Самоучитель

Никита Культин

Программирование в Turbo Pascal 7.0 и Delphi

3-е издание

От алгоритма до работающей программы

Язык программирования Turbo Pascal

Работа с файлами и графикой

Введение в объектно-ориентированное программирование

Работа в среде Delphi

bhv + CD

Никита Культин

Самоучитель
Программирование в Turbo Pascal 7.0 и Delphi 3-е издание

Санкт-Петербург

«БХВ-Петербург»

2007

УДК 681.3.068
ББК 32.973.26-018.1
К90

Культин Н. Б.

К90 Программирование в Turbo Pascal 7.0 и Delphi: 3-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2007. — 400 с.: ил. + CD-ROM — (Самоучитель)

ISBN 978-5-9775-0109-5

Книга позволяет научиться программированию на языке Pascal в среде Turbo Pascal. Рассмотрен весь процесс создания программы: от разработки алгоритма до получения результата — готовой программы. Приведено описание языка программирования и среды разработки; рассмотрены основные типы данных и алгоритмические структуры. Уделено внимание обработке символьной информации, использованию динамических структур данных, работе с файлами, выводу данных на печать, программированию графики. Описана среда визуального программирования Delphi и показаны основы разработки в ней Windows-приложений.

Книга отличается доступностью изложения материала, большим количеством наглядных примеров и адресована студентам, школьникам старших классов и всем, кто изучает программирование. На прилагаемом компакт-диске находятся приведенные в книге тексты программ.

Для начинающих программистов

УДК 681.3.068
ББК 32.973.26-018.1

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Игорь Шишигин*
Зав. редакцией	*Григорий Добин*
Редактор	*Ирина Иноземцева*
Компьютерная верстка	*Натальи Караваевой*
Корректор	*Виктория Пиотровская*
Дизайн серии	*Инны Тачиной*
Оформление обложки	*Елены Беляевой*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 28.07.05.
Формат $70\times100^{1}/_{16}$. Печать офсетная. Усл. печ. л. 32,25.
Тираж 3000 экз. Заказ №
"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 978-5-9775-0109-5

Оглавление

Предисловие

В компьютерном мире существует множество языков программирования. Программу, выполняющую одни и те же действия, можно написать на Бэйсике (BASIC), Паскале (Pascal), Си (C). Какой из языков лучше? Ответить на этот вопрос однозначно нельзя. Однако можно с уверенностью утверждать, что Pascal лучше других языков подходит для обучения программированию. И это не удивительно, ведь этот язык был разработан швейцарским ученым Никлаусом Виртом (Niklaus Wirth) специально для обучения программированию.

С момента появления Pascal за короткое время различными фирмами было создано достаточно большое количество компиляторов (компилятор — программа, переводящая инструкции языка программирования на язык команд процессора вычислительной машины). Одной из наиболее удачных стала разработка американской фирмы Borland, в которой были объединены редактор текста и высокоэффективный компилятор. Разработка получила название Turbo Pascal, а язык программирования, используемый в системе Turbo Pascal, стал называться Turbo Pascal. Следует обратить внимание, что Pascal лежит в основе используемого в среде разработки Delphi языка программирования Delphi.

Pascal — не "учебный", не "игрушечный" язык, он используется для разработки сложных, "профессиональных" программ, в том числе, предназначенных для работы в Windows.

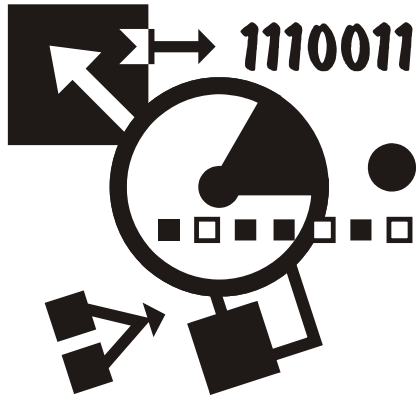
Книга, которую вы держите в руках, представляет собой учебное пособие по программированию на языке Turbo Pascal. Помимо описание языка программирования и среды разработки, в ней кратко изложены общие вопросы программирования. Материал книги несколько шире традиционного курса программирования. В ней изложен ряд тем, которые, как правило, остаются за рамками подобных учебников: обработка символьной информации, использование динамических структур, работа с файлами, отладка, вопросы объектно-ориентированного программирования.

Особое внимание в книге уделено практике составления программ и выработке навыков программирования; рассмотрены типичные ошибки, даны рекомендации по их устранению и предотвращению.

Последовательность изложения материала построена так, чтобы читатель как можно скорее приступил к самостоятельной работе на компьютере.

В качестве дополнительного материала в книге рассматриваются вопросы программирования для Windows в среде визуального программирования Delphi. Приведено описание среды, даны основные определения и термины, на конкретном примере изложена методика создания программ.

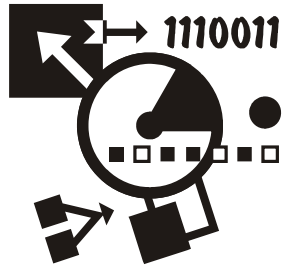
Научиться программировать можно только программируя, решая конкретные задачи. Поэтому, чтобы получить максимальную пользу от книги, вы должны работать с ней активно. Не занимайтесь просто чтением примеров. Вводите их в компьютер. Не бойтесь экспериментировать — вносите изменения в программы. Чем больше вы сделаете самостоятельно, тем большему вы научитесь.



Часть I

TURBO PASCAL

Первая часть книги посвящена программированию в Turbo Pascal. В ней приведено описание среды разработки, языка программирования, рассмотрены основные алгоритмические структуры и структуры данных, операции со строками, массивами, записями и файлами, программирование графики.



Глава 1

Среда программирования Turbo Pascal

Turbo Pascal представляет собой интегрированную среду программирования (разработки компьютерных программ) для операционной системы MS DOS. Термин "интегрированная" обозначает, что среда разработки объединяет в себе несколько элементов, а именно: редактор кода (так программисты называют редактор текста программ), компилятор и отладчик. В качестве языка программирования в Turbo Pascal используется алгоритмический язык Паскаль (Pascal).

Установка

Установка Turbo Pascal на компьютер выполняется путем запуска с установочного CD программы install.exe. В процессе установке на диске C: (по умолчанию Turbo Pascal устанавливается на этот диск) будет создан каталог TP, в подкаталоги которого будут помещены файлы, образующие Turbo Pascal.

По завершении процесса установки, перед тем как запустить Turbo Pascal первый раз, рекомендуется на диске компьютера создать два каталога: первый — для текстов программ, второй — для выполняемых файлов. Назвать эти каталоги можно, например, PAS и EXE&TPU, а разместить — в каталоге TP.

Запустить Turbo Pascal можно, набрав в окне **Запуск программы** (рис. 1.1) C:\TP\BIN\TURBO.EXE. Однако лучше создать на рабочем столе ярлык, обеспечивающий запуск Turbo Pascal.

