

УДК 681.3.06(075)
ББК 22.18
А47

Рецензенты:

кафедра «Автоматизированные системы управления» МАДИ (ГТУ)
(зав. кафедрой, д-р техн. наук, проф. *А.Б. Николаев*);
д-р техн. наук, проф. *И.П. Норенков*;
канд. техн. наук, доц. *Г.С. Иванова*

Алексеев Ю.Е., Ваулин А.С., Куров А.В.

А47 Практикум по программированию: Обработка числовых данных: Учеб. пособие / Под ред. Б.Г. Трусова. — М.: Изд-во МГТУ им. Н.Э. Баумана, 2008. — 288 с.

ISBN 978-5-7038-3159-5

Приведены обработка числовых типов данных, краткие теоретические сведения, примеры программ реализации изучаемых алгоритмов, а также задания для выполнения лабораторных работ по каждой из рассматриваемых в курсе тем.

Пособие ориентировано на среду программирования Delphi и содержит большое количество важных алгоритмов решения инженерных задач. Представлены полные комплекты заданий (не менее 25 вариантов), имеющих разнообразный характер, но одинаковый уровень сложности.

Материал пособия авторы используют при проведении практических занятий в МГТУ им. Н.Э. Баумана.

Для студентов первого курса машино- и приборостроительных специальностей. Может быть полезно преподавателям как сборник заданий при проведении лабораторных работ.

УДК 681.3.06(075)
ББК 22.18

ISBN 978-5-7038-3159-5

© Алексеев Ю.Е., Ваулин А.С.,
Куров А.В., 2008
© Оформление. Изд-во МГТУ
им. Н.Э. Баумана, 2008

Оглавление

Введение	6
1. Краткая справка по языку Object Pascal и разработке консольных приложений в среде Delphi	9
2. Программы линейной структуры	21
2.1. Средства разработки программ линейной структуры	21
Целые типы данных	21
Вещественные типы данных	23
Стандартные функции для обработки числовых данных	24
Арифметические выражения	25
Оператор присваивания	26
Ввод данных с клавиатуры	27
Вывод данных в окно программы	28
2.2. Приемы, используемые для минимизации вычислений	31
2.3. Примеры выполнения задания	32
2.4. Задания А для самостоятельной работы	36
2.5. Задания Б для самостоятельной работы	39
3. Программы разветвляющейся структуры	44
3.1. Средства разработки программ разветвляющейся структуры ..	44
Условные операторы	44
Булев тип	46
3.2. Примеры выполнения задания	48
3.3. Задания для самостоятельной работы	51
4. Программы циклической структуры	57
4.1. Средства разработки программ циклической структуры	57
Цикл с параметром	57
Цикл с предусловием	58
Цикл с постусловием	59
4.2. Вычисление и вывод данных в виде таблицы	60
4.3. Пример выполнения задания с использованием цикла while ..	62
4.4. Пример выполнения задания с использованием цикла for	64
4.5. Задания для самостоятельной работы	65
4.6. Сохранение результатов вычислений в массиве	78
4.7. Пример выполнения задания	80
4.8. Задания для самостоятельной работы	83
4.9. Приемы вычисления сумм, произведений и экстремальных значений	86
Вычисление суммы и произведения	86
Нахождение наибольшего или наименьшего значения	87
4.10. Пример выполнения задания А	90
4.11. Задания А для самостоятельной работы	92
4.12. Пример выполнения задания Б	95
4.13. Задания Б для самостоятельной работы	96
4.14. Вычисление суммы бесконечного ряда с заданной точностью	99

