ФГБОУ ВО «Воронежский государственный технический университет»

Кафедра информатики и графики

48-2018

ПАСКАЛЬ: ТИПЫ ДАННЫХ, ОПЕРАТОРЫ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

для самостоятельной работы по дисциплине «Информатика» для студентов 2-го курса всех направлений подготовки

Воронеж 2018

УДК 004.9.(07)
ББК 32,81я7

Составители:
канд. физ.-мат. наук А.Д. Кононов,
канд. техн. наук А.А. Кононов

Паскаль: типы данных, операторы: методические указания для самостоятельной работы по дисциплине «Информатика» для студентов 2-го курса всех направлений подготовки / сост.: А.Д. Кононов, А.А. Кононов. – Воронеж: ВГТУ, 2018. – 44 с.

Методические указания являются продолжением учебного пособия «Основы программирования на языке Паскаль. Основные понятия алгоритмического языка Паскаль» и ставят целью углубление и практическое закрепление навыков составления программ на этом популярном языке. Методические указания снабжены большим количеством примеров и контрольных заданий для аудиторной и самостоятельной работы.

Методические указания предназначены для студентов 2-го курса всех направлений подготовки.

Ил. 15. Табл. 2. Библиогр.: 7 назв.

УДК 004.9.(07)
ББК 32,81я7

Рецензент канд. техн. наук, доц. Д.В.Сысоев

Печатается по решению учебно-методического совета ВГТУ

ВВЕДЕНИЕ

На начальном этапе обучения студенты познакомились с важнейшими базовыми разделами дисциплины «Информатика» - алгоритмизацией и основными понятиями программирования на примере конкретного алгоритмического языка Паскаль. В данной работе, кроме краткого экскурса в основы программирования, внимание акцентируется на освоении и использовании структурированных типов данных, операторов, рассмотрении некоторых типовых приемов программирования, в том числе и вычислении сумм бесконечных рядов с помощью итерационных циклов.

Издание содержит авторские иллюстрации.

1. СТАНДАРТНЫЕ ТИПЫ ДАННЫХ

1.1. Структура программы

Программа на языке Паскаль состоит из заголовка и собственно программы, называемой блоком. Блок состоит из разделов. Классическое расположение разделов следующее:

1) раздел меток;
2) раздел констант;
3) раздел типов;
4) раздел переменных;
5) раздел процедур и функций;
6) раздел операторов.

Раздел операторов заключается в операторные скобки BEGIN … END. Остальные перечисленные разделы носят декларативный (описательный) характер. Любой раздел, кроме последнего, может отсутствовать. Разделителями между разделами и операторами служит точка с запятой (;). В конце раздела операторов должна стоять точка (.). Для компилятора это признак физического завершения программы.

В любое место программы, где по смыслу мог бы присутствовать пробел, могут быть включены комментарии, не влияющие на фактическое исполнение программы. Комментарии заключаются в фигурные скобки { . . .} или пары символов (* . . . *).

1.2. Описание стандартных типов данных

Программа оперирует некоторыми объектами, называемыми данными. Каждый элемент данных в программе является либо кон-

стантой, либо переменной.

Значения констант не могут изменяться во время выполнения программы. Описываются в разделе описания констант после служебного слова CONST. Типы констант не указываются, а определяются в соответствии с используемыми значениями.

Для каждой переменной напротив в разделе описания переменных после служебного слова VAR задается некоторый тип, определяющий возможные значения переменной и операции, которые могут над ней выполнятся.

Целый тип обеспечивает использование целых чисел. В конкретной вычислительной машине существуют ограничения на диапазон целых чисел от - MAXINT до MAXINT. Над переменными целого типа определены следующие арифметические операции: + (сложение), - (вычитание), * (умножение), / (деление), DIV (деление целых чисел), MOD (вычисление остатка от целочисленного деления). Результат всех этих операций целого типа (кроме /, где результат вещественный). Определены также операции отношения = , <> , < , > , <= , >= . Их результат – логический (булевского типа). С аргументами целого типа могут использоваться следующие стандартные функции: SIN(X), COS(X), LN(X), ARCTAN(X), EXP(X), SQRT(X), дающие результат вещественного типа, а также SQR(X) и ABS(X), которые в данном случае возвращают результат целого типа (совпадает с типом аргумента).

Определены также функции:

ORD(X) – результат TRUE (истина) для нечетного аргумента и FALSE (ложь) для четного аргумента;

SUCC(X) – вырабатывает следующее целое число, то есть число, на единицу большее, чем X;

PRED(X) – вырабатывает предыдущее целое число, то есть число, на единицу меньшее, чем X.

Вещественный тип употребляется в том же смысле, что и вещественные числа в математике. Могут быть представлены в естественной (с фиксированной десятичной точкой) и нормальной (экспоненциальной, с плавающей десятичной точкой) формах.

Над переменными вещественного типа определены следующие арифметические операции: + , - , * , / , а также операции отношения.

С аргументами вещественного типа могут использоваться все вышеперечисленные стандартные функции; все они вырабатывают результат вещественного типа.

Определены также стандартные функции преобразования значения вещественного типа в значение целого типа:

TRUNC(X) – вырабатывает целый результат путем отбрасывания дробной части аргумента;

ROUND(X) – вырабатывает целый результат путем округления до ближайшего целого, например,

для x=21.53 trunc(x) = 21; round(x) = 22;

для x= -2.7 trunc(x) = -2; round(x) = -3.

Символьный тип. Значениями символьного типа является набор элементов – знаков. Символ, заключенный в апострофы, обозначает константу символьного типа, например, ‘x’, ‘5’, ‘ = ‘.

Существуют две стандартные, обратные по отношению друг к другу функции, называемые функциями преобразования:

ORD(S) – дает порядковый номер символа S в упорядоченном множестве символов (Американский стандартный код для обмена информацией ASCII), например, ord(‘:’) = 58; ord(‘5’) = 53.

CHR(I) – дает символ, стоящий под номером I в упорядоченном множестве символов, например, chr(66) = ‘B’; chr(57) = ‘9’. Очевидно, что chr(ord(s)) = s ord(chr(i)) = i.

Над переменными символьного типа определены операции отношения. Пусть c1, c2 - переменные символьного типа. Отношение c1<c2 истинно тогда и только тогда, когда ord(c1)< ord(c2).

К аргументам символьного типа применимы также стандартные функции PRED(S) и SUCC(S). Функции PRED(S) дает предыдущий символ, а функция SUCC(S) – следующий, например, PRED(‘B’) =’A’; SUCC(‘9’) = ‘ : ’. Справедливы также равенства:

PRED(S) = CHR(ORD(S) – 1) и SUCC(S) = CHR(ORD(S) + 1).

Логический тип. Переменные логического (булевского) типа могут принимать одно из двух возможных значений – TRUE (истина) или FALSE (ложь). В достаточном для практики рассмотрении над ними определены три логические операции: NOT (отрицание), AND (конъюнкция), OR (дизъюнкция). Булевский тип определен так, что FALSE < TRUE, то есть операции отношения применимы и к переменным логического типа.

Значения TRUE и FALSE можно теперь рассматривать как упорядоченное множество, состоящее из двух элементов. При этом определены следующие значения соответствующих стандартных функций:

ORD(FALSE) = 0; ORD(TRUE) = 1;

SUCC(FALSE) = TRUE; PRED(TRUE) =FALSE.

Контрольные вопросы и упражнения

1. Можно ли выполнять операцию вещественного деления (/) над переменными целого типа?

2. Какого типа будет результат деления 15 на 4?

3. Чему равно ODD(15) и ODD(24)?

4. Чему равно SUCC(137) и PRED(26)?

5. Какие из приведенных записей вещественных чисел являются неправильными и почему?

а) 7.	б) - 6.1	в) 0.0	г) 9	д) .0E-2	е) 0.1E-5
ж) -5.3E4	з) +2.3E+3	и) -71	к)0.31	л) A56	м) 2,1

6. Чему равно?

а) trunc (5.61)	б) trunc (-5.61)
в) round (17.16)	г) round (17.96)
д) round(-17.16)	е) round (-17.96)
ж) round (16.5)	з) round(-16,5)

7. Какие из приведенных записей являются неправильными и почему?

а) odd (17.1)	б) cos(32.1)
в) cos (5)	г) sin 0.2
д) succ (3.2)	е) pred (7)

8. Чему равно?

а) ord (chr (59))	б) chr (ord (‘ * ’))
в) pred (‘ B ‘)	г) succ (‘ B ‘)

9. Какие операции определены над переменными булевского типа?

1.3. Перечисляемый и ограниченный типы данных

Помимо приведенных выше стандартных типов данных Паскаль предоставляет программисту широкие возможности для создания дополнительных типов, характеристики которых он может определять по своему усмотрению. Новые типы описываются в декларативной части программы в специальном разделе типов или могут быть определены непосредственно при описании переменных.

Перечисляемый скалярный тип данных задается списком значений (объектов), которые могут принимать переменные этого типа.

При этом каждый объект есть имя. Числа, логические и символьные константы не могут являться объектами перечисляемого типа.

Применение перечисляемого типа повышает наглядность программы и дает возможность автоматически контролировать допустимость значений переменных. Например, описание

```
type  day = (mo, tu, we, th, fr, sa, su);
var d1,d2,d3 : day;
```

означает, что переменные d1, d2, d3 типа day при выполнении программы могут принимать только одно из семи указанных значений. Объект, указанный в списке, может присутствовать не более, чем в одном описании. Например, описание

```
type t1= (one, two, three, four);
type t2 = (four, five, six);
```

неправильно, эти два задания типов несовместимы, так как four встречается в обоих типах.