Чтобы создать ярлык, обеспечивающий запуск Turbo Pascal, надо сделать щелчок правой кнопкой мыши на свободном месте рабочего стола, из появившегося меню выбрать команду **Создать ▸ Ярлык** и в появившемся окне **Создание ярлыка** указать имя файла, обеспечивающего запуск среды разработки (рис. 1.2). В следующем окне, которое становится доступным в результате щелчка на кнопке **Далее**, надо указать подпись ярлыка (рис. 1.3).

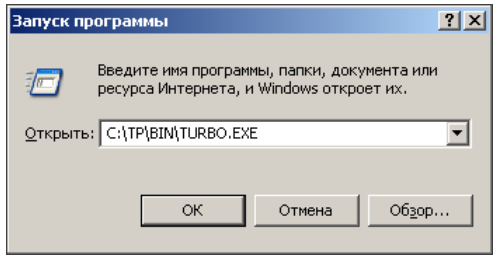


Рис. 1.1. Запуск Turbo Pascal

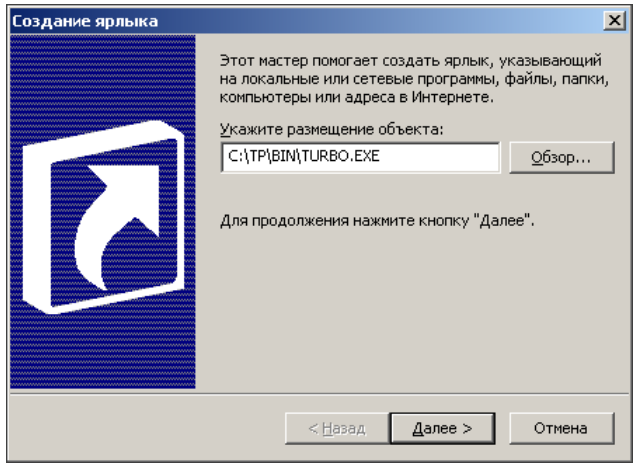


Рис. 1.2. Создание ярлыка Turbo Pascal (шаг 1)

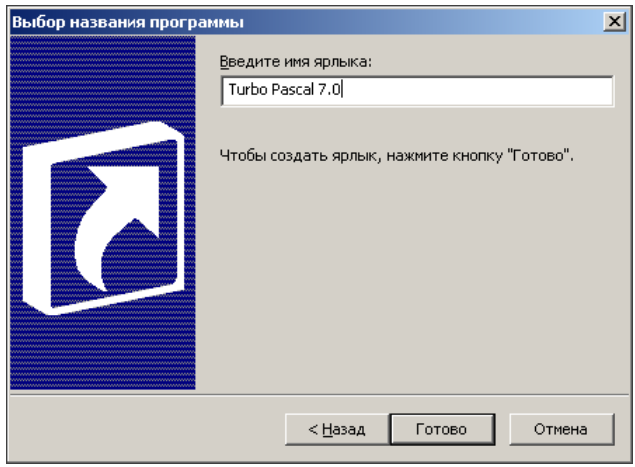


Рис. 1.3. Создание ярлыка Turbo Pascal (шаг 2)

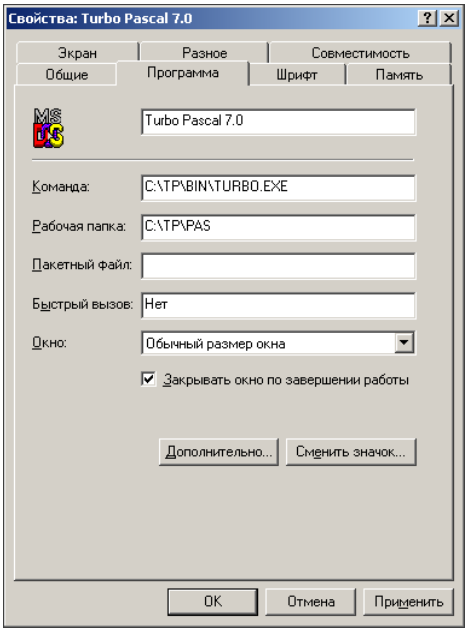


Рис. 1.4. Настройка рабочей папки

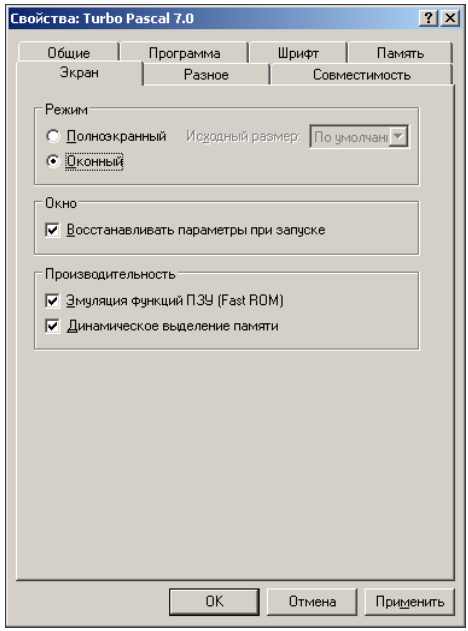


Рис. 1.5. Настройка режима окна

После того как ярлык будет создан, рекомендуется выполнить его настройку — задать рабочий, то есть используемый по умолчанию, каталог и режим экрана. Настройка ярлыка выполняется в окне **Свойства**, которое становится доступным в результате выбора в контекстном меню ярлыка команды **Свойства**. Имя рабочего каталога надо ввести в поле **Рабочая папка** вкладки **Программа** (рис. 1.4), а режим экрана (оконный) — задать на вкладке **Экран** (рис. 1.5).

Начало работы

Чтобы начать работу в Turbo Pascal, надо сделать двойной щелчок мышью на находящемся на рабочем столе ярлыке (процесс создания и настройки ярлыка подробно описан в предыдущем параграфе).

Вид окна, в котором работает Turbo Pascal, приведен на рис. 1.6. В верхней части находится строка главного меню, в нижней — строка подсказки. Основную часть окна занимает окно редактора кода (текста программы).

Шрифт, который используется для отображения текста в окне, и, как следствие, размер окна, можно изменить. Для этого надо сделать щелчок правой кнопкой мыши на значке, который находится в заголовке окна, в появившемся

меню выбрать команду **Свойства** и затем на вкладке **Шрифт** задать характеристики шрифта (например, точечный шрифт 10×18). Также можно увеличить количество строк текста, отображаемых в окне редактора (по умолчанию в окне отображается 23 строки текста). Для этого в меню **Options** надо выбрать команду **Environment ▸ Preferences** и в появившемся окне, в группе **Screen sizes** установить переключатель **43/50 lines**.

