., основ алгоритмизации, особенностей представления данных в компьютере, языка и среды программирования (т.е. основных правил работы на компьютере при наборе, редактировании, отладке и выполнении программы), умения интерпретировать полученные результаты и т.д. Изучение этих разделов по отдельности не имеет смысла, так как значение каждого проявляется (и становится понятным) только во взаимодействии с другими. С этим связаны трудности начального освоения программирования, так как продвигаться вперед можно только развивая параллельно все аспекты.

В пособии представлен материал, необходимый для выполнения лабораторного практикума по курсу «Информатика» (авторы Г.И. Светозарова, М.С. Бесфамильный, Ю.А. Кудрявцев).

Глава 1 дает общее представление о компьютере и основных понятиях программирования (переменная, алгоритм, операторы), типовых структурах алгоритма, современных методах разработки алгоритмов и программ. К главе 1 придется многократно возвращаться на

протяжении всего курса, и определенные в ней понятия будут полностью усвоены только после приобретения собственного опыта программирования.

В главе 2 дается описание основных средств языка Турбо-Бейсик, и даже из приведенных здесь на первых порах потребуется лишь их минимальный набор.

Глава 3 посвящена разработке циклических и разветвляющихся программ, глава 4 – работе с массивами.

Главы 1 – 4 формируют базовые понятия программирования и в сочетании с выполнением лабораторных работ 1 – 3 из Практикума позволяют освоить основы программирования.

Рекомендуется следующий порядок работы: прочитать главу 1, прочитать главу 2 (§ 2.1, 2.2, 2.4, п. 2.5.1, 2.5.4) и перейти к главе 3, § 3.1, где рассматриваются типовые алгоритмы циклической структуры и приемы разработки циклических программ. Параллельно с изучением § 3.1 необходимо решить задачи, предлагаемые для самостоятельного выполнения к главе 3 (№ 1 – 16), и выполнить лабораторную работу 1 из Практикума. Если возникают трудности при разработке алгоритмов и программ на Турбо-Бейсике, то необходимо вернуться к главе 1 и соответствующим фрагментам главы 2, а также использовать типовые алгоритмы циклической структуры, приведенные в главе 3. Заметим, что для понимания алгоритма, записанного, как правило, в виде программы или фрагмента программы, его нужно выполнить, используя правила, по которым работают операторы Турбо-Бейсика, задавая, если требуется, простые исходные данные, не требующие больших вычислений.

После составления программы ее целесообразно проверить вручную. Это значит, что Вы работаете как компьютер, выполняя операторы программы последовательно друг за другом (забыв о задаче, видя перед собой только программу), выписывая значения изменяющихся переменных. Если при реализации на компьютере программа дает не те результаты, которые были получены при ручном выполнении, то это означает, как правило, что Вы неправильно представляете, как работают операторы Турбо-Бейсика. Нужно вернуться к главе 2 и проверить свои знания.

Далее таким же образом изучить § 3.2 и 3.3, выполнив задания № 17 – 20 и лабораторную работу 2.

Глава 4 посвящена работе с массивами (средства Турбо-Бейсика для работы с массивами изложены в § 2.3). Усвоив типовые алгоритмы обработки массивов, решив задачи для самостоятельного

выполнения и выполнив лабораторную работу 3, уровень 1, можно перейти к решению реальных задач, требующих использования массивов (задачи 53 – 55 и лабораторная работа 3, уровень 2). Перед этим целесообразно прочитать главу 8.

Главы 5 – 7 посвящены более сложным приемам программирования и средствам языка, используемым для наглядного вывода результатов (гл. 5), реализации принципа программирования сверху вниз (гл. 6) и использованию внешних устройств для хранения файлов данных и программных файлов и их использованию в программах (гл. 7).

Глава 9 посвящена вопросам решения инженерных задач с использованием компьютера. Здесь наряду с численными методами рассматриваются также практические аспекты распространения ошибок, достижения необходимой точности и т.п. С этой главой согласованы лабораторные работы 6 – 9.

1. ОБЩЕЕ ПРЕДСТАВЛЕНИЕ О КОМПЬЮТЕРЕ. ОСНОВНЫЕ ПОНЯТИЯ ПРОГРАММИРОВАНИЯ

Современные компьютеры – мощное средство для решения самых разнообразных задач. Это также эффективное средство для хранения информации и связи с внешним миром (например, посредством глобальной сети Интернет).

В самом общем виде компьютер представляет собой устройство (точнее, совокупность взаимосвязанных устройств), которое способно хранить информацию и выполнять определенный набор элементарных арифметических и логических операций, таких, как «сложить два числа», «перемножить два числа», «сравнить два числа» и т.п. Выполнив одну операцию, компьютер может автоматически перейти к выполнению следующей и, таким образом, может выполнять длинные цепочки операций без вмешательства человека. Скорость выполнения операций компьютером очень высокая: до миллионов и миллиардов операций в секунду.