Имена объектов, указанных в описании перечисляемого типа, являются константами этого типа. Поэтому, имея описания

```
type name = (John, Tom, Nick, Ann, Jane );
type color = (red, blue, black);
var x, y, z : name; c1, c2, c3 :color;
```

можно записать следующие операторы присваивания:

```
c1:= blue;  x:= Ann;  y:= Tom;    c2:= red;
```

Нельзя присвоить переменной значение из описания другого типа, например, недопустимы присваивания

```
z:= black;          c3:= Tom;
```

Для перечисляемого типа данных существенен порядок указанных объектов. К данным перечисляемого типа применимы операции отношения. Например, из предыдущего описания и примеров операторов присваивания следует, что выражения

```
x>John ,    c2<c1,       y<x
```

имеют значение true.

Для объектов перечисляемых типов определены системные стандартные функции pred, succ и ord, имеющие тот же смысл, что и для стандартных скалярных типов. Например, для типа color результатом функции succ (red) является значение blue, результатом функции pred (black) – blue. Функция ord (blue) выработает значение, равное единице, так как нумерация объектов в списке начинается с нуля.

Ограниченный тип. Множество значений стандартных скалярных типов данных и перечисляемых скалярных типов упорядочено и

конечно (кроме вещественного типа). Если нам необходимо сузить диапазон значений, принимаемых некоторым объектом скалярного типа, то это можно осуществить в явном виде наложением ограничения на стандартный тип или определенный ранее перечисляемый скалярный тип, который в этом случае называется базовым. Например, в разделе описания типов

```
Type day = (mo, tu, we, th, fr, sa, su);
nom = 10 .. 25;
sss = ‘c’ .. ‘x’;
wd = sa .. su;
```

для ограниченного типа nom базовым является целый тип, для sss – символьный, для wd – определенный ранее перечисляемый тип day. Сначала указывается нижняя граница, потом верхняя (при этом нижняя граница не должна быть больше верхней). Попытка присвоить переменной ограниченного типа значение, не входящее в заданный диапазон, будет диагностирована как ошибка.

К переменным ограниченного типа применимы все операции и стандартные функции, которые допустимы при работе с переменными соответствующего базового скалярного (порядкового) типа.

Контрольные вопросы и упражнения

1. Описать переменную i как переменную ограниченного типа, принимающую целочисленные значения от 20 до 35.

2. Можно ли задать переменную ограниченного типа, принимающую вещественные значения на отрезке [2; 5]?

3. Описать переменную перечисляемого типа, принимающую значения названий дней недели.

4. Указать ошибки в следующих описаниях:

```
const a = 2 .. 30;  b:= ‘b’,  pi = 3,14159;
type length = (1 .. 20);
town = (Moscow; Tomsc; Omsc; Voronezh);
letter =(‘a’, ‘b’, ‘c’, ‘d’);
nom =(0,1,2,3,4,5,6,7,8,9).
```

5. Можно ли в разделе типов описать одновременно два следующих перечисляемых типа

```
type pencil = (blue, red, green);
```

color = (jellow, blue, black)?

6. Имеются следующие описания:

```
type figure = (circle, square, triangle);
var x,t : figure;  z,y : (basket, bag, portfolio).
```

Какие из перечисленных операторов присваивания неправильно записаны и почему?

а) x:= circle; б) z:= square;
в) t:= x; г) y:= ord(z) + 1.

7. Правильно ли описаны следующие переменные:

```
var text : (summer, autumn, winter, spring);
    text1: summer .. winter;
    text2 : spring .. autumn;
    text3 : winter .. winter.
```

8. Найти ошибки в следующей программе:

```
program error;
type month = (jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec);
autumn = sep .. nov;
var m : autumn;  d: '0' .. '9';  k: 0 .. 9;
begin
read(m, d, k);
if m>sep then d:=k else k:=ord(m) – 8;
writeln( k, d + k)
end.
```

Объяснить, какие правила языка Паскаль нарушены.

9. Описать переменную birthyear (год рождения), характеризующую допустимые года рождения учащихся группы. Какой тип данных целесообразно использовать?

10. Каким базовым скалярным типам соответствуют типы, описанные ниже? Какие операции и стандартные функции определены для переменных этих типов?

```
type computer = (ENIAC, Mir, Iskra, Elektronika, CM4, IBM);
var x,y : ENIAC .. Iskra;    i : 3 .. 9.
```

2. ПРОСТЫЕ ОПЕРАТОРЫ

2.1. Выражения и оператор присваивания

Значение *выражения* вычисляется с учетом расставленных скобок и старшинства операций (уровней приоритета):

1) вычисление значений функций;
2) NOT;
3) * , / , MOD, DIV, AND;
4) + , - , OR;
5) < ,<= , > , >= , <> , = .

Операции в одной строке (под одним номером) имеют одинаковый приоритет и выполняются слева направо в порядке их появления в выражении. Выражения в круглых скобках вычисляются в первую очередь.

При выполнении арифметических операций над величинами только вещественного, а также вещественного и целого типов вырабатывается результат вещественного типа. Тип выражения определяется последовательным вычислением типов результатов всех операций, выполненных при определении значения выражения. Например, для нахождения типа выражения

(ter – ln(x) > 2.5 * kl) and (ord(s) < ni),

у которого типы входящих переменных определены как

ni, kl : integer; ter,x :real; s :char;

следует последовательно определить:

1) a1 = ln(x) тип real;
2) a2 = 2,5 * kl тип real;
3) a3 = ter – a1 тип real;
4) a4 = a3 > a2 тип boolean;
5) a5 = ord(s) тип integer;
6) a6 = a5 < ni тип boolean;
7) a4 and a6 тип boolean.

Присваивание - это операция придания переменной ее значения.

Общий вид оператора присваивания

V := A;

где V – имя переменной, A – выражение, := – операция присваивания (неделимый символ).

Переменная и выражение должны быть одного типа. Допускает-

ся исключение: переменная вещественного типа, а выражение – целого. Обратное запрещено.

При прочтении данного оператора вычисляется выражение, стоящее в правой части, и его значение присваивается переменной в левой части, то есть найденное значение выражения A заносится в ячейку памяти с именем V.

Для преобразования значений вещественного типа в значения целого типа предназначены функции TRUNC(X) и ROUND(X).

Контрольные вопросы и упражнения

1) Какие из приведенных записей являются правильными?

а) ‘a’ < ‘c’ б) ‘1’ > ‘9’ в) false < true

г) ‘a’ < ‘b’ and 1 < 2 д) 0’ or ‘9’ e) (‘d’ > ‘g’) and (1 < 2)

2. Записать на языке Паскаль

а) $\frac{x^2 + 3x - y}{\arcsin x + e^x}$

б) $\frac{x^2 c^4}{|x| + d^2} + d$

в) $\sqrt[3]{\cos x^2} + 1 + \operatorname{arctg}\frac{x}{2}$

г) $\frac{y}{2x}|4xy - 2bz^3| + \cos\frac{x}{2^x}$

д) $\frac{2bc^3 + \sqrt[3]{(abc)} - y}{2x + |e^{2xy} - \arcsin x}$

е) $\frac{x^2 \sin^9(x+4) + x - \frac{\pi}{2}}{\operatorname{arctg}(2a) + e^x}$

ж) $\frac{2\ln^2(1+x^3) + e^{2ax} - y}{\operatorname{arctg}(x-3) + e^{2x} + \cos(k-2)}$

з) $\frac{x\sqrt[3]{x} + ax + bx^2 - y^3}{\sin 2x + e^x + |bx - 4d|}$

и) $\frac{\operatorname{arctg}^2 x + \sqrt[5]{3x - y}}{3 - x^7 \sin x + e^{2x}}$

к) $\frac{x\cos^2 x + \sin 3x^3 - 4\sqrt{y}}{3 - x^7 + c\sin\frac{x}{2} + e^{-2x}}$

л) $\frac{\cos x \cdot e^{-p^2} + \sqrt[3]{2xz} - y}{|6z^3 - y^2| \cdot c\sin x + \operatorname{tg}^2(e^x)}$

м) $\frac{2x^2 + \cos^2(x + 0.003)}{\sin\frac{2x}{3} + e^{-3x} + y^5}$

н) $\frac{\cos^2(x+1) + ax + c}{\operatorname{tg}^3 x + e^{-2x} - 5}$

о) $\frac{|a^3 + 3x^3| + cx - y}{cx^3 + \sqrt[3]{x+1} + \sin^3 e^x}$

п) $bx^7 + \frac{3x^2 \cos^2(x+1)}{\sqrt{ax^3} \cdot \sin x + e^x}$

р) $\frac{2\ln^2(1+x^{2)} + |3x - y|}{\operatorname{arctg}(x^4) \cdot \sin x + 5}$

3. Определить тип результата выражений, если i, j, k : integer; x, y : real;

а) i * i +j * j +2 *k/2 б) x < y

в) sin(x) + 2 * cos(y) г) I + sqrt(j)

4. Какие из приведенных ниже операторов присваивания являются правильными (если i, j, k :integer; x, y, z :real; a, b :boolean):

а) x:= y+ sin (sin(z)); б) a :=(x<y) or b and (i<>k);

в) x := i + j – b г) i := i + k/j

5. Вычислить значения выражений, если a = 2.5; b = 7.8; c = -17.3; m =5; x = 8.7; e = true

а) (a + b)/c * m б) 2 + x * x/(x + (a + b)/5)

в) (a < b) and (c > a) or (c < m) or e

6. Записать приведенные ниже высказывания в виде выражений:

а) значение a не принадлежит интервалу (0, 3);

б) значение a принадлежит отрезку [-2, 0];

в) значение a принадлежит одному из отрезков [-5, -4], [0, 2], [3.2, 7].

