Предисловие

Для кого предназначена эта книга

Эта книга написана для тех, кто желает не просто научиться программировать на языке Паскаль, а стремится использовать это знание для решения конкретных задач.

Внимательно прочитав (а лучше изучив) материал этой книги, многие с удивлением обнаружат, что важно не столько умение правильно записывать операторы языка программирования, сколько искусство расставлять их в правильной последовательности. И именно в этом состоит истинное умение программировать. И именно это является истинным проявлением интеллекта.

Умение решить задачу есть ни что иное, как четкое представление пути достижения ее решения в виде последовательности отдельных, относительно самостоятельных шагов. Каждый из них логически следует из предыдущих и является отправной точкой для последующих. Это и есть алгоритм! В свою очередь алгоритм, записанный по компьютерным правилам, представляет собой программу.

О чем эта книга

Книга посвящена основам алгоритмизации обработки структур данных: от обычных вычислений — к обработке простейших упорядоченных совокупностей данных, предусмотренных в стандартном языке Паскаль. Речь пойдет о массивах различной размерности, строках, множествах и текстовых файлах.

Прежде всего читатель получит фундаментальные знания об особенностях строения перечисленных структур. В книге серьезное внимание уделяется типичным методам и задачам их обработки с использованием хорошо подобранных примеров. Здесь читатель встретится также с модульным программированием, использованием динамической памяти, рекурсией и т.д.

Не обойдены вниманием и общие проблемы, в том числе касающиеся терминологии и классификации структур данных. Авторы книги убеждены в исключительной важности этих вопросов. То, что преподносится читателю, должно систематизировать его знания, а не вводить в заблуждение. В противном случае мы с вами будем считать обычные массивы “списками последовательного хранения”. К сожалению, приведенный пример терминологического казуса нередок даже в преподавательской среде.

В основу обучения программированию обработки структур данных мы положили широкое использование процедур и функций. Это полезно как само по себе, так и в плане дальнейшего перехода к объектному программированию. При этом большое внимание уделяется решению так называемых комбинированных задач, предполагающих применение типичных элементарных алгоритмов при построении более сложных алгоритмов.

Рекомендации для преподавателей

Наш опыт показывает, что обучать компьютерной грамотности — почти бесполезная трата времени. Тот, кто не общается с компьютером регулярно, очень быстро забывает школьные уроки информатики подобного рода. Тем более, что стремительный прогресс сводит к нулю их актуальность уже через год-два. А тот, кому компьютер нужен для решения практических задач, азы компьютерной грамотности на школьном уровне постигает вполне самостоятельно, причем гораздо быстрее и глубже, очень и очень далеко обгоняя своих учителей. Благо компьютерной литературы для "чайников" в наше время более чем достаточно. Пожалуй, единственная ценность такого школьного обучения состоит в получении некоторого навыка работы на клавиатуре.

Авторы считают, что цель школьного обучения, в том числе и на уроках информатики, — развитие интеллекта учащегося. Трудно представить себе лучшее средство для этого, чем правильное обучение программированию.

Общеизвестно, что обучение программированию в школе в настоящее время находится на нулевом уровне. И совсем не случайно, что так называемые олимпиады по информатике — единственные из всех, которые проводятся по предмету, в школе обычно не изучаемому. Имея многолетний и непрерывный опыт общения с учащимися в школе, с учителями в процессе повышения квалификации, со студентами — будущими учителями информатики, авторы пришли к выводу, что одной из причин существования такого уродливого явления, как "хаотическое программирование", является отсутствие учебников, в которых последовательно и планомерно был бы представлен весь процесс обучения, начиная с азов.

Поэтому авторы книги искренне надеются, что она будет использована учителями информатики. Но в то же время они и серьезно сомневаются в этом, так как объем книги явно не укладывается в отведенное на это школьной программой время.

Обойдемся без блок-схем!

Следуя корифеям, авторы считают практически бессмысленным использование блок-схем на этапе алгоритмизации процессов решения задач. Громоздкость и трудоемкость построения делают иллюзорной их "наглядность" даже на уровне

школьных алгоритмов средней сложности. Все "прелести" программирования с использованием блок-схем алгоритмов один из авторов испытал лично, работая в свое время в конструкторском бюро компьютерных систем управления.

Этим впечатлением стоит поделиться. Один из этапов проектирования программных комплексов в соответствии с государственными стандартами требовал представления блок-схем алгоритмов, с тем чтобы на последующих этапах по ним могли быть составлены программы на том или ином языке программирования. Однако никому из нас даже в голову не приходило поступать подобным образом. Вместо пресловутых блок-схем, строить которые никто и не думал, чаще всего использовались обычные спецификации функций или словесные описания разрабатываемых алгоритмов. По ним затем и составлялись программы.

Самое смешное начиналось потом. Поскольку требования стандартов не могли быть нарушены, то для предъявления комиссии блок-схемы составлялись по готовым (!!!) и отлаженным (!!!) программам, а чаще всего просто перекладывались из одного проекта в другой.

В современных условиях применения персональных компьютеров процессы алгоритмизации неотделимы от программирования. Согласитесь, что программист, с упоением рисующий блок-схемы, сидя за клавиатурой, выглядит, мягко выражаясь, несерьезно.

Справедливости ради, отметим все же целесообразность использования блок-схем для повышения наглядности при демонстрации элементарных алгоритмов в младших классах.

Технические подробности

Изложение книги ориентировано на использование интегрированной среды программирования (ИСП) Borland (Turbo) Pascal 7.0. Чтобы обеспечить независимость своих программ от конкретных настроек соответствующих компиляторов, читателям рекомендуется практически всегда использовать для них предваряющий набор директив компиляции, например, в виде **`{$B+,D+,E+,I+,L+,N+,Q+,R+,X-}`**.