4.15. Вывод рекуррентной формулы для вычисления члена ряда ...	100
Способы вычисления значения члена ряда	101
4.16. Примеры выполнения задания	102
4.17. Задания для самостоятельной работы	105
4.18. Уточнение корней уравнений	111
Метод простых итераций	111
Метод половинного деления	115
Метод касательных	116
4.19. Пример выполнения задания	117
4.20. Задания для самостоятельной работы	119
4.21. Вычисление определенных интегралов	121
4.22. Пример выполнения задания	125
4.23. Задания для самостоятельной работы	127
5. Организация программ со структурой вложенных циклов	129
5.1. Вычисление определенного интеграла с заданной точностью	132
5.2. Задания для самостоятельной работы	137
5.3. Вычисление наибольшего (наименьшего) значения функции с заданной точностью на заданном интервале	138
5.4. Задания для самостоятельной работы	144
5.5. Обработка матриц	146
5.6. Примеры выполнения задания на обработку матриц	150
5.7. Задания для самостоятельной работы	158
5.8. Методы сортировки массивов	163
Метод включения с сохранением упорядоченности (метод прямого включения или сортировка вставками)	163
Метод прямого обмена (метод пузырька)	165
Метод прямого выбора (сортировки посредством выбора) и его модификации	166
5.9. Пример выполнения задания	168
5.10. Задания для самостоятельной работы	169
6. Программирование с использованием подпрограмм	172
6.1. Процедуры	172
6.2. Пример выполнения задания	178
6.3. Задания для самостоятельной работы	180
6.4. Функции	184
6.5. Пример выполнения задания	188
6.6. Задания для самостоятельной работы	191
6.7. Рекурсивные подпрограммы	196
6.8. Пример выполнения задания на составление рекурсивной подпрограммы	199
6.9. Задания для самостоятельной работы	202
6.10. Дополнительные сведения о подпрограммах и массивах	208
Параметры — открытые массивы	209
Динамические массивы	212
Перегружаемые подпрограммы	215
Параметры со значениями по умолчанию	216
Примеры организации программ с подпрограммами	217

7. Модули пользователей	225
7.1. Создание и использование модулей	225
7.2. Пример выполнения задания	240
7.3. Задания для самостоятельной работы	251
7.4. Пример выполнения задания	258
7.5. Задания для самостоятельной работы	265
Приложение 1. Функция перекодировки кириллицы	273
Приложение 2. Дополнительные сведения по обработке исключений	274
Приложение 3. Основные формулы, используемые при решении геометрических задач	280
Список литературы	282
Алфавитный указатель	283

Введение

В связи с ростом роли информатики в жизни современного общества больше внимания уделяется и преподаванию этой дисциплины в вузах. По сложившейся традиции значительное место в курсе информатики в технических университетах занимает раздел, связанный с изучением языков программирования и разработкой на изучаемом языке алгоритмов решения важнейших инженерных задач.

Изучение алгоритмов решения основных инженерных задач (характерных приемов программирования) рассматривается как база для дальнейшего освоения дисциплины — студенты учатся работать с различными типами и структурами данных, создавать алгоритмы решения разнообразных задач. Конкретный язык программирования, на котором реализуются рассматриваемые алгоритмы, выступает в этом случае как конкретный инструмент для практического воплощения основных теоретических положений.

В ходе последующего изучения дисциплины при решении более сложных задач и обработке разных типов и структур данных объясняется значение простейших алгоритмов как своего рода строительных блоков, на базе которых разрабатывают алгоритм решения поставленной задачи. Уяснив постановку задачи и разрабатывая алгоритм ее решения, студенты должны выделить основные этапы решения, которые чаще всего представляют собой рассмотренные ранее приемы программирования и алгоритмы. Разработка и реализация алгоритмов решения задач позволяет попутно добиться еще одного важного результата — формирования основ логического мышления.

Многолетняя практика преподавания информатики первокурсникам свидетельствует о том, что уровень начальной подготовки студентов различается существенным образом, у многих из них отсутствуют умения и навыки логического построения алгоритма решения поставленной задачи. Проблема усугубляется в дальнейшем еще и тем, что в силу разных причин студенты должны образом не осваивают раздел, посвященный разработке и реализации основных типов алгоритмов и характерных приемов программирования.