г) точка a с координатами (x, y) лежит внутри круга с центром в начале координат и радиусом 2.3;

д) точка a с координатами (x, y) лежит на границе или вне единичного круга с центром в начале координат.

2.2. Организация ввода – вывода

Ввод – вывод связан с обменом информацией между оперативной памятью ЭВМ и внешними носителями информации.

Задание исходных данных (*ввод*) можно выполнить по-разному.

1. В разделе констант задать соответствующие значения. При этом тип константы автоматически определяется по содержимому правой части.

Пример. Вычислить значение $y = ax^2 + kx – t$ при a = 2,5; x = 7,3; k = -17,5; t = 548.

```
const a = 2.5; x = 7.3; k = -17.5;  t = 548;
. . . . . .
begin
                y:=a*x*x + k*x – t;
. . . . . .
   end.
```

Именованным константам a, x, к, t присваивается вещественный тип, а константе t – целый.

Однако такой способ задания данных не всегда удобен, поскольку позволяет вычислять y только для одного набора параметров, так как изменять значения констант в программе нельзя.

2. В разделе переменных описываются переменные a, x, к, t, а в разделе операторов им присваиваются соответствующие значения:

```
var a, x, k, t, y :real;
. . . . . .
begin
    a:=2.5;  x:= 7.3;  k:= -17.5;  t:= 548.0;
    y:=a*x*x + k*x – t;
. . . . . .
end.
```

В этом случае возможности варьирования значениями параметров расширяются, так как в программе можно организовать их изменение, однако, набор параметров будет статическим.

3. Для обмена информацией с окружающим миром в программах, написанных на языке Турбо Паскаль (ТП), можно использовать специальные стандартные (встроенные) процедуры. Процедура – это некоторая последовательность операторов языка ТП, к которой можно обратиться по имени. Так используемые для ввода данных операторы

```
READ (a1, a2, . . . ,an);        (1)
READLN (a1, a2, . . . ,an);      (2)
READLN;                          (3)
```

являются по своей сути операторами обращения к встроенным процедурам ввода данных. Здесь a1, a2, . . . ,an – список имен переменных, значения которых необходимо ввести, то есть эти операторы позволяют выполнить программы с различными наборами исходных данных. Типы переменных должны соответствовать типам вводимых значений. Ввод в языке Паскаль может быть только бесформатным. Могут вводиться данные целого, вещественного и символьного типа. Числа при вводе разделяются символом «пробел» или клавишей «ввод» (Enter). Символы при вводе не разделяются. Ввод строки завершается символом «ввод».

В отличие от READ оператор READLN осуществляет переход к следующей строке после ввода всех указанных в операторе данных.

Возможны различные методы организации ввода этих чисел:

а) с использованием одного оператора READ:

```
var a, b, d, t : real;
. . . . . .
begin
. . . . . .
read (a,b,d,t);
. . . . . .
end.
```

Информация набирается на одной или нескольких строках экрана, так как если чисел в строке больше нет, а список ввода не исчерпан, то автоматически осуществляется переход к новой строке. В дальнейшем рассматриваем ввод – вывод только с терминала в диалоговом режиме.

б) с использованием нескольких операторов READ:

```
. . . read (a, b, d);          read (t); . . .
```

Набор чисел как в предыдущем случае.

в) с использованием комбинации операторов READ и READLN:

```
… readln ( a, b, d ) ;      read ( t ) ; …
```

Здесь числа, соответствующие переменным a, b, d, набираются на одной строке, а число, соответствующее переменной t, – на другой, так как после окончания работы оператора readln (a, b, d) оператор read (t) может считать данные только с начала следующей строки.

Вывод данных. Для вывода данных используются процедуры

```
WRITE (a1, a2, . . . , an);          (4)
WRITELN (a1, a2, . . . , an);        (5)
WRITELN ;                            (6)
```

где a1, a2, . . . , an – список параметров или выражений, значения которых необходимо вывести; выводимые выражения могут быть целого, вещественного, символьного, логического или строкового типа.

Оператор (5) в отличие от (4) после вывода последнего значения в списке осуществляет переход к следующей строке.

Операторы вывода допускают использование в явном виде указания о ширине поля, отводимого под выводимое значение.

Так, общий вид процедуры при выводе значений целого, символьного, строкового или логического типа

```
WRITE (a : m);  или    WRITELN (a : m);
```

где a – выводимое выражение или имя переменной; m – константа или выражение целого типа, значение которого указывает число позиций, отводимых для записи значения выражения a.

При выводе значений вещественного типа с фиксированной десятичной точкой указывается ширина поля, отводимая для записи всего числа m и для записи дробной части числа n. Общий вид:

WRITE (b : m : n); WRITELN (b : m : n);

где b – выводимое выражение вещественного типа, m и n – константы или выражения целого типа.

Приведем примеры операторов вывода.

1. Организация вывода значений переменных a, b, c целого типа:

а) . . . write (a, b); writeln (c); . . .

Все числа печатаются на одной строке. Аналогичный результат можно получить, написав один оператор write (a, b, c);

б) . . . writeln (a, b); writeln (c); . . .

Значения a и b печатаются на одной строке; c – со следующей строки;

в) . . . write (a : 12, b : 5, c : 4); . . .

При печати под значение переменной a отводится 12 позиций, под b – 5, под c – 4 позиции.

Если число литер в представлении выводимого значения оказывается меньше, чем m, то оно слева дополняется пробелами. Если наоборот количество указанных позиций недостаточно, то происходит автоматическое увеличение поля до необходимых размеров;

г) … write (‘a =’, a: 2,’ b = ‘, b: 3, ‘ c=’, c: 1);

Здесь используется возможность вывода строк символов. Будет напечатано

a = 12 b= – 25 c= 4.

2. Организация вывода вещественных чисел предполагает задание общего количества позиций и количества позиций после запятой. Если при выводе вещественных значений не указывается n (или m и n), то результат получается в нормализованном виде с десятичным порядком, то есть в экспоненциальной форме.

3. При выводе значений символьного типа, если в явном виде не указывается количество позиций, под каждый символ отводится одна позиция. Например,

writeln (‘ s1=’, s1:5,’ s2 = ‘, s2);

напечатает (если символьной переменной s1 в программе присвоено значение ‘*’, а s2 – ‘w’) следующую строку:

s1 = * s2 = w.

4. При выводе значений булевского типа на печать выводится true или false, например, оператор writeln (a < b : 7) ; напечатает в от-

веденных семи позициях слово true, если значение переменной a меньше b, и слово false в противном случае.

Контрольные вопросы и упражнения

1) Можно ли вводить с помощью оператора read значения булевского типа?

2) Что получится в результате выполнения операторов?

а) write (‘a’:3, 2:1); writeln (5*3.2);
б) writeln (5<6.5 < =8:8);   writeln (‘ конец ‘);
в) writeln (5*3.2:1,’ результат ‘);

3) Написать программу для вычисления выражения

$$\frac{5.23 + 7.6^2 + \sin\frac{\pi}{7}}{\sin\frac{2\pi}{7} + 3.1}$$

без использования оператора присваивания.

2.3. Оператор безусловного перехода

Рассмотренные операторы присваивания и ввода – вывода относятся к простым операторам, так как не содержат в себе других операторов и служебных (зарезервированных) слов. Наряду с ними к группе простых операторов относится оператор безусловного перехода. Он применяется в случае, когда необходимо выполнить не следующий по порядку оператор, а какой-либо другой, отмеченный меткой (метка может содержать как цифровые, так и буквенные символы). Все метки, встречающиеся в программе, должны быть описаны в разделе меток после ключевого слова LABEL.

Формат оператора: GOTO < метка> ;

При прочтении данного оператора ЭВМ находит в программе оператор с указанной меткой и продолжает выполнение программы, начиная с этого оператора.

При использовании оператора GOTO необходимо выполнение правила: метка, на которую передается управление, должна находиться в том же блоке или модуле, что и сам оператор перехода; то есть областью действия метки является тот блок, в котором она описана, и переход возможен только в пределах блока.

Язык Паскаль является структурированным, поэтому особой

необходимости в операторе безусловного перехода нет.

Контрольные вопросы и упражнения

1) Чему будет равно значение переменной x после выполнения следующего фрагмента программы:

```
x:= 3;
goto 5;
3: x:=x +1.5;
goto 2:
5:x:=x+2.0;
goto 3;
2: . . . . . .
```

2) Что произойдет с программой

```
program primer;
label m1, m2;
. . . . . .
begin
m1: goto m2;
m2: goto m1
end.
```

2.4. РАБОТА С ЭВМ

2.4.1. Организация ввода – вывода

Цель задания:

1. Освоение простейшей структуры программы.
2. Получение навыков в организации ввода – вывода значений стандартных типов данных.
3. Получение практических навыков работы в диалоговом режиме.

Методические рекомендации

1. При вводе с терминала числа и символы можно набрать как на одной строке, так и на разных строках. При этом надо помнить, что ввод со следующей строки осуществляется в том случае, если предыдущим оператором ввода является readln.

2. Переменной логического типа можно присвоить значение либо в разделе описания констант, либо в операторе присваивания. Вве-

сти значения переменных логического типа нельзя.