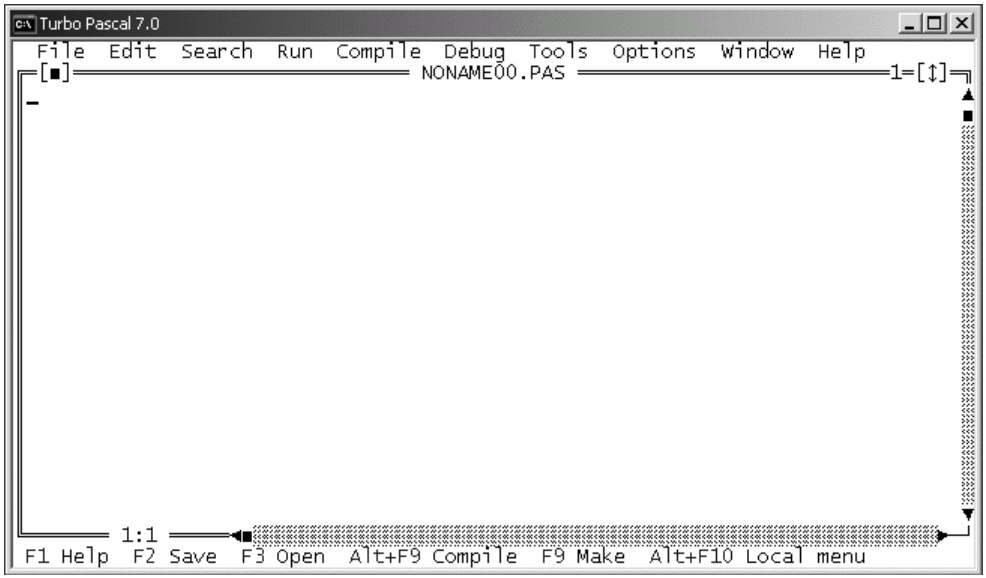


Рис. 1.6. Окно, в котором работает Turbo Pascal

Первая программа

Процесс работы в Turbo Pascal рассмотрим на примере. Создадим программу, при помощи которой можно рассчитать доход по вкладу в банке. Доход можно вычислять по формуле:

Доход = Сумма * (Процентная ставка / 365) * Срок

где: Сумма — сумма вклада; Срок — срок в клада (количество дней); Процентная ставка — процент, начисляемый на сумму вклада, при условии, что срок вклада — год; 365 — количество дней в году. На практике процентная ставка зависит от суммы вклада, чем больше сумма, тем больше процентная ставка. Будем считать, что процентная ставка равна 8%, если сумма вклада меньше 5 тыс. рублей, и 9,5%, если сумма вклада больше указанного значения.

Перед тем как приступить к непосредственной работе в Turbo Pascal на компьютере, рекомендуется разработать (составить) алгоритм решения задачи и написать программу на бумаге. Алгоритм программы (решения поставленной задачи) приведен на рис. 1.7, программа — в листинге 1.1 (текст программы принято называть листингом).

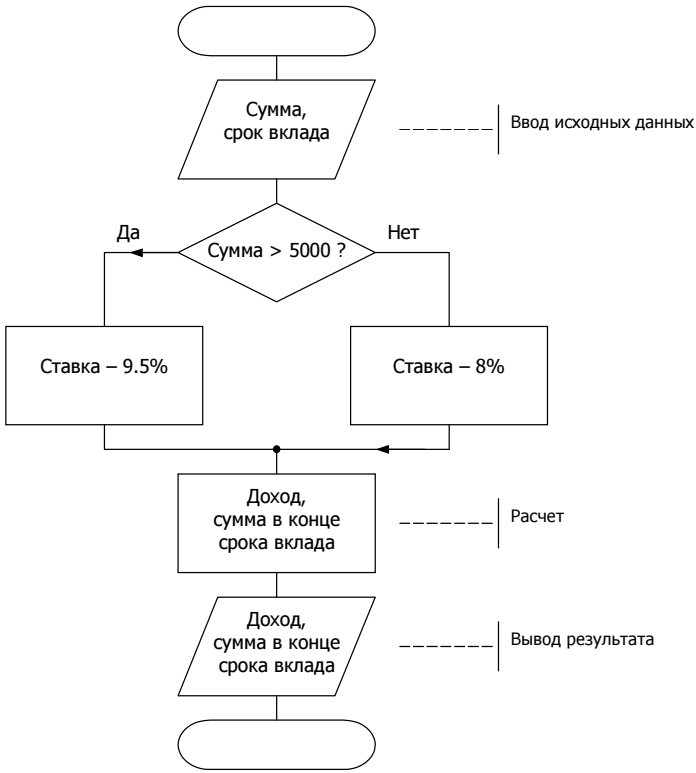


Рис. 1.7. Алгоритм программы вычисления дохода по вкладу

Листинг 1.1. Вычисление дохода по вкладу в банке (p1_1.pas)

```
{ Вычисление дохода по вкладу в банке }
program p1_1;


var
   sum: real;        { сумма вклада }
```

```
   percent: real;    { процентная ставка }
   period: integer; { срок вклада }

   profit: real;     { доход }
   result: real;     { сумма в конце срока вклада }

begin
   writeln('Вычисление дохода по вкладу в банке');
   writeln;

   { ввод исходных данных }
   write('Сумма (руб.) -> ');
   readln(sum);
   write('Срок вклада (дней) -> ');
   readln(period);

   { определить величину процентной ставки }
   if sum > 5000 then
           percent := 0.095 { ставка 9.5%}
        else
           percent := 0.08; { ставка 8%}

   profit := sum * percent/365 * period;
   result  := sum + profit;

   { вывод результата расчета }
   writeln('Сумма в конце срока вклада: ',
            result:6:2, ' руб.');
   writeln('Доход: ', profit:6:2, ' руб.');

   write('Для завершения работы программы нажмите <Enter>');
   readln;

end.
```

Первая строка программы, заключенный в фигурные скобки текст, — это комментарий. Комментарии включают в текст программы, чтобы пояснить назначение программы, переменных, ключевые точки алгоритма. Комментарии облегчают восприятие программы, позволяют понять, как она работает.

Следующая строка — это заголовок программы. В заголовке, за словом `program`, указано имя программы. Далее следует слово `var` (сокращение от `variable` — переменная), отмечающее начало раздела объявления переменных. В разделе объявления переменных перечислены (объявлены) переменные, используемые в программе. После имени переменной, через двоеточие, указан тип данных, для хранения которых предназначена переменная (`real` — дробный тип, `integer` — целый тип). Слово `begin` отмечает начало раздела выполняемых инструкций программы. Первая инструкция программы, инструкция `writeln('Вычисление дохода по вкладу в банке')`, выводит на экран текст **Вычисление дохода по вкладу в банке** и переводит курсор в начало следующей строки. Далее следуют инструкции, обеспечивающие ввод исходных данных с клавиатуры: инструкция `write('Сумма (руб.) ->')` выводит подсказку, а инструкция `readln(sum)` записывает данные, которые пользователь набрал на клавиатуре, в переменную `sum`. Аналогичным образом обеспечивается ввод значения переменной `period`. Следующие инструкции реализуют расчет. Сначала при помощи инструкции `if` определяется величина процентной ставки (значение переменной `percent`). Если значение переменной `sum` больше пяти тысяч, то переменной `percent` присваивается значение 0.095 (процентная ставка 9,5%), в противном случае, то есть если значение `sum` меньше или равно 5000, переменной `percent` присваивается значение 0.08 (процентная ставка 8%). Вычисление дохода (значения переменной `profit`) обеспечивает инструкция `profit := sum * (percent / 365) * period`, суммы в конце срока вклада — инструкция `result := sum + profit`. Далее следуют инструкции, которые выводят на экран результат расчета. Инструкция `writeln('Сумма в конце срока вклада: ', result:6:2, ' руб.')` выводит текст **Сумма в конце срока вклада:**, значение переменной `result` и слово **руб**. Инструкция `writeln('Доход: ', profit:6:2, ' руб.')` выводит значение переменной `profit`. Значения переменных `result` и `profit` отображаются с двумя цифрами после десятичной точки. В конец программы добавлена инструкция `readln`, которая приостанавливает работу программы до тех пор, пока пользователь не нажмет клавишу <Enter>. Так как инструкция `readln` является последней инструкцией программы, то после того, как пользователь нажмет <Enter>, программа завершит работу и окно, в котором работала программа, закроется (исчезнет с экрана).

