

Методические указания к выполнению лабораторных работ

Лабораторные работы выполняются индивидуально в соответствии с вариантом задания. Перед началом работы необходимо изучить теоретический минимум, который дается в начале описания каждой лабораторной работы. Для лучшего усвоения теории к каждой лабораторной работе предлагается практический пример выполнения работы, тщательный разбор которого поможет студенту выполнить индивидуальное задание. Все лабораторные работы выполняются на языке Turbo Pascal 7.0.

Написанная и отлаженная программа после запуска на выполнение должна выводить информацию об авторе, номере варианта, назначении программы (приводится лабораторное задание полностью). Вводу данных с клавиатуры обязательно должно предшествовать текстовое сообщение о типе и количестве вводимых данных.

Указанные требования распространяются на все лабораторные работы.

Лабораторная работа № 1

Условный оператор

Задание: Используя разветвляющуюся структуру, составить блок-схему вычисления значения составной функции, имеющей различный вид на разных участках аргумента, затем составить программу, реализующую данный алгоритм (значение аргумента функции вводится с клавиатуры).

Теоретический материал

Алгоритм – это любая конечная последовательность основных математических и логических действий, однозначно определяющих процесс преобразования исходных данных в искомые результаты решения задачи.

Для представления алгоритмов используются несколько **способов**:

- словесный (описание на естественном человеческом языке);
- с помощью одного из алгоритмических языков программирования;
- графический (на языке блок-схем);
- с использованием диаграмм;
- с использованием таблиц решений;

и др.

Рассмотрим способ составления алгоритма на языке блок-схем.

Блок-схема – это графическая интерпретация алгоритма, представляющая набор геометрических фигур, каждая из которых изображает какую-либо операцию или действие.

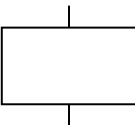
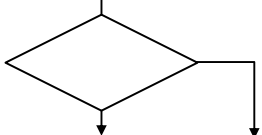
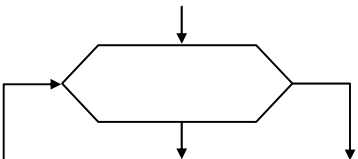
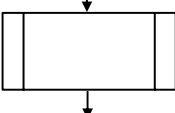
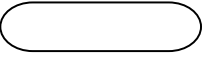

Правила построения алгоритмов на языке блок-схем

1. Блок-схема строится сверху вниз.
2. В любой блок-схеме имеется только один элемент, соответствующий началу алгоритма, и один элемент, соответствующий концу алгоритма.
3. Должен быть хотя бы один путь от начала блок-схемы к любому элементу.
4. Должен быть хотя бы один путь от каждого элемента блок-схемы к концу блок-схемы.

Основные элементы, которые будут использованы при построении блок-схемы решения задачи представлены в табл. 1.1.

Таблица 1.1

Основные элементы для построения блок-схем

Символ	Описание символа
	Процесс – формирование новых значений, выполнение арифметических или логических операций или действий, результаты которых запоминаются в оперативной памяти ЭВМ
	Ветвление – проверка условия: выбор одного из двух направлений выполнения алгоритма в зависимости от некоторого условия
	Модификация – организация циклических конструкций (начало цикла)
	Предопределенный процесс – вычисление по подпрограмме, использование ранее созданных и отдельно описанных алгоритмов
	Начало-конец программы или вход и выход в подпрограммах.
	Ввод–вывод данных – связь алгоритма с внешним миром. Вывод может осуществляться на бумагу, экран монитора, на магнитный диск.
	Комментарий – пояснения, содержание подпрограмм

Условный оператор `if`

Иногда требуется, чтобы часть программы выполнялась не всегда, а лишь при выполнении некоторого условия (а при невыполнении этого условия выполнялась другая часть программы). В этом случае пользуются оператором условного выполнения, который записывается в следующем виде:

```
if <условие> then <оператор1> else <оператор2>;
```

Под оператором понимается либо одиночный оператор (например, присваивания, вызова процедуры), либо так называемый составной оператор, состоящий из нескольких простых операторов, помещенных между словами `begin` и `end`. Важно заметить, что **перед `else` не ставится точка с запятой**. Часть `else` может и отсутствовать.