3. При работе в интерактивном (диалоговом) режиме следует перед операторами ввода использовать оператор вывода на экран приглашения – подсказки о том, что наступило время ввода информации и какой именно.

Пример. Составить программу для ввода и вывода данных различных типов.

Исходные данные: k = 37, l =432, m = 76, x=3.24, y = 56.678, s1= ‘*’, s2=’g’, s3=’#’, fam = ’ Петров И.Н.’

```
program  in _ out;
const bool = true;
var  k, l, m : integer; x,y: real; s1, s2, s3: char; fam: string [20];
begin
      write (‘введите фамилию:’);
   readln (fam);
   writeln;
   write ( ‘введите целые числа k,l,m:’);
   readln ( k, l, m);
   writeln;
writeln (‘ ‘:30, ‘целые числа’);
writeln;
writeln(‘ ‘:5, ’стандартный формат ‘, ‘ ‘:18,’ заданная ширина поля’);
writeln;
writeln (‘ ‘:11,k, ‘ ‘,l,’ ‘,m,’ ‘:26,’k=’,k:3,’l=’,l:2,’m=’,m:5);
writeln;
write (‘введите вещественные числа x, y:’);
readln (x, y);
writeln;
writeln (‘ ‘:25,’вещественные числа’);
writeln;
writeln (‘ ‘:5,’стандартный формат’,’ ‘:18,’заданная ширина поля’);
writeln;
writeln (‘ ‘,x, y, ‘ ‘:5,’x=’,x:8:2,‘ y=’,y:6:3);
writeln;
write (‘введите символы s1, s2, s3:’);
readln (s1, s2, s3);
writeln;
writeln (‘ ‘:20, ‘символьные переменные’);
writeln;
```

```
writeln(‘ ‘:5, ‘стандартный формат’,’ ‘:10, ‘с шириной поля 5’);
writeln;
writeln(‘ ‘:5, ‘ s1=’, s1,’ s2=’,s2, ‘ s3=’, s3,’ ‘:12, s1:5, s2:5, s3:5);
writeln;
writeln (‘логическая переменная=’, bool);
writeln;
writeln(‘ ‘:35, ‘программу выполнил :’,fam :12)
end.
```

Протокол работы программы:

введите фамилию: Петров И.Н.

введите целые числа k, l, m: -37 432 76

целые числа

стандартный формат заданная ширина поля

-37 432 76 к =-37 l =432 m = 76

введите вещественные числа x, y : 3.24 56.678

вещественные числа

стандартный формат заданная ширина поля

3.2400000000E+00 5.6678000000E+01 x= 3.24 y=56.678

Введите символы : s1, s2, s3 : *G #

Символьные переменные

стандартный формат с шириной поля 5

s1 = * s2 = G s3 = # * G #

логическая переменная = true

программу выполнил Петров И.Н.

2.4.2. Вычисление выражений. Использование стандартных функций

Цель задания:

1. Изучение порядка действий при вычислении выражений.

2. Приобретение навыков в записи выражений на языке Паскаль и использовании стандартных функций.

Постановка задачи:

а) Найти значение функции y(x) при заданном x. Используя стандартные функции, вычислить y11 = [y], где [ ] означает целую часть числа;

$$y22 = [\, y \pm 0.5], \text{ (« + » для } y > 0 \text{ и « - » для } y < 0).$$

б) Записать выражение, зависящее от заданных в условии координат точки x1 и y1 и принимающее значение true, если точка принадлежит заштрихованной области, и false, если не принадлежит. Для заданной точки вычислить и вывести значение этого выражения.

Методические рекомендации

1. Функции, отсутствующие в списке стандартных функций языка Паскаль, выразить через имеющиеся.

2. Вывод значения выражения в данной точке организовать, используя запись выражения в операторе writeln.

Пример. Составить на языке Паскаль программу, которая вычисляет $y(x) = 3^{-x+1}\sin(x)$ при заданном x, выводит y11, y22 и проверяет принадлежность точки с координатами (x1, y1) заштрихованной области.

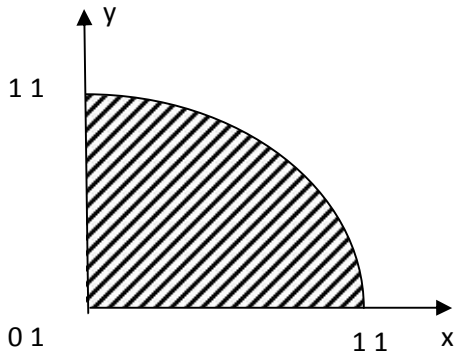

Рис. 1

Исходные данные: x = -1.5, x1 = 0.5, y1 = 1.2

```
program prin;
var x, x1, y1, y : real;
begin
        writeln (‘ введите x: ‘);
readln (x);
        y:= exp((-x + 1) * ln(3)) * sin (x);
writeln (‘ при x=’,x:8:3, ‘ y=’, y:8:4);
        writeln (‘ y11=’, trunc (y) : 4, ‘ y22=’, round (y) : 4);
        writeln (‘ введите координаты: x1, y1’);
        readln (x1, y1);
        writeln ((x1 >=0) and (y1 >=0) and (sqr(x) + sqr(y) <=1));
        writeln (‘ ‘:10, ‘ программу составил Петров И.Н.’)
end.
```

Протокол работы программы:

```
введите x:
 - 1.5
при x = -1.500 y = -15.5494
 y11 = -15  y22 = -16
введите координаты: x1, y1
 0.5 1.2
  false
        программу составил Петров И.Н.
```

Варианты заданий:

1. а) y = $2^{-x} \cdot \sqrt{x + \sqrt[4]{|x|}}$ при x = 4.145

б) координаты исследуемой точки: (0.6; 0.8).
Область изображена на рис. 2.

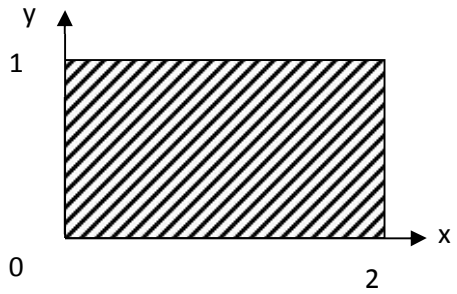

Рис. 2

2. а) y = $\sqrt[3]{e^x} - \sin x$ при x = 3.247

б) координаты исследуемой точки: (1.6; 0.6)
Область изображена на рис. 3.

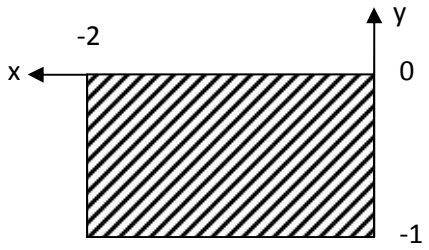

Рис. 3

3. а) $y = \sqrt[3]{|x^2 - 1| + \sin x}$ при x = 8.43109
б) координаты исследуемой точки: (0.4; 0.8)
Область изображена на рис. 4.

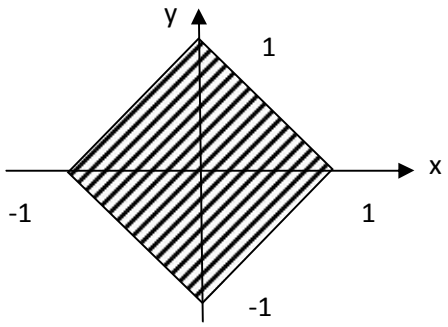

Рис. 4

4. а) $y = x\cos x + \sin^3 x$ при x = 16.241
б) координаты исследуемой точки: (0.75; -0.5)
Область изображена на рис. 5.

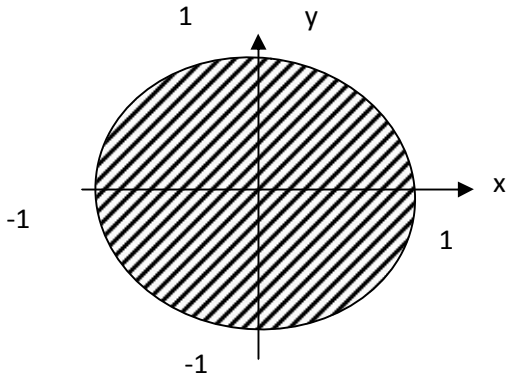

Рис. 5

5. а) $y = tg^2 x + |x|$ при x = -6.2312
б) координаты исследуемой точки: (0.3, 0.48)
Область изображена на рис. 6.

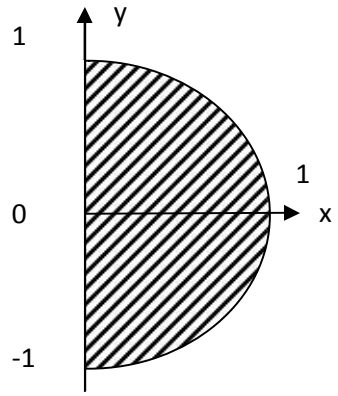

Рис. 6

6. а) $y = \sin x + \frac{1}{x} + \frac{1}{x^2}$ при x = -0.781

б) координаты исследуемой точки: (-0.4, -2.5).
Область изображена на рис. 7.

