

Лабораторная работа № 10

Разветвляющиеся вычислительные процессы

Цель работы: получение навыков написания программ, в которых выбор программой вариантов действий зависит от наступления определенных событий (ситуаций).

1. Теоретическая часть

В линейной программе все операторы выполняются последовательно, один за другим. Таким способом можно записывать только очень простые алгоритмы.

Для того чтобы в зависимости от конкретных значений исходных данных обеспечить выполнение разных последовательностей операторов, в программах на языке Паскаль применяются операторы ветвления *if* и *case*.

Оператор *if* обеспечивает передачу управления на одну из двух ветвей вычислений, а оператор *case* — на одну из произвольного числа ветвей.

Оператор варианта *case*

Оператор варианта (выбора) *case* используется для реализации разветвляющихся алгоритмов с множественным выбором. Формат оператора: ***case* выражение of**

константы_1 : оператор_1;

константы_2 : оператор_2;

...

константы_n : оператор_n-1

[else : оператор_n]

end;

При выполнении оператора *case* решение о том, какой именно из списка операторов - оператор_1, оператор_2, оператор_3 и т.д. надо выполнить, принимается после определения значения выражения после слова *case*. При известном значении выражения выполняется тот оператор, который записан после константы, значение которой совпало со значением выражения.

Если требуется выполнить одни и те же действия для нескольких констант, они записываются через запятую либо указывается диапазон значений констант.

Если по какой-либо ветви требуется записать не один, а несколько операторов, они должны образовывать составной оператор, т.е. заключаться между ключевыми словами *begin* и *end*.

Наличие строки, начинающейся со слова *else* в операторе *case*, не обязательно, оператор после *else* (оператор_n) соответствует случаю, когда значение выражения не совпадает ни с одной из констант.

Рассмотрим пример. Пусть необходимо написать программу, реализующую калькулятор на четыре арифметических действия.

Исходными данными для этой программы являются два вещественных операнда и знак операции, представляющий собой символ.

Пусть вид экрана во время выполнения программы будет следующим.

Простейший калькулятор.

Введите первый операнд: **5**

Введите операцию (+,-,*,/): *****

Введите второй операнд: **6**

Результат: **30**

Схема алгоритма работы программы (рис. 10.1) показывает, что в зависимости от значения введенного знака операции необходимо выполнить соответствующее действие над операндами. Программа, реализующая данный алгоритм, представлена на рис. 10.2.

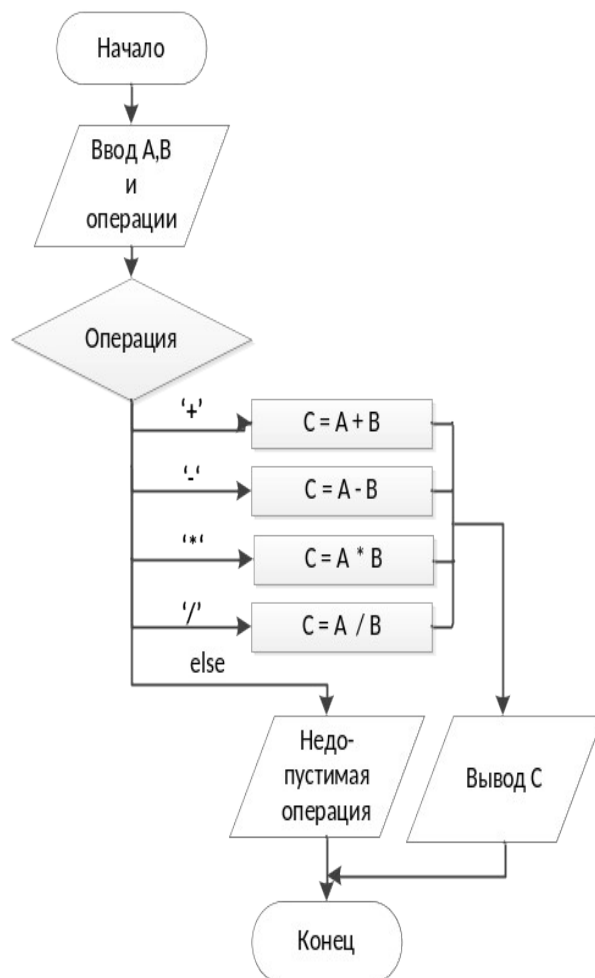


Рис. 10.1. Схема алгоритма для конструкции *case*

Program calculator:

```

var
  A, B, C : real; {исходные данные и результат}
  op      : char; {операция}
  error    : boolean; {признак недопустимой операции}
begin
  writeln('Простейший калькулятор. ');
  write(' Введите первый операнд : ');
  readln(A);
  write(' Введите операцию (+,-,*,/): ');
  readln(op);
  write(' Введите второй операнд: ');
  readln(B);
  error := false; {считаем, что введена действительная операция}
  case op of
    '+' : C := A + B;
    '-' : C := A - B;
    '*' : C := A * B;
    '/' : C := A / B;
  else
    begin
      writeln(' Недопустимая операция ');
      error := true;
    end;
  end; {case}
  if not error
  then writeln(' Результат: ',C:6:2);
  end. {конец программы}
  
```

Рис. 10.2. Программа с оператором *case*

Условный оператор *if*

Полный формат оператора имеет вид:

***if* выражение *then* оператор_1 *else* оператор_2 .**

Выражение, записанное после слова *if*, должно иметь логический тип (логическое выражение). При выполнении оператора *if* сначала вычисляется значение этого выражения. Если выражение имеет значение *true*,

выполняется оператор 1, иначе - оператор 2 (рис. 10.3, а). Ветвь *else* оператора *if* может отсутствовать (рис. 10.3, б), и в этом случае оператор имеет следующий сокращенный вид:

if выражение then оператор

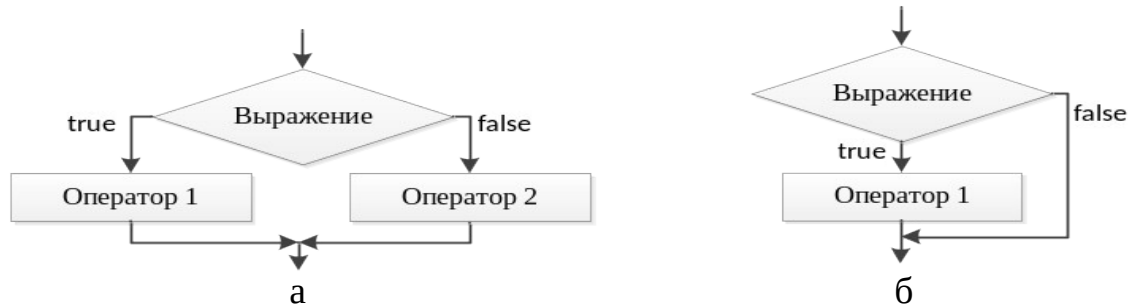


Рис. 10.3. Структурная схема условного оператора