Набор текста программы

По умолчанию при запуске среды разработки автоматически открывается окно редактора кода (текста программы). Поэтому сразу после запуска Turbo Pascal можно набирать текст программы.

Программу в окне редактора кода набирают обычным образом.

Примечание

Переключение на русский алфавит осуществляется путем нажатия правой клавиши <Shift> (при нажатой клавише <Ctrl>), на латинский — левой клавиши <Shift>. Следует обратить внимание, что комбинация <Ctrl> + "правый" <Shift> для переключения на русский алфавит работает только в том случае, если в настройках операционной системы в качестве языка ввода по умолчанию задан русский язык.

Редактор кода автоматически выделяет цветом ключевые слова языка программирования (`program`, `var`, `begin`, `end` и другие) и комментарии, что облегчает восприятие структуры программы.

В процессе набора текста необходимо соблюдать правила хорошего стиля программирования — записывать инструкции с отступами и добавлять в текст комментарии. В качестве примера на рис. 1.8 приведено окно редактора кода, в котором находится текст программы вычисления дохода по вкладу.

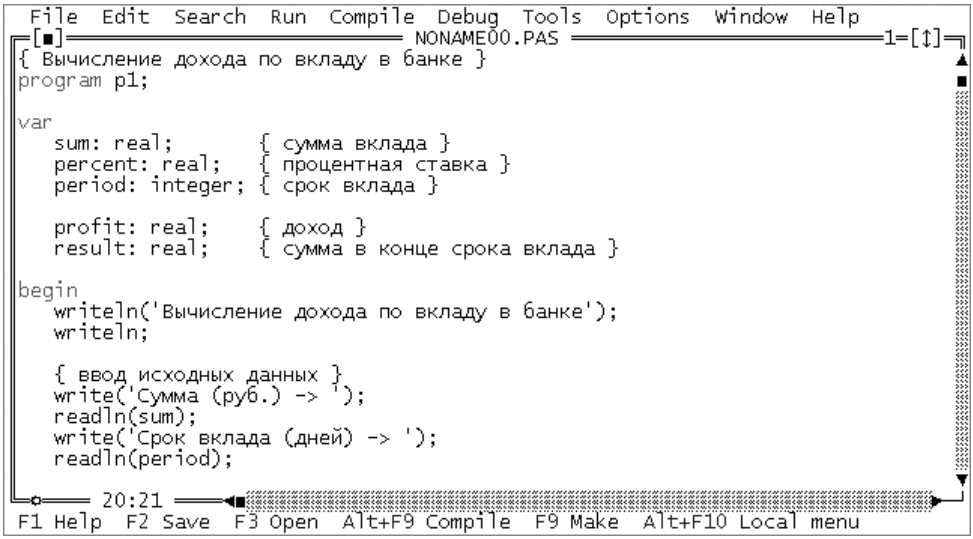


Рис. 1.8. Текст программы в окне редактора кода

После того как текст программы будет набран, его надо сохранить на диске. Для этого в меню **File** надо выбрать команду **Save**, в поле **Save file as** появившегося окна (рис. 1.9) ввести имя файла и сделать щелчок на кнопке **OK**. Следует обратить внимание, что в нижней части окна отображается имя каталога, в который будет помещен файл программы (если программист в поле **Save file as** явно не укажет диск и каталог). Расширение имени файла (PAS) можно не указывать, оно будет добавлено к имени автоматически.

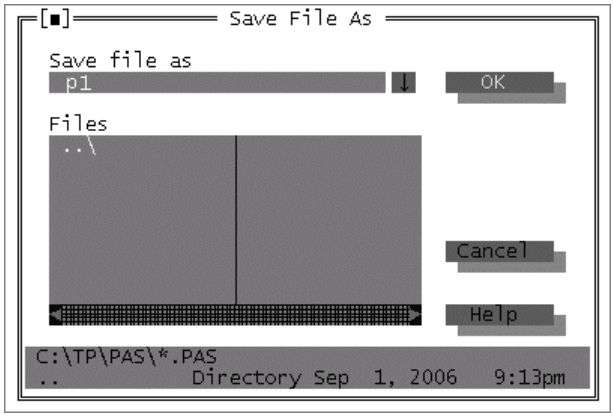


Рис. 1.9. Сохранение программы

Примечание

В Turbo Pascal при записи имен файлов разрешается использовать *только* буквы латинского алфавита и цифры (буквы русского алфавита и пробел использовать нельзя). Кроме того, количество символов в имени файла (без учета точки и расширения PAS) не должно превышать восьми.

Компиляция

Программа, представленная на языке программирования (набранная программистом в окне редактора кода), называется исходной. Это *текст*, понятный человеку. Для того чтобы программа могла быть выполнена процессором, ее необходимо преобразовать в выполняемую программу — последовательность машинных команд. Процесс преобразования исходной программы в выполняемую называется компиляцией.

Чтобы активизировать процесс компиляции, нужно в меню **Compile** выбрать команду **Compile**. В случае, если в программе нет синтаксических ошибок (то есть все инструкции набраны правильно), на экране появляется окно, информирующее об успешном завершении компиляции (рис. 1.10).

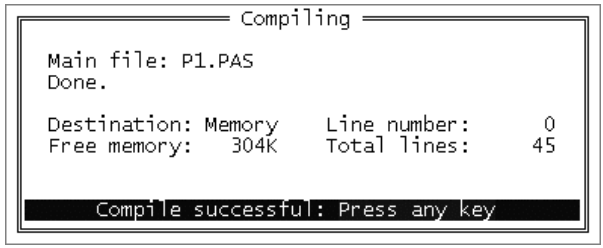


Рис. 1.10. Сообщение об успешном завершении компиляции

Turbo Pascal поддерживает два режима компиляции: "в память" (Memory) и "на диск" (Disk). В режиме Memory создаваемая компилятором выполняемая программа записывается в оперативную память компьютера, в режиме Disk — в exe-файл. Режим компиляции отображается в окне **Compiling** (рис. 1.10) и в строке **Destination** меню **Compile**. По умолчанию компиляция выполняется в память, поэтому запустить откомпилированную программу можно только из среды разработки. Чтобы программу можно было запустить из операционной системы, надо выполнить ее компиляцию на диск (в режиме Disk). Для этого необходимо изменить режим компиляции с Memory на Disk — в меню **Compile** выбрать команду **Destination** (рис. 1.11). Следует обратить внимание, что при компиляции "на диск" exe-файл будет помещен в каталог, имя которого указано в поле **EXE&TPU directory** окна **Directories** (это окно становится доступным в результате выбора в меню **Options** команды **Directories**) или, если имя каталога в этом поле не задано, в каталог, в котором находится компилируемый PAS-файл.