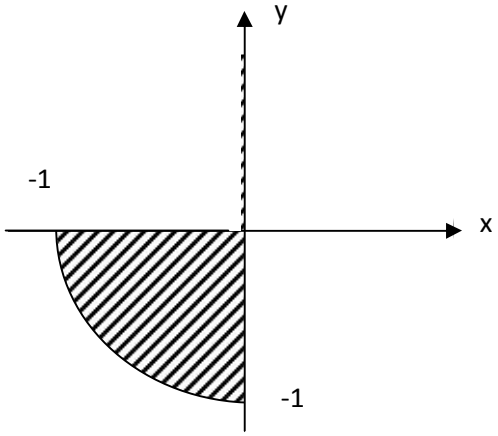

Рис. 7

7. а) $y = ch|x+1| + \sin^2 x$ при x = 5.324
б) координаты исследуемой точки: (0.6, -0.8).
Область изображена на рис. 8.

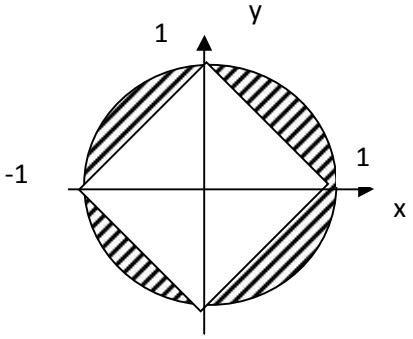

Рис. 8

8. а) $y = \arcsin x + \sqrt[3]{1-x}$ при x = 0.112
б) координаты исследуемой точки: (1.0, 0.713).
Область изображена на рис. 9.

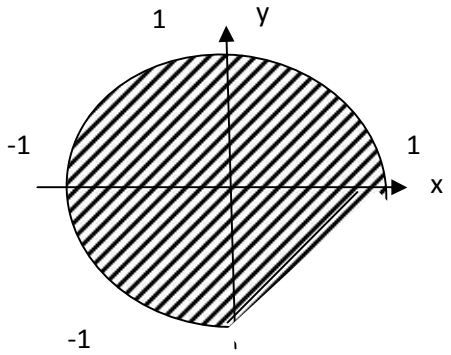

Рис. 9

9. а) y = sin (arctgx)+ $\sqrt{1+x^2}$ при x = -0.7129;
б) координаты исследуемой точки: (-0.5, 0.9)
Область изображена на рис. 10.

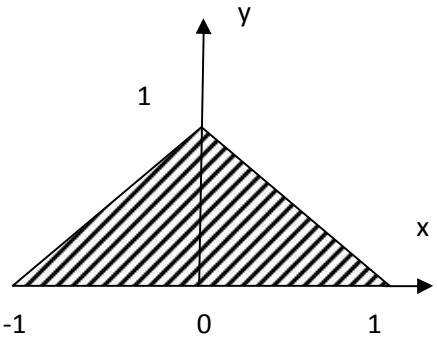

Рис. 10

10. а) y = $5\text{arctg}(x^2+1)$ при x = -2.1941
б) координаты исследуемой точки: (1.5, 0.021).
Область изображена на рис. 11.

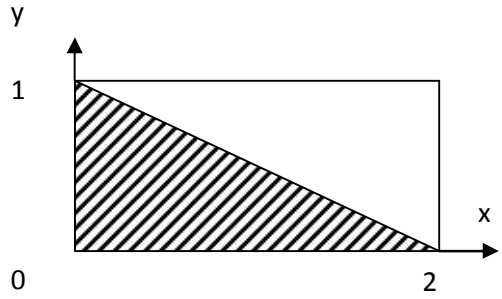

Рис. 11

11. а) y = $\sqrt[3]{\sin x}+\cos^2 x$ при x = -4.1231
б) координаты исследуемой точки: (-0.4, -0.91).
Область изображена на рис. 12.

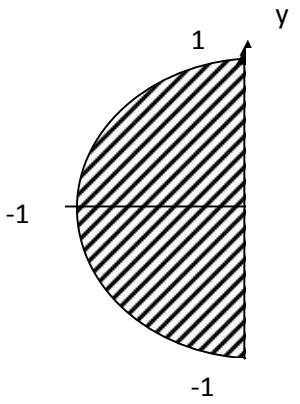

Рис. 12

12. а) $y = 4\text{arctgx} + \dfrac{x+7}{\sqrt{1+x^2}}$ при $x = -0.4178$

б) координаты исследуемой точки: (-0.9, -0.6).
Область изображена на рис. 13.

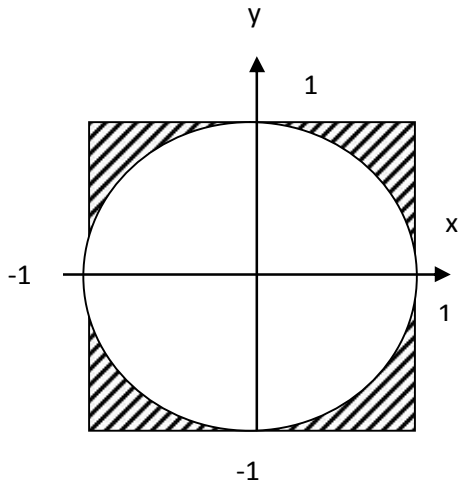

Рис. 13

13. а) $y = x^3 + 2x - \sqrt{|1 - \text{tgx}|}$ при $x = -0.61$

б) координаты исследуемой точки: (-0.4, -0.51).
Область изображена на рис. 14.

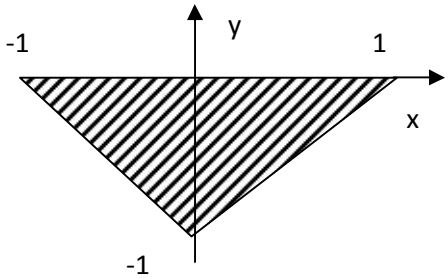

Рис. 14

14. а) $y = e^{\sin x} + \sqrt{|\cos 6x|}$ при $x = 0.212$

б) координаты исследуемой точки: (0.65, -0.7812).

Область изображена на рис. 15.

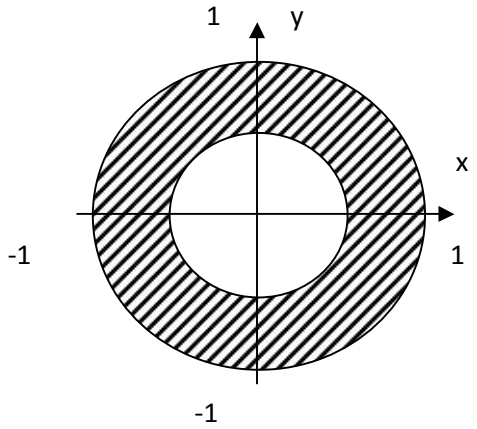

Рис. 15

3. СТРУКТУРНЫЕ ОПЕРАТОРЫ

К структурным (сложным) операторам относятся составной оператор, условные операторы, выбора, цикла, так как представляют собой конструкции, включающие в себя другие операторы.

3.1. Составной оператор

Представляет собой группу из произвольного числа операторов, отделённых друг от друга точкой с запятой, заключенных в так называемые операторные скобки BEGIN и END.

Формат оператора:

```
  begin
<оператор 1>;
 . . . . . .
< оператор n>
end;
```

Составной оператор воспринимается и обрабатывается как единое целое, может находиться в том месте программы, где синтаксис языка Паскаль требует присутствие только одного оператора.

3.2. Условные операторы

Используются для организации разветвляющихся вычислений и обеспечивают выполнение или невыполнение некоторого оператора (который может быть и составным) в зависимости от заданных условий.

Оператор IF может принимать одну из следующих форм:

1)IF <условие > THEN <оператор 1>

ELSE <оператор 2>; (альтернатива) ;

2) IF <условие > THEN <оператор>; (безальтернативная форма).

Условие – это выражение логического типа, оно может быть простым или сложным; оператор 1, оператор 2 – любые операторы языка Паскаль, в том числе составные. При записи простых условий могут использоваться все возможные операции отношения. Сложные условия образуются с помощью логических операций AND, OR, NOT.

Для условного оператора первого вида, если условие принимает значение TRUE, выполняется оператор 1, а оператор 2 игнорируется. Если же условие имеет значение FALSE, то оператор 1 игнорируется и выполняется оператор 2.

Для оператора второго вида при истинности условия выполняется оператор, стоящий после THEN, а в противном случае выполняется следующий оператор программы.

Поскольку любой из операторов 1 или 2 также может быть условным, а в то же время не каждый из условных операторов может иметь ветвь ELSE, то возникает неоднозначность трактовки условий. Поэтому по соглашению, принятому в ТП, часть ELSE всегда соответствует ближайшему к ней сверху оператору IF , не имеющему ветви ELSE.

Конструкций со степенью вложенности более трёх из-за сложности их анализа и отладки следует избегать.

Контрольные вопросы и упражнения

1. Какие из приведенных ниже операторов являются неправильными и почему?

а) if a<b then a:=a * a else b:=b * b

б) if x and y then s:=s + 1; else s:=s – 1;

в) if k <> m then k:=m;

г) if 5 then s:=s + 5;

д) 12: if (a < b) or c then c:= false;

е) if (a – b) and p then p:=p + 10.5;

2. Если a = 0.5, b = -1.7, то какие значения будут иметь переменные a и b в результате выполнения условного оператора:

If a < b then a:=b else b:=a;

3. Какая задача решается при выполнении оператора:

if x < y then max:= y else max:=x;

4. Задать с помощью условного оператора следующие действия:

а) меньшее из двух значений переменных вещественного типа x и y заменить нулем, а в случае их равенства – заменить нулями оба значения;

б) наибольшее из трех различных значений переменных целого типа x, y, z уменьшить на 2.