Последовательность операций, которую должен выполнять компьютер, записанная в доступной для него форме, называется *программой* (программой на машинном языке). То обстоятельство, что компьютер работает по программе, составляет сущность программного принципа управления работой компьютера. Процесс разработки программ для компьютера называется *программированием*. Программа на машинном языке является очень детальной, и форма ее представления неудобна для человека. Как правило, программы составляют на так называемых *алгоритмических языках*, более естественных и удобных для человека, а преобразование программы в машинную форму осуществляет сам компьютер с помощью специальных программ – *трансляторов*.

Основными устройствами, входящими в состав компьютера, являются *процессор, запоминающее устройство, устройства ввода и вывода*. Процессор выполняет операции и координирует работу всех устройств. В его состав входит арифметико-логическое устройство и устройство управления. В запоминающем устройстве (или памяти) хранится программа и данные, обрабатываемые этой программой. Команды на выполнение отдельных операций и данные, над которыми операция должна быть выполнена, по очереди передаются в процессор. Результаты выполнения операций передаются обратно в

память. Таким образом, память и процессор при выполнении программы постоянно обмениваются информацией.

Запоминающее устройство, о котором шла речь, называется *оперативным* запоминающим устройством (ОЗУ). Оно имеет высокое быстродействие и физически размещается рядом с процессором. Помимо этого имеется и *внешнее* (или долговременное) запоминающее устройство (ВЗУ). Оно имеет больший объем (при более низком быстродействии) и может хранить информацию достаточно долго, тогда как из оперативной памяти вся информация исчезает после выключения компьютера. Наиболее часто для длительного хранения информации используются жесткие диски (винчестеры), гибкие диски (дискеты), а также лазерные диски (CD-ROM).

Оперативная память состоит из элементов, каждый из которых может находиться в одном из двух устойчивых состояний (двоичных запоминающих элементов). Одному из двух возможных состояний ставится в соответствие 0, другому – 1. Минимальная единица информации, представленная одним двоичным элементом, носит название *бит* (bunary digit). Чтобы хранить более емкую информацию, двоичные запоминающие элементы объединяются в группы. Группа, состоящая из 8 бит, называется *байтом*. Все байты пронумерованы, номер байта называется его *адресом*. Байты могут объединяться в более крупные единицы памяти (ячейки памяти). Каждый компьютер имеет характерную для него длину (число байт) ячейки памяти. Такая ячейка характерной длины называется *словом*. Это не исключает возможности использования ячеек памяти другой длины (например, полуслово, двойное слово и т.п.).

Любая информация в компьютере, таким образом, должна представляться последовательностью нулей и единиц. Если учесть, что 0 и 1 являются цифрами системы счисления по основанию 2 (двоичной системы счисления), то можно такую последовательность нулей и единиц считать двоичным числом. Тогда соответствие между привычным представлением числа в десятичной системе (внешнее представление) и представлением этого числа в компьютере (внутреннее представление) может быть установлено правилами перевода чисел из десятичной системы счисления в двоичную (полностью справедливо только для целых чисел, см. ниже). Перевод чисел из внешнего представления во внутреннее и наоборот компьютер осуществляет автоматически.

Набором нулей и единиц можно закодировать также и нечисловую информацию, например, буква А (лат.) кодируется набором 01000001, буква В (лат.) – 01000010 и т.д. Это позволяет, помимо вычислительных, решать с помощью компьютера также информационные задачи, осуществлять обработку текстов и т.п. [1 – 3].

В качестве *устройства ввода* обычно используется клавиатура дисплея, а в качестве *устройства вывода* – экран дисплея или печатающее устройство (принтер).

Решение любой задачи на компьютере предполагает наличие программы и данных, которые этой программой должны быть обработаны, и в соответствии с этим перед нами встают две задачи: как описать порядок действий, необходимых для решения задачи, и как представить данные (числа) в памяти компьютера.

Самой простой структурой данных является *переменная* (простая переменная). Переменную можно представить себе как ящик с биркой, на которой написано имя. (В памяти компьютера это – ячейка памяти, имеющая определенный адрес.) Обращение к переменной осуществляется по имени. В переменной может одновременно храниться одно значение. При занесении в переменную другого значения старое значение пропадает.

Действия записываются с помощью операторов. Существует три основных оператора, с помощью различных сочетаний которых можно в принципе описать порядок решения любых задач. Это – операторы присваивания, условный оператор и оператор передачи управления.

Оператор присваивания имеет следующий вид:

переменная = *выражение*,

где знак = обозначает операцию присваивания, *выражение* аналогично знакомому из математики алгебраическому выражению (здесь речь идет о числовом выражении), задающему правило получения значения, а *переменная* – имя переменной, в которую будет помещен результат. (В гл. 2 даются более корректные определения введенных понятий.)

Оператор присваивания выполняется следующим образом: сначала вычисляется выражение, стоящее справа, а затем результат помещается в переменную, имя которой указано слева от знака операции присваивания.