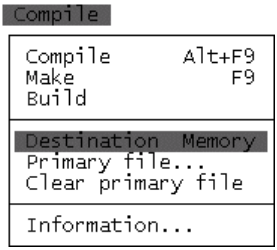


Рис. 1.11. Чтобы изменить режим компиляции, надо в меню **Compile** выбрать команду **Destination**

Ошибки времени компиляции

Очевидно, что в программе могут быть ошибки. Различают синтаксические и алгоритмические ошибки. Синтаксические ошибки — это ошибки записи инструкций программы. Алгоритмические ошибки — это ошибки, связанные с нарушением логики работы программы.

В процессе компиляции текст программы проверяется на отсутствие синтаксических ошибок. Компилятор просматривает программу от начала. Обнаружив ошибку, компилятор выводит сообщение об ошибке (код ошибки и пояснение) и устанавливает курсор в ту точку текста программы, в которой, скорее всего, она находится. В качестве примера на рис. 1.12 приведен результат компиляции программы, в которой есть ошибка. В приведенном

примере ошибка заключается в том, что имя переменной записано неверно: в программе объявлена переменная `sum` (см. листинг), а в инструкции `readln` указано `summ`.

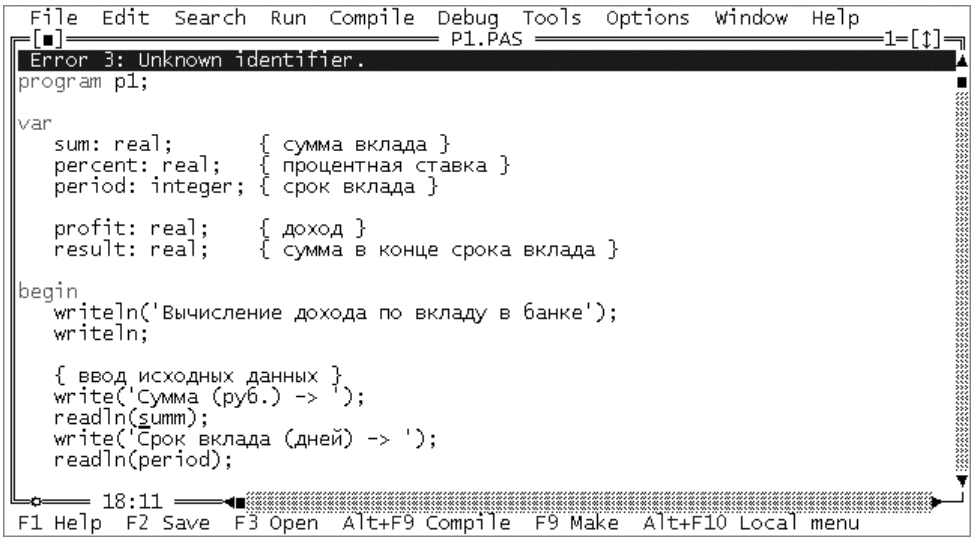


Рис. 1.12. Пример сообщения об ошибке

После анализа причины ошибки и ее устранения (в приведенном примере надо удалить лишнюю букву m) надо снова активизировать процесс компиляции. Таким образом, последовательно исправляя ошибки, обнаруживаемые компилятором, можно устранить все синтаксические ошибки.

В табл. 1.1 приведены сообщения о наиболее типичных ошибках.

Таблица 1.1. *Сообщения компилятора о типичных ошибках*

Сообщение компилятора	**Вероятная причина**
3: Unknown identifier (Неизвестный идентификатор)	Используется переменная, не объявленная в разделе `var` программы; Ошибка записи имени переменной. Например, в разделе `var` объявлена переменная `sum`, а в тексте программы написано `suma`
26: Type mismatch (Несоответствие типов)	В инструкции присваивания тип выражения не соответствует типу переменной, которой присваивается значение выражения

Таблица 1.1 *(окончание)*

Сообщение компилятора	Вероятная причина
85: ";" expected (Ожидается символ "точка с запятой")	Не поставлен символ "точка с запятой" после инструкции
113: Error in statement (Ошибка в выражении)	Неверный синтаксис оператора; например, в инструкции `if` поставлен символ "точка с запятой" после инструкции, следующей за словом `then` (фактически перед `else`)

Запуск программы

Если компиляция программы завершена успешно, то программу, текст которой находится в окне редактора кода, можно запустить. Для этого в меню **Run** надо выбрать команду **Run**. В результате запуска программы становится доступным окно прикладной программы. В это окно программа выводит сообщения, и из этого окна она получает от пользователя данные. На рис. 1.13 в качестве примера приведено окно, в котором работает программа вычисления дохода по вкладу.

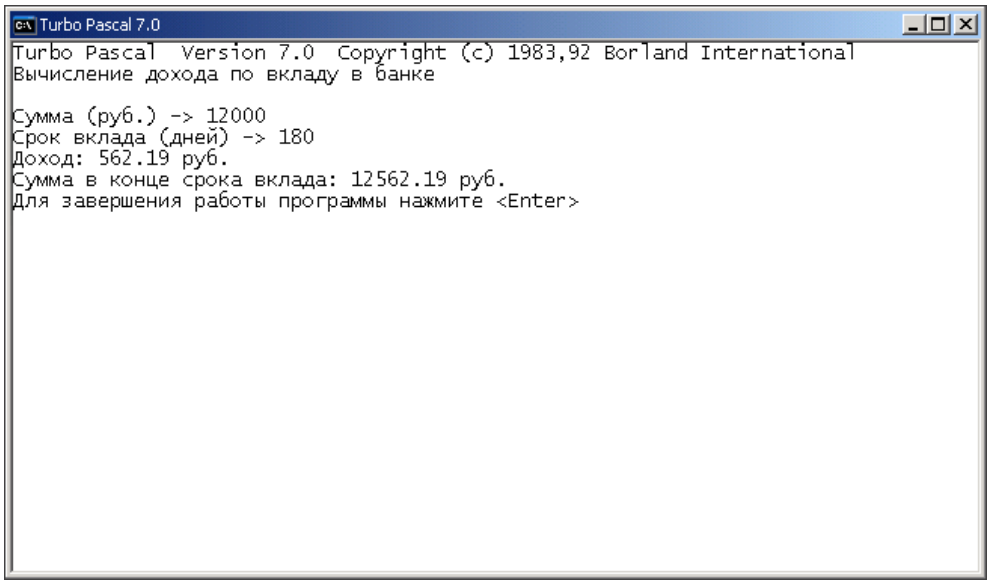


Рис. 1.13. Окно, в котором работает программа, запущенная из Turbo Pascal

По завершении работы программы окно, в котором она работала, автоматически закрывается и вновь становится доступным окно Turbo Pascal (увидеть окно, в котором работала прикладная программа, можно, выбрав в меню **Debug** команду **User Screen** или нажав <Alt> + <F5>).