Цель данного учебного пособия — изложить в краткой форме основные принципы, правила построения и программирования алгоритмов различных типов (линейных, разветвляющихся, циклических, в том числе и с вложенными циклами), а также характерных приемов; показать возможности этих алгоритмов при решении практических задач. Теоретический материал сопровождается примерами программ, а также заданиями для выполнения лабораторных работ. В отличие от подобных пособий авторы стремились составить более сложные и интересные задания, в рамках одной темы выдержать одинаковый уровень сложности для разных вариантов. Систематическое выполнение предлагаемых заданий позволит студентам успешно решать задачи, предлагаемые при проведении рубежного контроля и на экзамене.

В пособии рассматривается программирование основных алгоритмических структур: линейной, разветвляющейся, циклической (циклы с заранее известным и неизвестным числом повторений). Изложены принципы организации простых и вложенных циклов, а также следующие характерные приемы программирования: 1) нахождение сумм и произведений; 2) нахождение минимального и максимального значения функции на заданном интервале; 3) вычисление суммы бесконечного ряда и уточнение корней уравнений; 4) запоминание результатов вычислений; 5) сортировка элементов массива (по убыванию или возрастанию).

Реализация указанных простейших алгоритмов требует умения программировать основные алгоритмические структуры и уже на базе этих простых алгоритмов решать более сложные инженерные задачи. Так, определение суммы позволяет вычислить значение определенного интеграла; умение найти значение интеграла и организовать вложенные циклы — вычислить значение интеграла с заданной точностью; умение находить минимальный (максимальный) элемент массива в сочетании с программированием вложенных циклов — отсортировать массивы и т. д.

В заключительном разделе пособия рассмотрено использование подпрограмм. Приведены основные правила организации подпрограмм-функций и подпрограмм-процедур, виды параметров и механизмы их передачи в подпрограммы. Изложены возможности языка программирования по созданию библиотек подпрограмм, одним из средств разработки которых является механизм модулей, а также правила организации модулей и примеры их использования при разработке достаточно объемных приложений.

В качестве инструментального средства программной реализации алгоритмов применяется язык Pascal, который изучается на большинстве специальностей университета. Авторы сочли необходимым включить раздел, содержащий основные сведения о среде программирования Delphi и подготовке в ней консольных приложений, поскольку именно эта среда используется при изучении курса «Информатика».

1. Краткая справка по языку Object Pascal и разработке консольных приложений в среде Delphi

Среда программирования Delphi предоставляет возможность разработки и отладки различных программных продуктов, в том числе приложений, работающих как с использованием графического интерфейса пользователя, так и в консольном режиме. Последний имеет интерфейс пользователя в виде текстового окна, называемого *окном программы*, в котором последовательно, строка за строкой, отображаются данные, вводимые пользователем с клавиатуры и выводимые программой. Позицию начала ввода или вывода в окне программы указывает *курсор* — мигающий символ, имеющий вид подчеркивания в режиме вставки или прямоугольника — в режиме замены. По умолчанию длина строки равна 80 знакам, а количество строк — 50. Изменить эти и другие параметры окна программы позволяет диалоговое окно, открывающееся при вводе команды *Свойства* в системном меню.

При вводе пользователь имеет возможность редактировать последние вводимые данные, используя символьные клавиши, а также клавиши BackSpace (удаление последнего введенного символа), Delete (удаление символа справа от курсора), Insert (переключение режимов вставки и замены), Стрелка вверх (удаление всех введенных символов), Стрелка влево (перемещение курсора в предыдущую позицию), Стрелка вправо (перемещение курсора в следующую позицию). Если в диалоговом окне команды *Свойства* установить на вкладке *Общие* флажок *Выделение мышью*, то становится возможным выделять части текста буксировкой мыши, копировать выделенное в буфер обмена щелчком правой клавиши и затем вставлять в позицию курсора щелчком правой клавиши. Завершается ввод нажатием клавиши Enter, при этом курсор перемещается в начало новой строки. Максимальная длина вводимой последовательности символов равна 254.

Вывод данных из программы выполняется в виде текста, символ за символом при автоматическом перемещении курсора в очередную позицию строки, а при достижении ее конца — в начало новой строки.