Пример выполнения лабораторной работы

В качестве примера рассмотрим составную функцию вида

$$y = \begin{cases} x - 2 & x > 2.5 \\ 1 + x^2 & 0 \leq x \leq 2.5 \\ x \cdot \ln|\cos(x)| & x < 0 \end{cases}$$

Как видно из задания, функция вычисляется на трех диапазонах аргумента x по трем различным формулам. Составим блок-схему алгоритма решения данной задачи (рис.1.1).

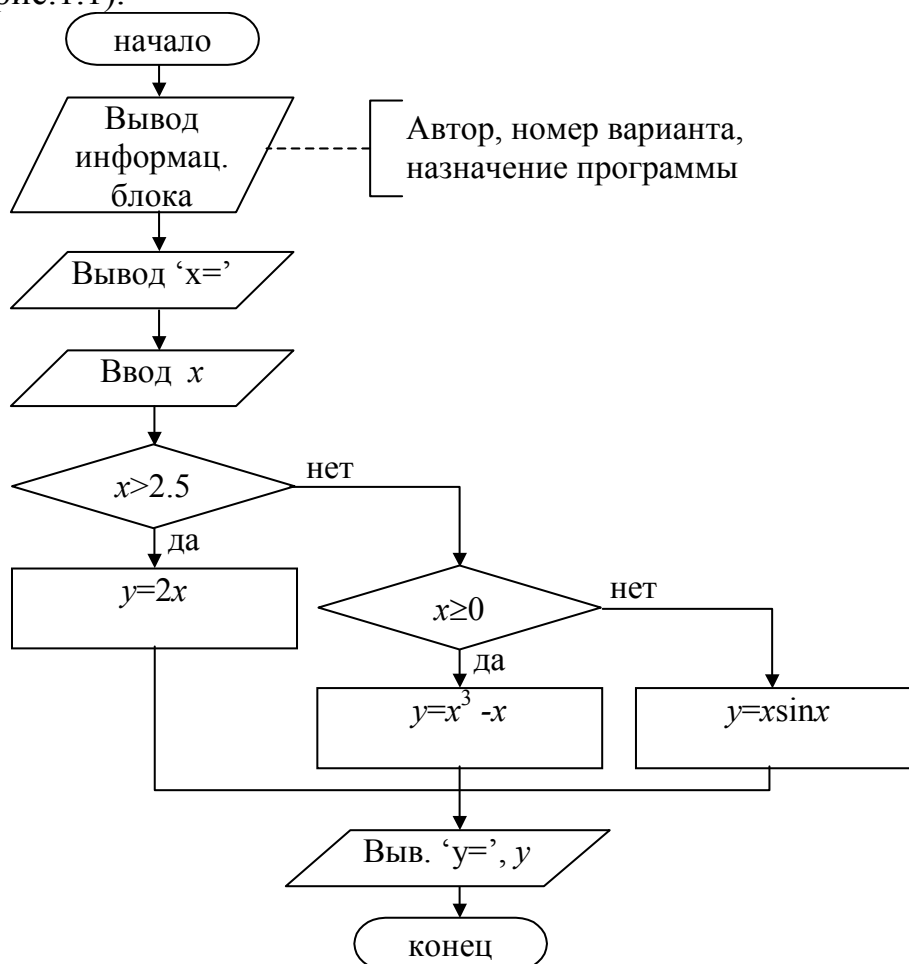


Рис. 1.1. Блок-схема вычисления значения составной функции

Теперь по данному алгоритму составим программу (а не наоборот) на языке Pascal.

Любую программу рекомендуется (но не обязательно) начинать с заголовка.
`Program Lab1_varX;`

Далее следует раздел описания переменных. В нашем примере понадобятся две переменные для хранения аргумента x и значения функции y . Так как и аргумент, и сама функция могут принимать дробные значения, то их необходимо описывать вещественным типом данных.

```
Var x, y: real;
```

Затем описываем основное тело программы. Как следует из блок-схемы алгоритма, вначале необходимо вывести информационный блок (автор, номер варианта, назначение программы), после чего вывести текстовую строку, которая подскажет пользователю, что нужно вводить значение аргумента. А затем считать с клавиатуры введенное пользователем число. Не забываем, что тело программы начинается с зарезервированного слова `begin`.

```
Begin
  writeln;
  writeln('      Автор - Иванов И.П., студент гр. ИСЭд-11');
  writeln('      Вариант № 100');
  writeln('Программа вводит значение аргумента X и вычисляет зна-
чение функции Y');
  writeln('      |2*x                x>2.5');
  writeln('Y= |x^3-x                0<=x<=2.5');
  writeln('      |x*(sin(x))         x<0');
  writeln;
  writeln('введите x= ');
  readln(x);
```

В данном фрагменте используется два вида оператора вывода: с параметром и без параметра. В первом случае на экран выводится текст, который указан в качестве параметра, а во втором случае – пустая строка (для того чтобы сделать отступ между строками).

