МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

Юго-Западный государственный университет

(ЮЗГУ)

Кафедра вычислительной техники

ПРОГРАММИРОВАНИЕ РАЗВЕТВЛЕННЫХ АЛГОРИТМОВ

Методические указания по выполнению лабораторной работы

для студентов направления подготовки 09.03.01

Курск 2016

УДК 621.3
Составитель: Э.И. Ватутин

Рецензент
Кандидат технических наук, доцент *В.С. Панищев*

Программирование разветвленных алгоритмов: методические указания по выполнению лабораторных работ по дисциплине «Программирование» / Юго-Зап. гос. ун-т; сост.: Э.И. Ватутин; Курск, 2016. 10 с.: ил. 1.

Методические рекомендации содержат сведения по разработке разветвленных программ на современных языках программирования высокого уровня.

Предназначены для студентов направления подготовки 09.03.01 «Информатика и вычислительная техника».

Текст печатается в авторской редакции

Подписано в печать _____. Формат 60x84 1/16.
Усл. печ. л. Уч. – изд.л. Тираж 30 экз. Заказ . Бесплатно.
Юго-Западный государственный университет
305040, Курск, ул. 50 лет Октября, 94.

Содержание

Введение

Целью работы является получение практических навыков при программировании разветвленных программ с использованием условного оператора и оператора выбора.

Область применения линейных программ, в которых операторы выполняются строго последовательно, существенно ограничена. В реальных программах зачастую требуется разветвление вычислительного процесса, т.е. выполнение тех или иных действий в зависимости от значения некого условия (например, мы берем с собой зонт если на улице идет дождь).

Условный оператор (или *оператор условия*) позволяет разветвление вычислительного процесса на две ветви, называемые *условными* или *альтернативными*, в зависимости от значения условия булевского типа. Граф-схема и соответствующая ей конструкция языка приведены ниже.

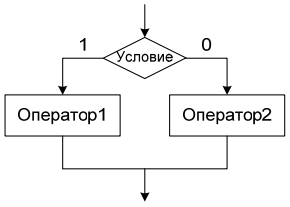

Рис. 1. Граф-схема условного оператора

Фрагменты программ, соответствующие граф-схеме на рис. 1, приведены ниже.

Delphi

```
if Условие then
  Оператор1
else
  Оператор2;
```

C++

```
if (Условие)
  Оператор1;
else
  Оператор2;
```

Оператор 1 выполняется в случае истинности значения условия, при ложном значении выполняется оператор 2. В случае необходимости

допускается отсутствие в операторе ветви `else` (подобная форма оператора иногда называется *неполной альтернативой*):

Delphi

```
if Условие then
  Оператор;
```

C++

```
if (Условие)
  Оператор1;
```

В Delphi символ «;» отделяет операторы друг от друга. При записи условного оператора перед ключевым словом `else` никогда не ставиться точка с запятой, т.к. в языке нет оператора, начинающегося ключевым словом `else`. В языке C++ перед `else` всегда ставится точка с запятой.

В случае необходимости условные операторы могут быть вложены друг в друга, причем степень вложенности ничем не ограничена. Пример подобной ситуации приведен ниже.

Delphi

```
if Условие1 then
  if Условие2 then
    Оператор1
  else
    Оператор2;
```

C++

```
if (Условие1)
  if (Условие2)
    Оператор1;
  else
    Оператор2;
```

В приведенном примере имеется два условных оператора, первый из которых является неполной альтернативой, а второй выполняется в случае истинности значения условия 1. Ветвь `else` относится к ближайшему (наиболее вложенному) оператору ветвления (в данном случае ко второму). В случае необходимости соотнесения ветви `else` с первым оператором необходимо использовать операторные скобки:

Delphi

```
if Условие1 then begin
  if Условие2 then
    Оператор1;
end else
  Оператор2;
```

C++

```
if (Условие1) {
  if (Условие2)
    Оператор1;
} else
  Оператор2;
```

Пример программы

В качестве примера рассмотрим вычисление значения функции

$$y = \begin{cases} \dfrac{1}{x-3}, & x < 4; \\ \sqrt{x^2 - 50}, & 4 \le x \le 8; \\ \ln^2\left(\dfrac{x}{10} - 4\right) & \text{иначе.} \end{cases}$$

Во избежание появления сообщений об ошибках и аварийного прерывания процесса выполнения программы будем проверять некорректные ситуации (например, попытку деления на ноль) и выдавать соответствующие сообщения. Соответствующая программа на Delphi приведена ниже.

```
var
  X, Y, Tmp: Double;

begin
  Write('X = ');
  Readln(X);

  if X < 4 then begin
    { Проверка деления на ноль }
    if X = 3 then begin
      Writeln('Division by zero');
      Readln;
      exit;
    end;
    Y := 1 / (X - 3);
  end else if X <= 8 then begin { Проверку условия 4 <= X можно опустить }
    Tmp := X*X - 50;    { Сохраняем значение подкоренного выражения, чтобы не вычислять
                          его дважды }
    if Tmp < 0 then begin
      Writeln('Negative argument of square root');
      Readln;
      exit;
    end;
    Y := Sqrt(Tmp);
  end else begin
    Tmp := X/10 - 4;
    if Tmp < 0 then begin
      Writeln('Negative argument of logarithm');
      Readln;
      exit;
    end;
    Y := Sqr(Ln(Tmp));
  end;

  Writeln('Y = ', Y:5:2);
  Readln;
end.
```