Консольный режим обычно используется, когда необходимо минимизировать время счета и расход оперативной памяти. Кроме того, консольный режим удобен для быстрой проверки и

2. Программы линейной структуры

Программой линейной структуры называют такую, каждый оператор которой выполняется один и только один раз. Она может строиться только из простых операторов, не меняющих естественный порядок вычислений: из операторов присваивания и процедур. Из числа последних в этом разделе нас будут интересовать только операторы процедур ввода и вывода для стандартных устройств — клавиатуры и монитора.

2.1. Средства разработки программ линейной структуры

Рассмотрение алгоритмизации задач и приемов программирования удобнее всего проводить на примерах обработки числовых данных. Рассмотрим в первую очередь *стандартные типы* (имеющиеся в Delphi и не требующие объявления в программе) числовых данных.

Целые типы данных

К числу стандартных целых типов относят:

Int64 — представляющий целые со знаком от -2^{63} до $+2^{63} - 1$, занимает 8 байт;

Integer — представляющий целые со знаком от -2147483648 до $+2147483647$, занимает 4 байта;

LongInt — эквивалентный типу **Integer**;

SmallInt — представляющий целые со знаком от -32768 до $+32767$, занимает 2 байта;

ShortInt — представляющий целые со знаком от -128 до $+127$, занимает 1 байт;

Byte — представляющий целые без знака от 0 до 255, занимает 1 байт;

Word — представляющий целые без знака от 0 до 65535, занимает 2 байта;

LongWord — представляющий целые без знака от 0 до 4294967295, занимает 4 байта;

Cardinal — эквивалентный типу **LongWord**.

Например, чтобы объявить переменные с именами **I** и **K** как переменные типа **Integer** и **N** как **Byte**, в программе следует записать

3. Программы разветвляющейся структуры

3.1. Средства разработки программ разветвляющейся структуры

Программой разветвляющейся структуры называют такую, в которой в зависимости от исходных данных возможны различные последовательности выполнения операторов, причем на любой последовательности каждый оператор выполняется только один раз.

Для реализации программ или фрагментов программ с разветвляющейся структурой используют *сложные* операторы¹: условные **if** и выбора **case**. В этом разделе ограничимся рассмотрением полной формы условного оператора — оператора **if then else** и его сокращенной формы — оператора **if then**.

В случае применения условных операторов ветвление алгоритма обусловлено проверками логических выражений (в языке Object Pascal их называют булевыми выражениями), результатом вычисления которых могут быть лишь два значения: *истина* и *ложь*. В условных операторах используются как простейшие булевы выражения, основанные на сравнении выражений других типов, так и сложные, с логическими операциями.

Условные операторы

Оператор **if then else** имеет синтаксическую диаграмму, показанную на рис. 3.1, где БВ — *булево выражение* (см. ниже), значением которого может быть либо *истина*, либо *ложь*. Оп1 и Оп2 — операторы, каждый из которых может быть пустым.

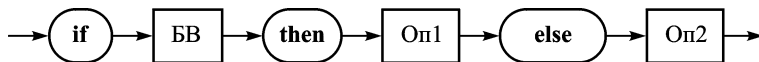


Рис. 3.1

При выполнении оператора **if then else** вначале вычисляется выражение БВ, и если результат *истина*, то — оператор Оп1, если результат имеет значение *ложь* — оператор Оп2.

¹ Сложные операторы включают в себя другие операторы.

4. Программы циклической структуры

4.1. Средства разработки программ циклической структуры

Программой циклической структуры называют такую, в которой операторы могут повторно, при изменяющихся значениях *переменных* выполняться несколько раз, образуя *цикл*. Различают следующие виды циклов (для их организации используют специальные сложные операторы — *операторы циклов*):

- *цикл с заданным числом повторений* или *цикл с параметром* (операторы цикла **for**: **for to** и **for downto**),
- *цикл с предусловием* (оператор цикла **while**),
- *цикл с постусловием* (оператор цикла **repeat until**).

В циклах можно выделить управляющие части, определяющие начало и условия выполнения, и части из одного или нескольких операторов (*тело*), выполняющие необходимые преобразования данных. Цикл называют *простым*, если в его теле нет других циклов.