После того, как значение X введено оператором ввода `readln(x)`, нужно определить, по какой формуле должна вычисляться функция. Для этого в алгоритме предусмотрены проверки значения аргумента.

```
If x>2.5 then
  Y:=2*x
Else
  If x>=0 then y:=x*x*x-x
  Else      Y:=x* sin(x);
```

Как видно, в данном фрагменте программы признак конца оператора (точка с запятой) ставится только один раз в самом конце, т. к. условный оператор заканчивается именно на последней строке, перед `else` точку с запятой не ставят.

В конце программы нужно вывести результаты вычислений и завершить тело программы служебным словом `end` с точкой.

```
writeln('Y= ', y:7:3);
readln
end.
```

Последний оператор вывода отличается от всех предыдущих. В данном случае стандартная процедура вывода `writeln` содержит два параметра, перечисленных через запятую. Первый параметр – это текстовая строка ('Y= '), которую надо вывести на экран. Второй параметр – переменная Y, значение которой требуется вывести в определенном формате, о чем говорят два числа, написанные через двоеточие. **Первое число – количество позиций, отводимых под вывод всего числа** (включая знак, целую часть числа, точку и дробную часть числа), а второе – число разрядов после запятой. В нашем случае под вывод всего числа запланировано 7 позиций, из них 3 позиции – под дробную часть, одна позиция – под точку, остается 3 позиции под целую часть и знак. Следует заметить, что если программист указал недостаточное количество позиций под вывод всего числа, то это число будет автоматически увеличено до требуемого для вывода значения.

Для того, чтобы результаты работы программы оставались на экране после выполнения программы, используем оператор ввода без параметров `readln`. В этом случае программа выполнит все необходимые действия и будет ожидать от пользователя нажатия клавиши Enter.

Итак, программа написана, ниже приведен ее полный текст.

```

Program Lab1_variant100;
Var x,y: real;
Begin
  writeln;
  writeln('      Автор - Иванов И.П., студент гр. ИСЭд-11');
  writeln('      Вариант № 100');
  writeln('Программа вводит значение аргумента X и вычисляет зна-
чение функции Y');
  writeln('      |2*x                x>2.5');
  writeln('Y= |x^3-x                0<=x<=2.5');
  writeln('      |x*(sin(x))        x<0');
  writeln;
  writeln('введите x= ');
  readln(x);
  If x>2.5 then
    Y:=2*x
  Else
    If x>=0 then
      y:=x*x*x-x
    Else
      Y:=x*sin(x);
  writeln('Y= ', y:7:3);
  readln
end.