Аналогичная программа на языке C++ приведена ниже.

```
#include <iostream>
```

```
using namespace std;

void main()
{
	double X, Y, T;

	cout << "X = ";
	cin >> X;

	if (X < 4.0) {
		// Проверка деления на ноль
		if (X == 3.0) {
			cout << "Division by zero";
			system("pause");
			return;
		}
		Y = 1 / (X - 3.0);
	} else if (X <= 8.0) { // Проверку условия 4 <= X можно опустить
		T = X*X - 50.0;    // Сохраняем значение подкоренного выражения, чтобы
				   // не вычислять его дважды
		if (T < 0.0) {
			cout << "Negative argument of square root";
			system("pause");
			return;
		}
		Y = sqrt(T);
	} else {
		T = X/10.0 - 4.0;
		if (T < 0.0) {
			cout << "Negative argument of logarithm";
			system("pause");
			return;
		}
		T = log(T);
		Y = T*T;
	}

	cout << "Y = " << Y << endl;
	getchar();
}
```

Индивидуальные задания

1. Проверить, является ли введенное число целым квадратом. Например, $25 = 5^2$ – целый квадрат, а $15 = 3 \cdot 5$ – нет.
2. Проверить, является ли введенное число из 4 цифр палиндромом. (Палиндром – число, читающееся слева направо и справа налево одинаково. Например, «1221» – палиндром, а «1254» – нет.)
3. Проверить, является ли введенный номер трамвайного билета счастливым (билет считается счастливым, если сумма первых трех его цифр равна сумме трех последних).

4. Найти решение системы уравнений $\begin{cases} ax + by = c \\ dx + ey = f \end{cases}$ по заданным значениям коэффициентов a, b, c, d, e, f. Предусмотреть случаи, когда решение не существует или является не единственным.
5. Найти действительные корни биквадратного уравнения $ax^4 + bx^2 + c = 0$ при известных коэффициентах a, b, c. В случае отсутствия корней выдать соответствующее сообщение.
6. Заданы координаты расположения двух коней на шахматной доске. Определить, может ли один из коней атаковать другого.
7. Заданы координаты расположения двух ладей на шахматной доске. Определить, может ли одна из ладей атаковать другую.
8. Заданы координаты расположения двух слонов на шахматной доске. Определить, может ли один из слонов атаковать другого.
9. Заданы координаты расположения двух ферзей на шахматной доске. Определить, может ли один из ферзей атаковать другого.
10. По заданным длинами сторон определить, является ли треугольник остроугольным, прямоугольным или тупоугольным. Предусмотреть случай, когда указанные длины сторон не могут образовать треугольник.
11. Задана точка с координатами на плоскости. Определить ее принадлежность к координатным четвертям, осям или началу координат.
12. Заданы координаты центра окружности, ее радиус и координаты точки. Определить, лежит ли точка в пределах окружности, на окружности или за ее пределами.
13. На плоскости заданы две прямые $y = a_1 x + b_1$ и $y = a_2 x + b_2$. Определить координаты точки их пересечения. Предусмотреть случай, когда прямые параллельны.
14. На плоскости заданы две прямые $y = a_1 x + b_1$ и $y = a_2 x + b_2$. Определить, под каким углом они пересекаются. Предусмотреть случай, когда прямые перпендикулярны.

15. По введенной длине волны излучения определить его цвет.
16. По введенным оценкам *A*, *B*, *C* и *D* за сессию определить размер стипендии (стипендия не выплачивается при наличии хотя бы одной тройки, повышенная стипендия выплачивается при наличии всех пятерок, в противном случае выплачивается обычная стипендия).
17. Составить программу, которая вводит значения *a*, *b*, *c*, сравнивает их между собой и перераспределяет таким образом, чтобы *a* содержала наименьшее из значений, *b* – среднее, *c* – максимальное. Результат выводится.
18. Вводятся координаты трех точек. Определить, лежат ли точки на одной прямой.
19. Заданы декартовы координаты точки на плоскости. Определить ее полярные координаты $r = \sqrt{x^2 + y^2}$ и $\theta = \operatorname{arctg}\frac{y}{x}$. Предусмотреть возможность деления на ноль и возможность нахождения точки в различных координатных четвертях.
20. Среди значений *a*, *b*, *c* выбрать значение, отличающееся от *x* на наименьшую величину.

Содержание отчета

1. Титульный лист.
2. Индивидуальное задание.
3. Краткое описание стратегии решения.
4. Листинг программы.
5. Тестовые примеры, результаты тестирования.
6. Выводы.

Контрольные вопросы

1. Что представляют из себя разветвленные программы?
2. Что называется неполной альтернативой?
3. Что такое составной оператор и как он применяется совместно с операторами ветвлений?
4. Какие ключевые слова используются при записи условных ветвлений?

Библиографический список

1. Емельянов С.Г., Ватутин Э.И., Панищев В.С., Титов В.С. Процедурно-модульное программирование на Delphi: учебное пособие. М.: Аргамак-Медиа, 2014. 352 с.
2. Зотов И.В., Ватутин Э.И., Борзов Д.Б. Процедурно-ориентированное программирование на C++: учебное пособие. Курск: КурскГТУ, 2008. 211 с.