Цикл с параметром

Структура оператора цикла **for to** описывается синтаксической диаграммой (рис. 4.1), где используют следующие обозначения: I — параметр цикла — переменная *ординального* (*порядкового*), в частности целого, типа; B1 и B2 — выражения того же типа, что и параметр цикла, или совместимые с ним; Оп — оператор, выполняемый внутри цикла.

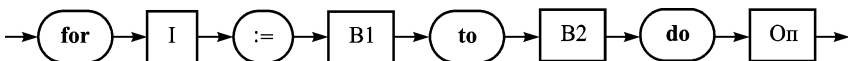


Рис. 4.1

Часть, предшествующая оператору Оп, — *заголовок цикла* — является управляющей, а сам оператор Оп — *телом цикла*. Оператор Оп будет последовательно выполняться при автоматическом увеличении с минимальным шагом значения параметра цикла I от B1 до B2 включительно (для целых типов шаг равен 1). При B1 > B2 оператор Оп не будет выполняться вообще.

5. Организация программ со структурой вложенных циклов

Структурой с вложенным циклом называют такую, в которой внутри одного цикла находится один или несколько других. Цикл, расположенный внутри другого, называют *внутренним*. Цикл, внутри которого находятся другие, называют *внешним*. Таким образом, один и тот же цикл может выступать и в роли внешнего (если он содержит внутри себя другие циклы), и в роли внутреннего (если он расположен внутри другого цикла). Правильная организация вложенного цикла состоит в том, что внутренний цикл должен целиком располагаться внутри внешнего.

Допустимыми являются следующие варианты организации вложенных циклов. Первый вариант вложенного цикла — внутри внешнего цикла последовательно расположено несколько внутренних:

```
for I:=1 to N do //внешний цикл
begin
    . . . . .
    for J:=1 to M do //первый внутренний цикл
    begin
        . . . . .
    end;      // конец первого внутреннего цикла
    . . . . .
    for K:=1 to L do //второй внутренний цикл
    begin
        . . . . .
    end;      // конец второго внутреннего цикла
    . . . . .
end;  // конец внешнего цикла
```

Частным случаем первого варианта организации вложенных циклов является следующий:

```
for I:=1 to N do
    for J:=1 to M do
        for K:=1 to L do
            A[I,J,K] := I*J*K;
```

В приведенном примере все циклы оканчиваются в одном и том же месте.

6. Программирование с использованием подпрограмм

Подпрограмму можно определить как относительно самостоятельный фрагмент программы, оформленный таким образом, что его можно выполнять многократно, передавая ему управление из разных частей программы для обработки разных данных.

Использование подпрограмм позволяет уменьшить размер программы (если в различных частях программы необходимо выполнять обработку данных по одному алгоритму) и сделать ее исходный текст более удобным для понимания процесса обработки данных (если алгоритм подпрограммы обладает функциональной законченностью, а имя подпрограммы отражает ее назначение, как, например, у стандартных подпрограмм $\text{Sin}(X)$ или $\text{Abs}(X)$). Преимущества использования подпрограмм проявляются также при разработке больших программ, так как становится возможным распараллелить процесс создания программного продукта, поручив разработку отдельных подпрограмм разным исполнителям, и, что более важно, — упростить процесс написания и отладки.

Разбиение программы на подпрограммы производится прежде всего по функциональному признаку: подпрограмма должна реализовывать одну, но законченную функцию. При этом надо стремиться к сокращению количества межпрограммных связей (количеству передаваемых параметров). Рекомендуемый размер подпрограммы составляет 10–60 строк текста. Нецелесообразно создавать слишком короткие подпрограммы, а размещение подпрограммы в пределах одной страницы позволит программисту охватить весь текст одним взглядом и не тратить время на переключение внимания с одной страницы на другую.

В языке Object Pascal используются два вида подпрограмм — функции и процедуры. При общих принципах оформления функции обладают дополнительными возможностями. Поэтому сначала рассмотрим объявление и использование процедур, а затем особенности функций.