```

Варианты заданий

№ вар.	Функция	№ вар.	Функция
1.	$y = \begin{cases} x - 2 & x > 2.5 \\ 1 + x^2 & 0 \leq x \leq 2.5 \\ x \ln \cos(x) & x < 0 \end{cases}$	2.	$y = \begin{cases} \sin(2.3x - 1) & x > 2.5 \\ 1 - 3 \ln 1 - x & 0 \leq x \leq 2.5 \\ \frac{x^2}{2 - x} & x < 0 \end{cases}$
3.	$y = \begin{cases} \sqrt{\operatorname{tg}(x^2 - 1)} & x > 1 \\ -2x & 0 \leq x \leq 1 \\ e^{\cos(x)} & x < 0 \end{cases}$	4.	$y = \begin{cases} x^2 - 3 + 2.5x^2 & x > 12.5 \\ e^x + 5 + \cos(0.001x) & 0 \leq x \leq 12.5 \\ x^2 & x < 0 \end{cases}$
5.	$y = \begin{cases} 1 + \sqrt{ \cos(x) } & x > 1 \\ x + 1 & -0.5 \leq x \leq 1 \\ 1 - x^2 & x < -0.5 \end{cases}$	6.	$y = \begin{cases} 2.5 \cdot x^3 + 6 \cdot x^2 - 30 & x > 1.5 \\ x + 1 & 0 \leq x \leq 1.5 \\ x & x < 0 \end{cases}$
7.	$y = \begin{cases} 1 + x & x > 14.5 \\ e^{-x} & 3 \leq x \leq 14.5 \\ \cos(x) & x < 3 \end{cases}$	8.	$y = \begin{cases} \ln 1 + x & x > 3.8 \\ e^{-x} & 2.8 \leq x \leq 3.8 \\ \cos(x) & x < 2.8 \end{cases}$
9.	$y = \begin{cases} 1 + \sqrt{\cos(x)} & x > 4 \\ x + 1 & 0 \leq x \leq 4 \\ 1 - x^2 & x < 0 \end{cases}$	10.	$y = \begin{cases} e^{-(x+8)} & x > 3.61 \\ 1 & 0 \leq x \leq 3.61 \\ \frac{x}{5} & x < 0 \end{cases}$
11.	$y = \begin{cases} x & x > 1.5 \\ 2x^2 \sqrt{ \cos(2x) } & 0 \leq x \leq 1.5 \\ e^{-\cos(3x)} & x < 0 \end{cases}$	12.	$y = \begin{cases} 1 - \sqrt{\cos(2x)} & x > 2.5 \\ x^2 - x & 1 \leq x \leq 2.5 \\ 1 + x^2 & x < 1 \end{cases}$

Окончание таблицы 1.2

13.	$y = \begin{cases} 2x & x > 4.5 \\ 1 - \ln 1 - x^2 & 0 \leq x \leq 4.5 \\ e^{-x} & x < 0 \end{cases}$	14.	$y = \begin{cases} \sqrt{\ln(x^2 - 1)} & x > 2 \\ -2x^3 & 0 \leq x \leq 2 \\ e^{\sin(x)} & x < 0 \end{cases}$
15.	$y = \begin{cases} 1 - \frac{2x^3}{1 - x^2} & x > 3.5 \\ \sqrt{\cos(2x - 1)} & 0 \leq x \leq 3.5 \\ e^{-\cos(2x)} & x < 0 \end{cases}$	16.	$y = \begin{cases} x + 1 & x > 2.5 \\ 1 - x^5 & 0 \leq x \leq 2.5 \\ x + \ln \sin(x) & x < 0 \end{cases}$
17.	$y = \begin{cases} x - 2 & x > 2.5 \\ 1 + x^2 & 0 \leq x \leq 2.5 \\ x \ln \cos(x) & x < 0 \end{cases}$	18.	$y = \begin{cases} 1 + 3x & x > 4.5 \\ e^{-2x} & 1 \leq x \leq 4.5 \\ \cos(2x) & x < 1 \end{cases}$
19.	$y = \begin{cases} \sqrt{ \operatorname{tg}(x^2 - 1) } & x > 4 \\ -2x & 0 \leq x \leq 4 \\ e^{\cos(x)} & x < 0 \end{cases}$	20.	$y = \begin{cases} e^{(x+2)} & x > 1 \\ 1 - 2x & -1 \leq x \leq 1 \\ -\frac{2x^3 - 3}{5} & x < -1 \end{cases}$
21.	$y = \begin{cases} 1 - x^3 & x > 2 \\ -2x & 0 \leq x \leq 2 \\ e^{\cos(x)} & x < 0 \end{cases}$	22.	$y = \begin{cases} 1 + \sqrt{ \operatorname{tg}(x) - 1 } & x < -3.14 \\ x & -3.14 \leq x \leq 3.14 \\ 1 + x^2 & x > 3.14 \end{cases}$
23.	$y = \begin{cases} \frac{1}{\ln(x^3)} & x > 4.5 \\ 2x + 0.1 & 0 \leq x \leq 4.5 \\ 3 & x < 0 \end{cases}$	24.	$y = \begin{cases} x^2 - 3 + 2.5x^3 & x > 2 \\ e^{(x)} + 5 + \cos(0.001x) & -1 \leq x \leq 2 \\ \ln \operatorname{tg}(2x) - 1 & x < -1 \end{cases}$
25.	$Y = \begin{cases} \cos(2.3x - 1) & x > 5.5 \\ 1 - 3 \ln(1 + x) & 0 \leq x \leq 5.5 \\ \frac{x^2}{2 - x} & x < 0 \end{cases}$		