5. Поменять местами значения целых переменных a, b, c, чтобы оказалось $a \geq b \geq c$.

6. Булевской переменной b присвоить значение true, если значения переменных x и y равны, и значение false в противном случае.

7. Составить блок–схему и написать программу для вычисления функции:

а) $y = \begin{cases} \sqrt[3]{x + \ln|x+1|} & x \leq 2 \\ 3x^2 + tg^2 x & 2 < x \leq 4 \\ a\sin(x+1) + c & x > 4 \end{cases}$

б) $y = \begin{cases} 1 + x^2 arctgx, & x < -3 \\ 1 + xtg^2(x+1) & -3 \leq x < 2 \\ 2\cos(\ln(x+2)) & 2 \leq x < 4 \end{cases}$

в) $y = \begin{cases} 2\cos^2\frac{x}{4} & x < \frac{\pi}{4} \\ a(1 + tg^2\frac{x}{2}) & \frac{\pi}{4} \leq x \leq \frac{\pi}{2} \\ \sin^2\frac{x}{2}(1 + cx) & x > \pi \end{cases}$

г) $y = \begin{cases} \ln\left|\frac{x^{-1}}{ax+b}\right| - e^{2x^2} & 4 < x \leq 8 \\ 4arctgx + \frac{ax+7}{\sqrt{|b-x|}} & x > 8 \\ x^3 + 2ax - \sqrt[3]{tg^2 x} & x < 0 \end{cases}$

д) $y = \begin{cases} 3x^2 + \ln x^2 & x < 0 \\ tg(ax-b) - b|x| & 0 \leq x < 3 \\ cx^3 + arctg(2x + 0.5) & x > 3 \end{cases}$

е) $y = \begin{cases} \dfrac{1}{2x\sqrt{4x+1}} & x > 0 \\ a^2bx + cx^4 & -3 \le x \le 0 \\ cd + \sqrt[3]{x^2} + \sin(\dfrac{x}{10}) & x < -3 \end{cases}$

ж) $y = \begin{cases} 3.2 + tg^2(\dfrac{x}{5}) & -2 < x \le 4 \\ 2x^4 + \sqrt[3]{x+1} & x > 4 \\ \ln(1+x^2) + a^2\cos 2x & x = -2 \end{cases}$

з) $y = \begin{cases} 2x^3 + \cos^3 x & o < x \le 1 \\ \ln(1+x^2) + \sqrt{x-1} & 1 < x \le 3 \\ 3a^2 + tg(ax - b) & x > 3 \end{cases}$

и) $y = \begin{cases} \cos x \cdot e^{-p} & o < x < 3 \\ \dfrac{\sin 2x}{2x} & 3 \le x \le 5 \\ \ln^3(x+1) + p & x > 5 \end{cases}$

к) $y = \begin{cases} e^{-x^2} + \sqrt[3]{x} + 1 & x \le 3 \\ \cos^3(\dfrac{x}{10}) & 3 < x < 9 \\ \ln^3(x+1) & x = 9 \end{cases}$

л) $y = \begin{cases} \ln(1 - 0.2x^2) & -2 < x < -1 \\ \dfrac{arctgx}{\sqrt{1+x^2}} & -1 \le x \le 1.5 \\ arctg\ \dfrac{1}{\sqrt{x}} & x > 2.5 \end{cases}$

м) $y = \begin{cases} k + tg^2(\dfrac{x}{2}) & x \le 3 \\ 1 + \ln(|\cos(\dfrac{x}{2})|) & 3 < x \le 4 \\ c - \sin^2(x^2+1) & x > 5 \end{cases}$

н) $y = \begin{cases} \ln(\sqrt[3]{x} + 1) & 1 < x < 2 \\ \sqrt[3]{x + \sin^2 x} & -2 < x \le 1 \\ 2x^4 + \sqrt{x^2 + 1} & x \le -2 \end{cases}$

о) $y = \begin{cases} 2bc + \sqrt[3]{x^4} & 3 < x < 5 \\ ax^3 + \ln(1 + x^3) & 1 \le x \le 3 \\ b\cos(x + \frac{1}{x}) & x < 1 \end{cases}$

п) $y = \begin{cases} a\ln|\sqrt[3]{1 - x^2}| & 0 < x < 1 \\ d\sin 2x + c\ln x & 1 \le x \le 4 \\ \sqrt{x^3 + 2} + \frac{1}{x - a} & 4 < x < 5 \end{cases}$

3.3. Оператор выбора

Оператор выбора CASE является обобщением оператора IF для произвольного числа разветвлений. Обеспечивает организацию вычислений путём выбора одного из нескольких операторов. Он состоит из выражения, называемого селектором, и списка операторов, каждому из которых предшествует список констант выбора (список может состоять и из одной константы). Как и в операторе IF здесь может присутствовать слово ELSE, имеющее тот же смысл.

Формат: CASE < выражение – селектор > OF
< список 1> : < оператор 1 > ;
< список 2> : < оператор 2 > ;
……..……
< список N> : <оператор N >
ELSE < оператор >
END;

При прочтении данного оператора ЭВМ вычисляет значение выражения – селектора, затем переходит к выполнению того оператора, которому предшествует константа выбора, равная значению селектора. Если ни одна из констант не равна текущему значению селектора, выполняется оператор, стоящий за ELSE. Если вновь ELSE отсутствует, то управление передаётся оператору, находящемуся за словом END, то есть первому оператору за границей блока CASE.

При использовании оператора выбора CASE должны выпол-

няться следующие правила:

а) селектор должен относиться к одному из целочисленных (находящихся в диапазоне – 32768 … 32767), логическому, символьному или пользовательскому типу;

б) список констант выбора состоит из произвольного количества значений или диапазонов, отделённых друг от друга запятыми;

в) все константы выбора должны иметь тип, совместимый с типом селектора;

г) все константы должны быть уникальны, диапазоны не должны пересекаться.

Примеры

1. Программа, которая в зависимости от номера месяца выдаёт сообщение о времени года

```
program seasons;
var k: integer;
begin
  writeln (‘ введите номер месяца ‘);
  read ( k );
   case k of
    1, 2, 12 : writeln (‘ зима ’);
    3, 4, 5 :  writeln (‘ весна ’);
    6, 7, 8 :  writeln (‘ лето ‘);
    9, 10, 11: writeln (‘ осень ’ )
   end
end.
```

2. Селектор интервального типа

```
case f of
  1 .. 10 :  writeln (‘ число’, f:4, ‘в интервале [1 .. 10]’);
 11 .. 20 :  writeln (‘ число’, f:4, ‘в интервале [11..20]’);
 21 .. 30 :  writeln (‘ число’, f:4, ‘в интервале [21..30]’)
 else writeln (‘ число’, f:4, ‘ вне интервала [1 .. 30]’)
end;
```

3. Селектор символьного типа

```
 var ch : char;
 . . . . . .
 begin
```

```
. . . . . .
  readln (ch);
    case ch of
     ‘n’, ‘N’ : writeln (‘ Нет’);
     ‘y’, ‘Y’ : writeln (‘ Да ‘)
end;
. . . . . .
```

4. Дан текст, содержащий только три буквы a, b, c и оканчивающийся точкой. Подсчитать количество вхождений каждой буквы. Написать программу с оператором CASE.

```
program variant;
var s : char;  k1, k2, k3 : integer;
begin
readln;
write ( ‘ => ‘);
k1:=0;  k2:=0;  k3:=0;
repeat
read (s);
case s of
 ‘  . ‘ : ;
 ‘ a ‘ : k1:= k1 + 1;
 ‘ b ‘ : k2:=k2 + 1;
 ‘ c ‘ : k3:=k3 + 1
end
until s = ‘ . ‘;
writeln (‘ букв a - ‘, k1 : 3,’ букв b – ‘, k2 : 3,’ букв c – ‘, k3 : 3)
end.
```

5. Селектор с явным описанием перечисляемого типа

```
var oper : ( plus, minus, mult );
. . . . . .
begin
. . . . . .
case oper of
plus :    x:=x + y;
minus :  x:=x – y;
mult :    x:=x * y
end;
```

3.4. Операторы цикла

Циклы – многократно повторяемые этапы вычислений, позволяющие выполнять группу операторов при изменении одного или нескольких параметров одновременно. В зависимости от постановки задачи различаются два основных типа: циклы с известным числом повторений и итерационные циклы. По структуре циклы делятся на простые (не содержат в себе других циклов) и сложные (содержащие в себе один или несколько циклов). Программы циклической структуры могут быть организованы с помощью условных операторов и операторов перехода. Однако в языке Паскаль имеются специальные операторы цикла: с параметром (FOR), с предусловием (WHILE) и с постусловием (REPEAT).

3.4.1. Оператор цикла FOR

Если число повторений цикла заранее известно, то чаще всего применяется оператор цикла с параметром. Общий вид

FOR I := M1 TO M2 DO S,

где I – параметр цикла (переменная порядкового типа);

M1 и M2 – выражения того же типа, что и параметр цикла, означающие соответственно начальное и конечное значения параметра цикла;

S – произвольный оператор (простой или составной).

При прочтении данного оператора ЭВМ присваивает параметру цикла I начальное значение M1

I := M1;

затем I сравнивается со своим конечным значением M2, то есть проверяется условие

I <= M2.

Если это условие выполняется, то выполняется оператор S, затем параметр цикла получает новое значение

I := SUCC(I),

и цикл повторяется. В противном случае оператор цикла FOR завершает свою работу.

Если параметр цикла целого типа, то при очередном повторе его значение увеличивается на единицу.