6.1. Процедуры

При работе с подпрограммами следует различать термины *объявление подпрограммы* (*описание подпрограммы*) и *обращение к подпрограмме* (*вызов подпрограммы*).

7. Модули пользователей

7.1. Создание и использование модулей

Приложение Delphi помимо стандартных модулей может использовать модули, создаваемые пользователями. Каждый модуль пользователя является отдельно подготовленной и хранящейся в отдельном файле с расширением .pas программной единицей, которая может быть использована любой программой.

В Delphi реализован модульный принцип программирования, причем модули играют роль наборов заранее подготовленных и отлаженных подпрограмм, именованных констант, типов, переменных, которые могут использоваться в тех частях программы (в основной программе и модулях), куда они подключены с помощью *предложения использования*, показанного на рис. 7.1.

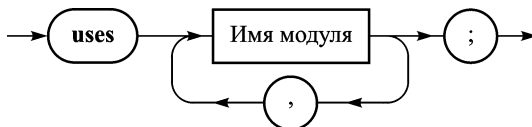


Рис. 7.1

Модули также могут содержать код, выполняемый до передачи управления в указанные части программы, и код, выполняемый после возврата управления из них.

Исходный текст модуля имеет следующую структуру:

unit <имя модуля>;	{заголовок модуля}
interface	{интерфейсная часть}
<предложение использования>;	
<объявление именованных констант>	
<объявление типов>	
<объявление переменных>	
<объявление заголовков подпрограмм>	
implementation	{часть реализации}
<предложение использования>;	
<объявление меток>	
<объявление именованных констант>	
<объявление типов>	

Список литературы

1. *Алексеев В.Е., Алексеев Ю.Е., Ваулин А.С.* и др. Практикум по программированию: Учеб. пособие по курсу «Вычислительная техника и информационная технология» / Под ред. Б.Г. Трусова. М.: Изд-во МГТУ им. Н.Э. Баумана, 1989. 132 с.
2. *Алексеев В.Е., Ваулин А.С., Петрова Г.Б.* Вычислительная техника и программирование. Практикум по программированию: Практ. пособие / Под ред. А.В. Петрова. М.: Высш. шк., 1991. 400 с.
3. *Архангельский А.Я.* Object Pascal в Delphi. М.: БИНОМ, 2002. 384 с.
4. *Блюмин А.Г., Гусев Е.В., Федотов А.А.* Численные методы. Метод. указания к лабораторным работам / Под ред. Г.А. Колотушкина. М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. 48 с.
5. *Выгодский М.Я.* Справочник по высшей математике. М.: Наука, 1976. 872 с.
6. *Иванова Г.С.* Основы программирования: Учебник. М.: Изд-во МГТУ им. Н.Э. Баумана, 2007. 415 с.
7. *Йенсен К., Вирт Н.* Паскаль: руководство для пользователя / Пер. с англ. А.Я. Архангельского. М.: БИНОМ, 2002. 384 с.
8. *Кнут Д.* Искусство программирования для ЭВМ. В 3 т. Т. 3. Сортировка и поиск: Пер с англ. М.: Мир, 1978. 844 с.
9. *Мак-Кракен Д., Дорн У.* Численные методы и программирование на Фортране: Пер. с англ. М.: Мир, 1977. 584 с.
10. *Смолянский М.Л.* Таблицы неопределенных интегралов. М.: Физматгиз, 1961. 108 с.
11. *Фаронов В.В.* Система программирования Delphi. СПб.: БХВ-Петербург, 2005. 912 с.