> ***СОВЕТ***
>
> Чтобы окно, в котором работает программа, не исчезало с экрана сразу после завершения работы программы и, чтобы пользователь смог спокойно оценить результат ее работы, добавьте в конец программы инструкции: `writeln('Программа завершила работу. Нажмите <Enter>'); readln;`

Ошибки времени выполнения

Во время работы программы возможны так называемые ошибки времени выполнения (runtime error). В большинстве случаев причинами ошибок во время работы программы являются неверные данные.

При возникновении ошибки работа программы завершается. Если программа запущена из Turbo Pascal, то в окно редактора кода выводится сообщение об ошибке и курсор устанавливается в ту строку текста программы, в которой находится инструкция, при выполнении которой возникла ошибка. В качестве примера на рис. 1.14 приведено окно Turbo Pascal после возникновения ошибки во время работы программы вычисления дохода по вкладу в результате ввода пользователем неверных данных — строки `10000,00` (1000 рублей 00 копеек). Курсор в окне редактора кода находится перед инструкцией `readln(sum)`. Это показывает, что ошибка произошла во время выполнения именно этой инструкции. Сообщение `Error 106: Invalid numeric format` (неверный формат числа) информирует о том, что причина ошибки — неверные данные (программа ожидает ввода дробного числа, а пользователь ввел строку, которая дробным числом не является).

В табл. 1.2 приведены типичные ошибки времени выполнения программы.

Таблица 1.2. *Типичные ошибки времени выполнения программы*

Сообщение	Ошибка	Вероятная причина
Error 106: Invalid numeric format	Неверный формат числа	Программа ожидает ввода целого числа, а пользователь ввел дробное число При вводе дробного числа в качестве разделителя целой и дробной частей числа пользователь вместо точки ввел запятую
Error 200: Division by zero	Деление на ноль	Второй операнд (делитель) оператора деления равен нулю

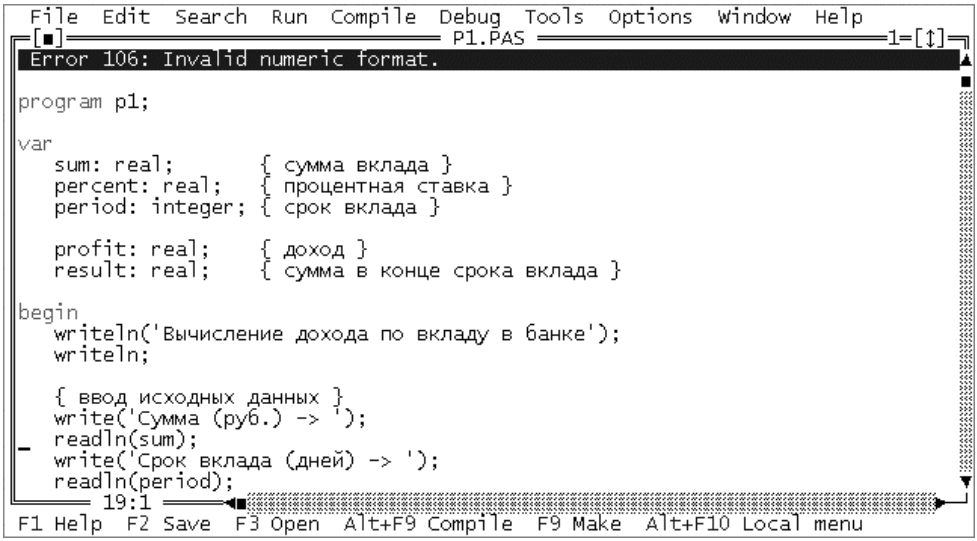


Рис. 1.14. Курсор показывает инструкцию, при выполнении которой произошла ошибка

Создание exe-файла

Чтобы иметь возможность запустить программу из операционной системы, нужно создать исполняемый (exe) файл программы — установить режим компиляции "на диск" (см. *параграф "Компиляция"* этой главы) и выполнить повторную компиляцию программы (в меню **Compile** выбрать команду **Compile**). Созданный компилятором exe-файл будет помещен в каталог, имя которого указано в поле **EXE&TPU Directory** окна **Directories** (это окно становится доступным в результате выбора в меню **Options** команды **Directories**), или, если каталог не задан, в каталог, в котором находится компилируемый pas-файл.

Завершение работы с Turbo Pascal

Чтобы завершить работу с Turbo Pascal, надо в меню **File** выбрать команду **Exit**. Если программа еще не была сохранена на диске или с момента сохранения в нее были внесены изменения, то на экране появится сообщение и вопрос о необходимости сохранить изменения (рис. 1.15). Чтобы сохранить изменения, надо сделать щелчок на кнопке **Yes**. Щелчок на кнопке **Cancel**

отменяет команду завершения работы с Turbo Pascal (в результате вновь становится доступным окно редактора кода).

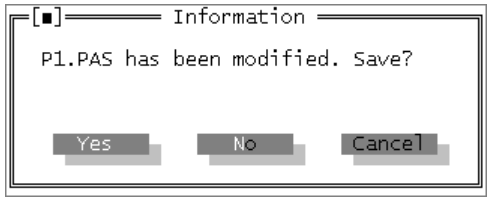


Рис. 1.15. Чтобы сохранить изменения, надо сделать щелчок на кнопке **Yes**

Внесение изменений в программу

Чтобы изменить программу, ее надо загрузить в редактор кода, внести необходимые изменения и выполнить компиляцию (предварительно установив режим компиляции на диск). Загрузка текста программы в редактор кода выполняется следующим образом. Сначала в меню **File** надо выбрать команду **Open**. В появившемся окне **Open a File** следует выбрать программу (PAS-файл программы), которую надо изменить, и щелкнуть на кнопке **Open** (рис. 1.16). Следует обратить внимание, что в списке **Files** окна **Open a File** отображаются имена PAS-файлов, которые находятся в *рабочем* каталоге (имя рабочего каталога отображается в поле **Рабочая папка** вкладки **Программа** окна свойств ярлыка, используемого для запуска Turbo Pascal).

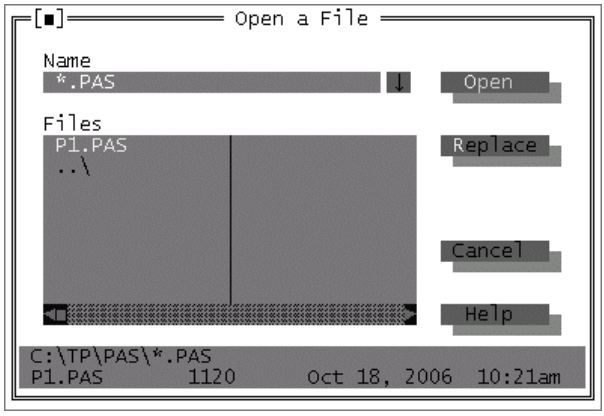


Рис. 1.16. Окно **Open a File**

Запуск программы из операционной системы

Чтобы запустить созданную в Turbo Pascal программу из операционной системы, надо раскрыть окно **Запуск программы** и в поле **Открыть** ввести полное (то есть указать путь) имя exe-файла программы (рис. 1.17).