Примеры

1. Вычислить значения функции $y = x^i$ для I = 1,2, …, 10.

```
program cvp1;
var x,y : real;  I : byte;
begin
      write (‘ введите x’);
      readln (x);
      y := 1;
      for I := 1 to 10 do
            begin
                  y := y * x;
                  writeln ( ‘ y=’,y : 8 : 3);
            end
end.
```

Отметим, что параметром цикла может быть любой порядковый тип, например, символьный.

2. Вывести на экран последовательность латинских букв a, b, c, …, z. Несложная программа может иметь вид:

```
program ch1;
var i, j, k : integer;
begin
      i := ord (‘a’);    j := ord (‘z’);
      for k := i to j do write (chr (k), ‘ ‘)
end.
```

Здесь возможна и более изящная реализация:

```
program ch2;
var c : char;
begin
      for c := ‘a’ to ‘z’ do write ( c, ‘ ‘)
end.
```

Существует еще одна форма счетного оператора цикла FOR

FOR I := M1 DOWNTO M2 DO S;

Замена зарезервированного слова TO на DOWNTO означает, что параметр цикла на очередном повторе принимает новое значение согласно выражению

I := PRED (I);

а управляющее условие приобретает вид

I >= M2

Цикл завершается, как только I < M2. Если параметр цикла целого типа, то при очередном повторе его значение уменьшится на единицу.

Таким образом, задать в операторе цикла FOR шаг изменения параметра цикла, отличный от +1 или -1, нельзя.

Для оператора FOR существуют следующие ограничения.

1. Начальное и конечное значения параметра цикла изменять внутри цикла нельзя.

2. Войти в цикл можно только через его начало (заголовок цикла), а выйти – либо по завершении циклической процедуры, либо при выполнении оператора перехода по метке, расположенной вне данного цикла.

Итак, оператор FOR следует использовать в тех случаях, когда заранее известно число повторений или его можно подсчитать. Например, при изменении значения переменной от $x_н$ до $x_к$ с шагом h количество повторений n подсчитывается по формуле

$$n = [\frac{x_к - x_н}{h}] + 1 .$$

Квадратные скобки указывают на то, что результат округляется до целой части путем отбрасывания дробной.

Примеры

1. Протабулировать функцию $y = tg^2 \frac{x}{2} + 1$ в интервале [$x_н$, $x_к$] с шагом $\Delta x = h$

```
program cvp2;
var xn, xk, h, y : real;  i, n : integer;
begin
      writeln (‘ введите xn, xk, h’);
      readln (xn, xk, h);
x: = xn;
n: = trunc ((xk – xn)/h) + 1;
            begin
                  y := sqr( sin ( x/2)/ cos ( x/2 )) +1;
                  writeln ( x: 10 : 2,  y : 16 : 2);
                  x: = x+h
    end
      end.
```

2. Дан текст, содержащий 30 символов. Подсчитать количество вхождений символов a и в.

```
program   kol;
   var  k, l, i : integer; a : char;
   begin
      k: = 0 ;  l: = 0;
         for  i : = 1 то 30 do
            begin
         read ( a);
           if a = ‘ a ‘  then k: = k+1
                        tlse if a = ‘ b ‘ then l := l +1;
            end;
         readln;
         writeln ( ‘ символов a-‘,k : 3,’  символов в-‘ , l : 3)
   end.
```

Контрольные вопросы и упражнения

1. Найти значения s для следующих циклов с параметром при условии, что раздел описания переменных имеет следующий вид:

Var s, i : integer; sim : char ; l : boolean;

а) s: = 0; for i : = 5 to 7 do s : = s+1;

б) s: = 0; for i: = 10 downto 6 do s: = s+1;

в) s: =0;   for sim: = ‘a’ то ‘d’ do s: = s+1;

г) s: = 0;   for l : = false то true do s: = s+1;

2. Найти сумму целых положительных чисел, кратных 4, из диапазона [1, 100].

3. Найти сумму целых положительных четных чисел, меньших 100.

4. Найти сумму целых положительных нечетных чисел, меньших 200.

5. Найти сумму целых положительных чисел, больших 20, меньших 100 и кратных 3.

3.4.2. Оператор цикла WHILE

Оператор цикла с предусловием позволяет организовать цикл с неизвестным числом повторений. Общий вид записи оператора

WHILE B DO S;

где B – выражение логического типа, S – простой или составной оператор (тело цикла).

Выполнение оператора начинается с вычисления значения B. Если оно имеет значение TRUE, то выполняются операторы, входящие в тело цикла. Как только логическое выражение примет значение FALSE, выполнение операторов цикла прекращается. Если выражение булевского типа было ложно сразу при первом входе в цикл, то оператор S не выполнится ни разу. Очевидно, что один из операторов, находящихся внутри цикла должен влиять на значение выражения B, поскольку иначе цикл будет повторяться бесконечно.

Примеры

1. Вывести на экран последовательность четных чисел от 2 до 12.

```
program test1;
var k : integer;
begin
      k := 0;
      while k <= 10 do
            begin
                  k := k +2;  write (k : 4)
            end
end.
```

2. Составить программу для вычисления и вывода на экран функции $y = \frac{2x\sqrt{x+1} - e^{-x}}{x^2 + 1}$ при изменении x от -2 до 3 с шагом 0.25.

```
program f1;
var x, y : real;
begin
      x := -2;
      while x <= 3 do
            begin
```

```
            y := (2 * x * sqrt (x + 1) – exp (-x)) / ( x * x +1);
            writeln (x : 6 : 1, y : 7 : 3);
            x := x + 0.25
        end
end.
```

3. Составить программу для определения наименьшего целого положительного к, при котором функция $\frac{x^k}{k}$ становится меньше заданного а.

Очевидно, что в этом примере число повторений цикла заранее не известно, оно зависит от значений х и а. Логическим выражением здесь является выражение $\frac{x^k}{k} < a$.

```
program f2;
var x, y, a, p : real;  k : integer;
begin
    writeln (‘ введите х, а’);
    readln (x, a);
    k := 1; p := x;
    while p / k < a do
        begin
        p := p * x;
        k := k + 1
        end;
    writeln (k)
end.
```

В этой программе для сокращения вычислений выражение $\frac{x^k}{k}$ для значений k = 1, 2, … определяется как произведение p := p * x, начальное значение p равно x.

4. Дано целое число n. Подсчитать количество цифр данного числа.

```
program f3;
var m, n : integer;
k : integer;   { счетчик цифр }
begin
    writeln (‘ введите целое число ‘);
    readln (n);
```

```
k := 0;  m := n;
        while m <> 0 do
                begin
                  k := k + 1;
                  m :=m div 10 { уменьшение числа на
                                 последнюю цифру }
                end;
            writeln (‘ количество цифр в числе n’, n, ‘ равно ‘, k)
end.
```

Контрольные вопросы и упражнения

1. Какое значение примет переменная f после вычисления следующих операторов:

а) i := 1; f := 2; while i < 6 do begin i := i + 1; f := f + i end;

б) i :=1; f := 2; while I < 6 do begin i := i + 1; f := f * I end;

2. Сколько раз выполнится тело цикла для следующих операторов?

x := 2; f := 3; while x <= 5 do begin f := f + sqr (x); x := x + 0.25 end;

3. Дана последовательность операторов:

```
a := 1;  b := 1;
while a + b < 8 do begin a := a + 1; b := b + 2 end;
s := a + b;
```

Сколько раз выполнится проверка логического выражения в операторе while? Определите значения переменных a, b и s после выполнения этой последовательности операторов.

3.4.3. Оператор цикла REPEAT

Оператор цикла с постусловием также не требует предварительного определения числа повторений. В отличие от оператора WHILE в операторе REPEAT условие проверяется после выполнения тела цикла.

Общий вид

```
REPEAT
    S1;
    S2;
    . . .
```

```
        SN
    UNTIL B;
```

где S1, S2, . . . , SN – операторы тела цикла;

B – логическое выражение.

При прочтении данного оператора ЭВМ сначала выполняет операторы тела цикла. Затем вычисляется выражение B и если оно имеет истинное значение, то осуществляется выход из цикла. Если же значение выражения B ложно, то выполнение операторов S1, S2, … , SN повторяется, а затем снова вычисляется выражение B.

Отметим, что тело цикла REPEAT может содержать произвольное число операторов без записи их в составном операторе.

Примеры

Составим программы для некоторых примеров предыдущего раздела с использованием оператора REPEAT.

1. Вывести на экран последовательность четных чисел от 2 до 12.

```
program test2;
var k : integer;
begin
    k := 0;
    repeat
        k := k + 2; writeln (k : 3)
    until k > 10
end.
```

2. Составить программу для вычисления и вывода на экран функции $y = \frac{2x\sqrt{x+1} - e^{-x}}{x^2 + 1}$ при изменении x от -2 до 3 с шагом 0.25.

```
program f2;
var x, y : real;
    begin
        x := -2;
        repeat
          y := (2 * x * sqrt (x+1) – exp (-x)) / (x * x+1);
          writeln (x : 6 :1, y : 7 : 3);
          x := x + 0.25
        until x > 3
    end.
```

3. Составить программу, которая вводит символ и выводит на экран его код

```
program char_code;
const cr = 13;
var ch : char;
begin
      repeat
        writeln (‘ введите символ’);
        readln (ch);
        write (ch, ‘ = ‘, ord (ch))
      until ord (ch) = cr
end.
```

Оператор цикла repeat повторяет команды тела цикла до тех пор, пока условие, записанное после until, не станет истинным, то есть для завершения работы программы нужно дважды нажать ENTER.