Алфавитный указатель

Б

Бесконечный ряд 99
Блок 13
Булево выражение 45, 47

В

Вектор 80
Вложенные циклы 129
Внешний цикл 129
Внутренний цикл 129
Входной параметр 174, 175
Вызов подпрограммы 172
Выходной параметр 175

Д

Двумерный массив 146
Дизъюнкция 46
Динамические массивы 212
Директива 11
Дополнительный параметр
цикла 59

З

Заголовок модуля 225
Заголовок программы 11
Заголовок процедуры 173
Заголовок функции 184
Заголовок цикла 57

И

Именованные константы 13
Индексная переменная 78
Интерфейсная часть 225
Исключения 274

К

Ключевые слова 11
Комментарий 11
Конъюнкция 46
Курсор 9

М

Массив 78
Массив одномерный 78
Матрицы 146
Метод касательных 116
Метод парабол 121
Метод половинного деления 115
Метод простых итераций 111
Метод прямого включения 164
Метод прямого выбора 166
Метод прямого обмена 165
Метод прямоугольников 121
Метод пузыря 165
Метод Симпсона 121
Метод трапеций 121
Модули пользователей 12, 225

Н

Накопление произведения 86
Накопление суммы 86
Начальные значения
переменных 16
Неравнозначность 46

О

Обработка исключений 274
Обращение к подпрограмме 174
Обращение к функции 184

Общая формула члена ряда 100
 Объявление функции 184
 Объявление подпрограммы 173
 Объявление типа двумерного массива 146
 Окно наблюдения 19
 Окно программы 9
 Окно редактирования программы 11
 Оператор выбора 44
 Операторные скобки 46
 Операторы 11
 Операторы циклов 57
 Описание подпрограммы 172
 Ординальный тип 57, 78
 Отладка 16
 Отрицание 46

П

Параметр — открытый массив 209
 Параметр цикла 57
 Параметр-константа 177
 Параметр-результат 177
 Параметры со значениями по умолчанию 216
 Параметры-значения 175
 Параметры-переменные 175
 Перегружаемые подпрограммы 215
 Переменная 13
 Поиск наибольшего 87
 Поиск наименьшего 87
 Порядковый тип 57, 78
 Последовательность 80
 Предложение использования 11, 225
 Программы линейной структуры 21
 Программы разветвляющейся структуры 44
 Программы циклической структуры 57
 Проект 12
 Пустой оператор 45

Р

Раздел операторов 11
 Расширенный синтаксис 185
 Рекуррентная формула члена ряда 100
 Рекурсивные подпрограммы 196

С

Синтаксические ошибки 15
 Смешанный способ вычисления члена ряда 101
 Сортировки 163
 Список фактических параметров 173, 174
 Список формальных параметров 173
 Стандартные типы 21
 Стандартные подпрограммы 12
 Стандартный модуль 12
 Статический массив 78
 Сумма ряда 99

Т

Тело цикла 57
 Тип данных 12
 Тип функции 185
 Трассировка 18

У

Упорядочения 163
 Условная компиляция 17
 Условные операторы 44

Ф

Фактические параметры 173, 174
 Формальные параметры 173

Ц

Цикл 57
 Цикл с заданным числом повторений 57
 Цикл с параметром 57
 Цикл с постусловием 57
 Цикл с предусловием 57

Ч

Часть инициализации 226

Часть реализации 225

Часть финализации 226

Член ряда 99

Ш

Шаблон консольной программы 11

Э

Элемент массива 78

Учебное издание

**Алексеев Юрий Евтихович
Ваулин Анатолий Сергеевич
Куров Андрей Владимирович**

ПРАКТИКУМ ПО ПРОГРАММИРОВАНИЮ

Редактор *В.М. Царев*
Техн. редактор *Э.А. Кулакова*
Корректор *И.М. Мартынова*
Художник *Н.Г. Столярова*
Компьютерная графика *Т.Н. Аверчивой*
Компьютерная верстка *Н.Ф. Бердавцевой*

Оригинал-макет подготовлен в издательстве МГТУ им. Н.Э. Баумана

Подписано в печать 21.04.08. Формат 60×90/16. Бумага офсетная.
Печать офсетная. Усл. печ. л. 18,0. Уч.-изд. л. 16,75.

Тираж 2000 экз. Заказ

Издательство МГТУ им. Н.Э. Баумана
105005, Москва, 2-я Бауманская, д. 5

ISBN 978-5-7038-3159-5



9 785703 831595