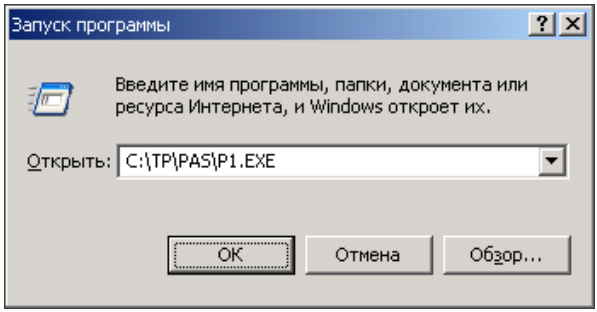
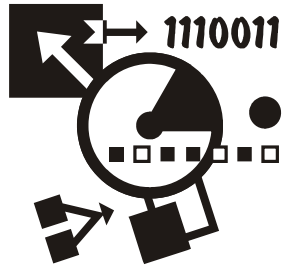


Рис. 1.17. Запуск программы Доход по вкладу из Windows

Задание

1. Установите Turbo Pascal на компьютер.
2. В каталоге C:\TP создайте каталог PAS.
3. На рабочем столе создайте ярлык, обеспечивающий запуск Turbo Pascal, и выполните его настройку: задайте, что рабочим каталогом является каталог C:\TP\PAS.
4. Запустите Turbo Pascal, наберите программу вычисления дохода по вкладу в банке (см. листинг 1.1), сохраните программу в каталоге C:\TP\PAS.
5. Выполните компиляцию программы.
6. Запустите программу вычисления дохода по вкладу, убедитесь, что она работает правильно (введите: сумма — 1000, срок вклада — 365); результат должен быть: доход: 80 руб., сумма в конце срока вклада: 1080 руб.).
7. Завершите работу с Turbo Pascal.
8. * Внимательно изучите листинг программы вычисления дохода по вкладу и "по аналогии" напишите программу пересчета цены из долларов в рубли.



Глава 2

Введение в программирование

Программа, работающая на компьютере, часто отождествляется с самим компьютером, так как человек, использующий программу, "вводит в компьютер" исходные данные с клавиатуры, и "компьютер выдает результат" на экран. На самом деле преобразование исходных данных в результат выполняет процессор.

Процессор выполняет обработку данных в соответствии с программой — последовательностью команд, составленной программистом. Таким образом, чтобы компьютер выполнил некоторую работу, необходимо разработать последовательность команд, обеспечивающую выполнение этой работы, или, как говорят, написать программу.

Этапы разработки программы

Выражение *написать программу* отражает только один из этапов создания компьютерной программы, когда разработчик программы (программист) действительно пишет команды (инструкции) на бумаге или в редакторе текста.

Программирование — это процесс создания программы, который можно представить как последовательность следующих этапов:

- ❒ Определение требований к программе (спецификация)
- ❒ Разработка алгоритма
- ❒ Написание команд (кодирование)
- ❒ Отладка
- ❒ Тестирование

Определение требований к программе

Определение требований к программе — один из важнейших этапов. На этом этапе подробно описывается исходная информация и формулируются требования к результату. Кроме того, описывается поведение программы в особых случаях.

Разработка алгоритма

На этапе разработки алгоритма необходимо определить последовательность действий, которые надо выполнить для достижения результата. Многие задачи можно решить различными способами. В этом случае программист, используя некоторый критерий, должен выбрать лучший. Результатом этапа разработки алгоритма должен быть алгоритм, представленный в виде словесного описания или блок-схемы.

Кодирование

После того как будет разработан алгоритм решения задачи, можно приступить к кодированию — записи алгоритма на языке программирования. Результатом этого этапа является исходная, т. е. представленная на языке программирования, программа.

Отладка

Отладка — процесс проверки работоспособности программы, поиска и устранения ошибок. Ошибки бывают двух типов: синтаксические (ошибки в тексте) и алгоритмические. Синтаксические ошибки — это наиболее легко устраняемые ошибки. Алгоритмические ошибки обнаружить труднее. Этап отладки можно считать завершенным, если программа правильно работает на одном-двух наборах исходных данных, если результат, полученный при использовании программы, совпадает с результатом, полученным методом "ручного" счета.

Тестирование

Этап тестирования особенно важен, если вы предполагаете, что вашей программой будут пользоваться другие. На этом этапе следует проверить, как ведет себя программа на как можно большем количестве входных наборов данных, в том числе и на заведомо неверных. Например, следует проверить, как ведет себя программа вычисления дохода по вкладу, если сумма вклада меньше, равна или больше пяти тысяч.

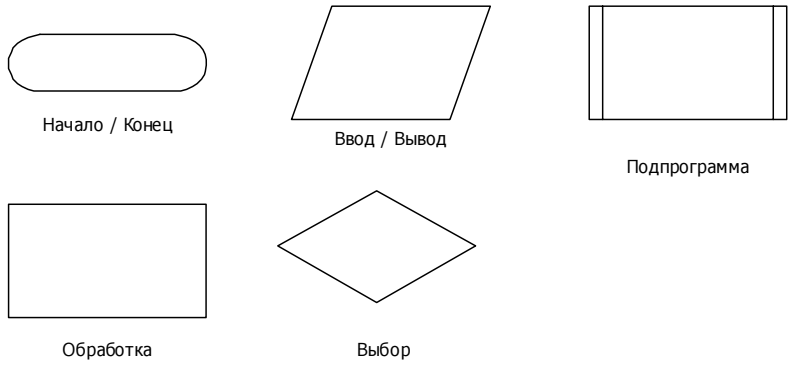


Рис. 2.1. Основные элементы для изображения блок-схем алгоритмов

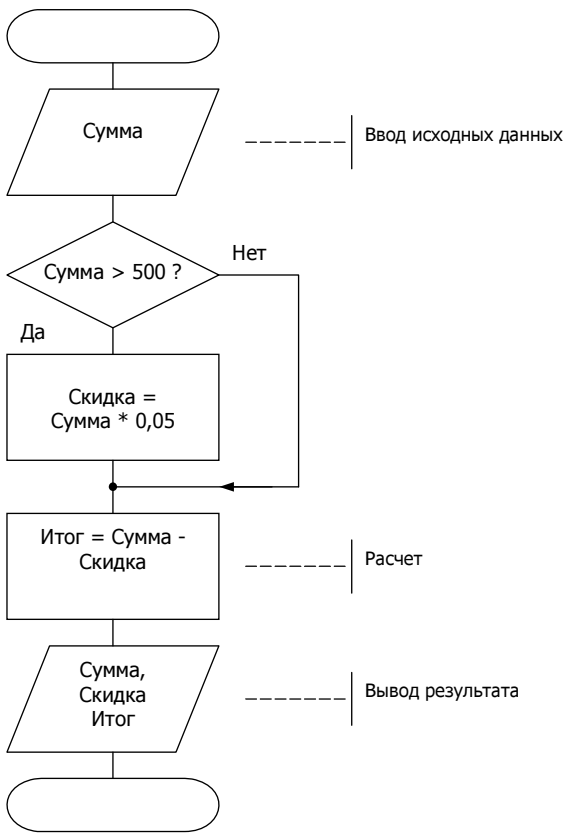


Рис. 2.2. Блок-схема алгоритма вычисления суммы покупки с учетом скидки

Алгоритм

На одном из первых этапов создания программы программист должен определить последовательность действий, которые необходимо выполнить, чтобы решить поставленную задачу, или разработать *алгоритм* — точное предписание, которое определяет процесс перехода от исходных данных к результату.