Контрольные вопросы и упражнения

1. Определите значение переменной s после выполнения следующих операторов:

```
s := 0;  i := 2;
repeat s := s+1 / i;  i := i – 1  until i < 1;
```

2. Пользуясь оператором repeat, описать вычисление f.
3. Определить какое количество последовательных натуральных чисел необходимо сложить, чтобы их сумма превысила 200.

3.5. РАБОТА С ЭВМ

3.5.1. Циклический вычислительный процесс

Цель задания – получение навыков в использовании операторов цикла (тип цикла по указанию преподавателя).

Постановка задачи.

Составить программу вычисления функции f (x) на отрезке [a, b] в точках $x_i = a + ih$, где $h = \frac{(b-a)}{m}$, m – заданное число.

Содержание отчета:

1. Постановка задачи (вариант).
2. Текст программы.
3. Таблица результатов.

4. Анализ результатов и допущенных ошибок.

Методические рекомендации

1. Для задания значений x и соответствующих значений функции использовать простые переменные.
2. Значение шага h должно вычисляться один раз.
3. При изменении значения аргумента x использовать оператор присваивания x := x + h, а не операцию умножения x := a + i * h, что существенно сокращает время выполнения программы.

Таблица 1

Варианты заданий

№	Табулируемая функция	Параметры		
		a	b	m
1	x – sinx	0	$\frac{\pi}{2}$	10
2	sinx	$\frac{\pi}{4}$	$\frac{\pi}{2}$	15
3	cosx	$\frac{\pi}{3}$	$\frac{2\pi}{3}$	20
4	arcsinx	0	1	20
5	arccosx	0.5	1	10
6	arctgx	2	7	15
7	sinx - cosx	0	$\frac{\pi}{2}$	20
8	x ·sinx	0	3 π	10
9	$\sin\frac{1}{x}$	$\frac{\pi}{8}$	$\frac{2}{\pi}$	15
10	$\cos\frac{1}{x}$	$\frac{\pi}{4}$	$\frac{4}{\pi}$	20
11	$\sin(x^2)$	$\frac{\pi}{6}$	$\frac{2\pi}{3}$	10
12	$\cos(x^2)$	$\frac{\pi}{3}$	$\frac{3\pi}{2}$	15
13	$\mathrm{tg}\frac{x}{2}$	0	$\frac{2\pi}{3}$	15

3.5.2. Разветвление в цикле

Составить программу для вычисления y с оператором цикла и номером варианта по указанию преподавателя.

На основании табл. 2 для вычисленного номера варианта написать программу на алгоритмическом языке Паскаль с использованием любого оператора цикла.

Таблица 2

Варианты к выполнению задания 3.5.2

№ варианта	Табулируемая функция	Пределы изменения x	Шаг
0	$y=\begin{cases}\ln x, & 0<x\le 1\\ 4\sin(x+1), & x>1\\ h+x, & x\le 0\end{cases}$	[-1,3; 1,8]	0,2
1	$y=\begin{cases}\sqrt{x+5x^3}, & x\ge 0{,}5\\ \ln x, & 0<x<0{,}5\\ n-x, & x\le 0\end{cases}$	[-1; 0,8]	0,1
2	$y=\begin{cases}\sin(2x+1), & x\le 4\\ x^2-\sqrt{x}, & 4<x\le 6\\ 3\ln x, & x>6\end{cases}$	[3,0; 7,5]	0,25
3	$y=\begin{cases}\sin x^2, & x\le -1\\ \cos 2x, & 6\le x\\ 4x, & -1<x<6\end{cases}$	[-5; 10,1]	0,9
4	$y=\begin{cases}\sin x, & x>20\\ \sqrt{x}, & 5<x\le 20\\ \cos^2 x, & x\le 5\end{cases}$	[0,5; 25]	1,5
5	$y=\begin{cases}15x+7, & 4\le x\\ 20x^2, & x\le 1\\ 4\ln x, & 1<x<4\end{cases}$	[0; 5,1]	0,25

Продолжение табл. 2

№ варианта	Табулируемая функция	Пределы изменения x	Шаг
6	$y = \begin{cases} \cos\ x^2, & x < 2 \\ x^3, & 2 \le x \le 9 \\ \sin\ \sqrt{x}, & x > 9 \end{cases}$	[-3,4; 14,7]	1
7	$y = \begin{cases} \ln(x-5), & x > 6 \\ \sin x - r, & x \le -2 \\ \cos x, & 6 \ge x > -2 \end{cases}$	[-4,8; 10,4]	0,91
8	$y = \begin{cases} \dfrac{x^2}{x^3+1}, & x \ge 0{,}5 \\ \cos^3 x^4, & x \le -1 \\ \dfrac{u}{2}, & -1 < x < 0{,}5 \end{cases}$	[-2,3; 1,2]	0,2
9	$y = \begin{cases} \dfrac{x^5}{34}, & -1 \le x \le 4 \\ \cos(x+5), & x < -1 \\ \dfrac{\ln x}{2}, & x > 4 \end{cases}$	[-3; 7,2]	0,82
10	$y = \begin{cases} \sqrt{x + \operatorname{tg}\frac{x}{2}}, & x > 2 \\ \operatorname{ctg}\frac{x}{3} + \cos x, & -1 < x < 0 \\ x^2 - 4x + 7, & x < -1 \end{cases}$	[-4.8, 3.6]	0.4
11	$y = \begin{cases} \ln(x-1), & x > 2 \\ \cos x + \operatorname{ctg} x, & -1 < x \le 2 \\ \cos(x+1), & x < -1 \end{cases}$	[-3, 6]	0.5

Продолжение табл. 2

№ варианта	Табулируемая функция	Пределы изменения x	Шаг
12	$y=\begin{cases}3x^9+1, & 0<x\le 1\\ \cos\frac{x}{4}+1.8, & x\le 0\\ \cos x+\ln x, & x\ge 1\end{cases}$	[-2, 3.5]	0.5
13	$y=\begin{cases}\sqrt{x}+\frac{\sin x}{x}, & 0<x<4\\ 9x-\operatorname{ctg}\frac{x}{2}, & x\le 0\\ 1+\ln(x+1), & x\ge 4\end{cases}$	[-2, 8]	0.8
14	$y=\begin{cases}x^4-5x+1, & x>0\\ 3\cos x, & -3\le x\le 0\\ \cos x\sin x, & x<-3\end{cases}$	[-6, 3]	1.0

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Гильмутдинов В.И. Информатика: учеб. пособие / В.И. Гильмутдинов, А.Д. Кононов, А.А. Кононов. – Воронеж: ВГАСУ, 2010. – 56 с.

2. Кононов А.Д. Информатика: учеб. пособие / А.Д. Кононов, А.А. Кононов. – Воронеж: ВГАСУ, 2016. – 53 с.

3. Кононов А.Д. Основы программирования на языке Паскаль: учебное пособие / А.Д. Кононов, А.А. Кононов. – Воронеж: ВГТУ, 2017. – 53 с.

4. Кудинов Ю.И. Основы современной информатики / Ю.И. Кудинов, Ф.Ф. Пащенко. – М: Лань, 2010. – 256 с.

5. Могилев А.В. Информатика: учеб. пособие для пед. вузов / А.В. Могилев, Н.И. Пак, Е.К. Хеннер. – М.: Академия, 2004. – 848 с.

6. Окулов С.М. Основы программирования / С.М. Окулов. – М.: БИНОМ, 2013. – 440 с.

7. Хлебников А.А. Информатика: учебник / А.А. Хлебников. – Ростов н/Д: Феникс, 2013. – 443 с.

СОДЕРЖАНИЕ

Введение……... 1
1. Стандартные типы данных……. 1
1.1. Структура программы……. 1
1.2. Описание стандартных типов данных…….……... 1
1.3. Перечисляемый и ограниченный типы данных…… 4
2. Простые операторы…… 8
2.1. Выражения и оператор присваивания……..… 8
2.2. Организация ввода – вывода…..…….. 10
2.3. Оператор безусловного перехода……..….. 14
2.4. Работа с ЭВМ……... 15
2.4.1. Организация ввода – вывода. Изучение стандартных типов данных……..…… 15
2.4.2. Вычисление выражений. Использование стандартных функций…….. 18
3. Структурные операторы…… 24
3.1. Составной оператор……..…… 24
3.2. Условные операторы……..... 24
3.3. Оператор выбора……..……. 28
3.4. Операторы цикла……..……. 31
3.4.1. Оператор цикла FOR……..…… 31
3.4.2. Оператор цикла WHILE……..…….. 35
3.4.3. Оператор цикла REPEAT……...…… 37
3.5. Работа с ЭВМ…… 39
3.5.1. Циклический вычислительный процесс……... 39
3.5.2. Разветвление в цикле……..…… 41
Библиографический список……... 43

Паскаль: типы данных, операторы

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

для самостоятельной работы по дисциплине
«Информатика» для студентов 2-го курса
всех направлений подготовки

Составители
Кононов Александр Давыдович
Кононов Андрей Александрович

Подписано в печать 04.05.2018.
Формат 60×84 1/16. Бумага для множительных аппаратов.
Усл.-печ. л. 2,9. Тираж 253 экз. Заказ №

ФГБОУ ВО «Воронежский государственный
технический университет»
394026 Воронеж, Московский проспект, 14

Участок оперативной полиграфии издательства ВГТУ
394026 Воронеж, Московский проспект, 14