Назначение основных из них таково:

- **`$B+`** — установка полной схемы вычислений логических выражений;
- **`$D+`** — создание отладочной информации в процессе компиляции;
- **`$E+`** — разрешение эмуляции; наличие сопроцессора определяется автоматически; он используется в случае его обнаружения, в противном же случае его работа эмулируется;
- **`$I+`** — автоматический контроль правильности выполнения операций ввода-вывода;
- **`$L+`** — создание отладочной информации не только для глобальных, но и для локальных переменных;

- **$N+** — ориентирование компилятора на выполнение операций с вещественными числам или с помощью сопроцессора (если он есть), или посредством эмуляции (если сопроцессор отсутствует);
- **$Q+** — автоматический контроль переполнения при выполнении арифметических операций;
- **$R+** — автоматический контроль выхода за допустимые границы, предусмотренные типами;
- **$X-** — запрет расширенного синтаксиса, в том числе запрет обращения к функциям как к процедурам;
- **$F+** — установка дальней модели памяти;
- **$S+** — автоматический контроль переполнения стека.

Более подробно эти директивы рассматриваются в книге по ходу изложения.

Как хранить тексты программ

Настоятельно рекомендуем читателю обратить внимание на возможность хранить тексты нужных ему программ в виде единого файла приложения Word. Это существенно эффективнее, нежели иметь множество PAS-файлов. Так вам будет удобнее их комментировать и легче просматривать. К программам можно будет составить оглавление для упрощения и ускорения поиска. И наконец, вы избавитесь от необходимости придумывать бесконечное количество имен файлов, не путаясь при этом.

Задача переноса текста нужной программы из окна открытого текстового файла сборника программ в окно редактора ИСП решается достаточно просто:

- запустите ИСП и установите оконный режим его работы, нажав клавиши <Alt+Enter>;
- в ИСП выберите команду **Options⇨Environment⇨Editor**, после чего в диалоговом окне **Editor Options** сбросьте флажок **Auto indent mode** и щелкните на кнопке **OK**;
- откройте в ИСП пустое окно, выбрав команду **File⇨New**;
- откройте текстовый файл со сборником программ с помощью приложения Word, найдите текст нужной программы, выделите его обычным образом и скопируйте в буфер обмена, нажав клавиши <Ctrl+Insert>;
- щелкните правой кнопкой мыши на заголовке окна ИСП, в появившемся контекстном меню выберите команду **Изменить**, а затем в раскрывшемся каскадном меню щелкните левой кнопкой мыши на команде **Вставить**;
- сохраните в ИСП текст программы обычным образом.

Чтобы осуществить перенос текста программы из ИСП в открытый документ приложения Word, необходимо поступить следующим образом:

- установите оконный режим работы ИСП, нажав клавиши <Alt+Enter>;
- щелкните правой кнопкой мыши на заголовке окна ИСП, в раскрывшемся контекстном меню выберите команду **Изменить**, а затем в раскрывшемся каскадном меню щелкните левой кнопкой мыши на команде **Пометить**;
- удерживая левую кнопку мыши в нажатом состоянии, выделите текст программы (или нужный его фрагмент), после чего нажмите клавишу <Enter>;
- активизируйте окно документа приложения Word, установите текстовый курсор в нужное место и нажмите клавиши <Shift+Insert>;
- выделите в документе вставленный текст программы и установите для него шрифт **Courier New**.

Условные обозначения

В книге принята следующая система обозначений программ, реализующих те или иные алгоритмы:

- **L** — линейные алгоритмы;
- **V** — алгоритмы с разветвлениями;
- **C** — циклические алгоритмы;
- **P** — алгоритмы обработки одномерных массивов;
- **D** — алгоритмы обработки двумерных массивов;
- **S** — алгоритмы обработки строк;
- **R** — рекурсивные алгоритмы;
- **M** — алгоритмы обработки множеств;
- **H** — алгоритмы поиска;
- **U** — алгоритмы сортировки;
- **F** — алгоритмы обработки текстовых файлов.

Благодарности

Авторы благодарны своему сыну Леониду, который, несмотря на загруженность уроками, разработал множество программ, приведенных в этой книге, в том числе и по модификации модуля **Crt**.

Авторы благодарны ему также и за то, что он творчески воспринял все методические идеи, представленные в данной книге. Авторы выражают надежду, что в самое ближайшее время в своих новых книгах им удастся отразить те достиже-

ния, которые привели его к успеху на республиканских школьных олимпиадах по информатике всех уровней.

Авторы также заранее благодарны всем заинтересованным читателям, которые направят свои замечания, пожелания и указания на допущенные ошибки по адресу: **a_n_m@rambler.ru**.

От издательства "Диалектика"

Вы, читатель этой книги, и есть главный ее критик. Мы ценим ваше мнение и хотим знать, что было сделано нами правильно, что можно было сделать лучше и что еще вы хотели бы увидеть изданным нами. Нам интересны любые ваши замечания в наш адрес.

Мы ждем ваших комментариев и надеемся на них. Вы можете прислать нам бумажное или электронное письмо либо просто посетить наш Web-сервер и оставить свои замечания там. Одним словом, любым удобным для вас способом дайте нам знать, нравится ли вам эта книга, а также выскажите свое мнение о том, как сделать наши книги более интересными для вас.

Отправляя письмо или сообщение, не забудьте указать название книги и ее авторов, а также свой обратный адрес. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию новых книг.

Наши электронные адреса:

E-mail: **info@dialektika.com**
WWW: **http://www.dialektika.com**

Наши почтовые адреса:

в России: 115419, Москва, а/я 783
в Украине: 03150, Киев, а/я 152