Существует несколько способов представления алгоритмов. На практике наиболее широко используют графический способ представления алгоритмов — блок-схемы. В блок-схемах для обозначения логически различных элементов программы используются стандартные символы, некоторые из них приведены на рис. 2.1.

Представление алгоритма в виде блок-схемы позволяет наглядно отразить последовательность действий, которые надо выполнить для решения поставленной задачи.

В качестве примера на рис. 2.2 приведен алгоритм вычисления суммы покупки с учетом скидки.

Программа

После разработки алгоритма решения задачи можно перейти к *кодированию* — составлению программы.

Программа — это последовательность инструкций (иногда вместо термина инструкция используют термины "оператор" или "команда"), записанных в соответствии с правилами языка программирования. В качестве примера в листинге 2.1 (текст программы принято называть листингом) приведена программа вычисления суммы покупки с учетом скидки.

Листинг 2.1. Программа вычисления суммы покупки с учетом скидки (p2_1.pas)

```
program p2_1;
var
   sum: real;       { сумма покупки}
   discount: real;  { скидка}
   total: real;     { к оплате }
begin
   writeln('*** Сумма покупки с учетом скидки ***');
   write('Сумма покупки (руб.) ->');
   readln(sum);
```

```
   if sum >= 500
      then discount := sum * 0.05;
   total := sum - discount;

   writeln;
   writeln('------------------------------');
   writeln('Сумма покупки: ',sum:6:2,' руб.');
   writeln('Скидка: ',discount:6:2,' руб.');
   writeln('К оплате: ',total:6:2,' руб.');

   writeln('Для завершения работы программы нажмите <Enter>');
   readln;
end.
```

Компиляция

Программа, представленная на языке программирования, называется исходной. Она состоит из инструкций, понятных человеку, но не понятных процессору. Чтобы процессор смог выполнить работу в соответствии с инструкциями исходной программы, исходная программа должна быть переведена на машинный язык — язык команд процессора. Решение этой задачи обеспечивает специальная программа — компилятор. Схема работы компилятора приведена на рис. 2.3. В процессе работы компилятор сначала проверяет программу на отсутствие синтаксических ошибок, затем, если в программе нет ошибок, создает (генерирует) выполняемую программу — машинный код. Следует отметить, что генерация машинного кода свидетельствует только об отсутствии в тексте программы синтаксических ошибок. Убедиться в правильности работы программы можно только во время ее тестирования — пробных запусков программы и анализа полученных результатов.

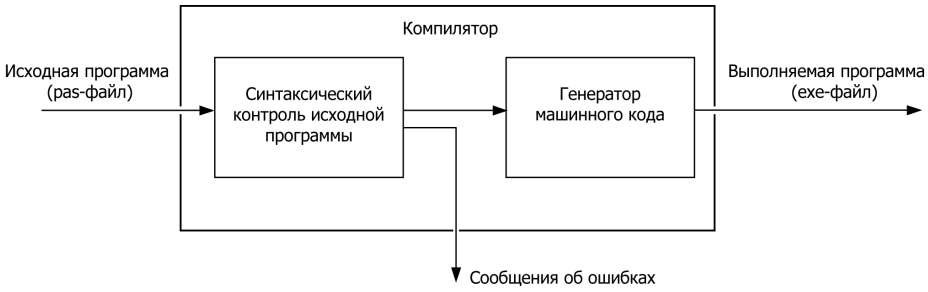


Рис. 2.3. Схема работы компилятора

Тип данных

Программа может оперировать с данными различного типа: целыми и вещественными (дробными) числами, символами, строками символов и логическими величинами.

Целый тип

В языке Pascal определены пять целых типов: `shortint`, `integer`, `longit`, `byte` и `word` (табл. 2.1). На практике наиболее широко используется тип `integer`.

Таблица 2.1. *Целые типы*

Тип	Диапазон	Размер (в байтах)
`shortint`	-128 .. 127	1
`integer`	-32768 .. 32767	2
`longint`	-2147483648 .. 2 147483647	4
`byte`	0 .. 255	1
`word`	0 .. 65535	2

Вещественный тип

В языке Pascal определены четыре вещественных типа: `real`, `single`, `double` и `extended` (табл. 2.2). Самым универсальным является тип `real`.

Таблица 2.2. *Вещественные типы*

Тип	Диапазон	Точность	Размер (в байтах)
`real`	$2.9 \cdot 10^{-39}$ ... $1.7 \cdot 10^{38}$	11-12	6
`single`	$1.5 \cdot 10^{-45}$ ... $3.4 \cdot 10^{38}$	7-8	4
`double`	$5.0 \cdot 10^{-324}$ ... $1.7 \cdot 10^{308}$	15-16	8
`extended`	$3.4 \cdot 10^{-4932}$ ... $1.1 \cdot 10^{4932}$	19-20	10

Символьный тип

Символьный тип (`char`) — это совокупность *символов* — букв русского и латинского алфавитов, цифр и других знаков.

Строковый тип

Строковый тип (`string`) — совокупность всевозможных строк, состоящих не более чем из 255 символов.

Логический тип

Логический (`boolean`) тип — это совокупность двух логических значений: `True` (истина) и `False` (ложь).

Переменная

Переменная — это область памяти компьютера, в которой находятся данные: исходные, промежуточные и результат.

Объявление переменной

Для того чтобы программа могла манипулировать данными, программист должен зарезервировать для этих данных место в оперативной памяти компьютера. Резервирование места для данных (исходных, промежуточных, результата) выполняется путем *объявления* переменных.

Инструкция объявления переменной в общем виде выглядит так:

Переменная`: `*тип*`;`

где:

Переменная — имя переменной, которое используется для доступа к данным;

тип — тип данных, хранение которых обеспечивает переменная.

Примеры:

```
n: integer;
profit: real;
FirstName: string;
found: boolean;
```

Если в программе несколько переменных одного типа, то объявить их можно в одной инструкции, разделив имена переменных запятыми.

Пример:

```
profit, percent, total: real;
```

Объявляя переменную, программист должен придумать для нее имя (идентификатор). В языке Pascal в качестве имени переменной можно использовать последовательность символов, состоящую из букв латинского алфавита