Лабораторная Работа

На тему: "**Оператор условия if-else"**

**1. Назначение и синтаксис**

Оператор условия if-else служит для выбора направления работы программы в зависимости от условий, сложившихся в данной точке программы на момент ее выполнения.

Общая форма записи условного оператора

**if (**<условие>**)**

**{**

<блок операторов 1>;

**}**

**else**

**{**

<блок операторов 2>;

**}**

Если на момент выполнения <условие> истинно, программа передает управление <блоку операторов 1> и, далее, первому оператору за пределами конструкции if-else. При этом <блок операторов 2> не выполняется. Иначе, если <условие> ложно, выполняется <блок операторов 2>, а <блок операторов 1> пропускается. Соответствующая блок-схема

блок операторов 1

условие

блок операторов 2

true (+)

false (-)

Фигурные скобки в синтаксисе оператора if-else используются для выделения в тексте блоков 1 и 2. Старайтесь располагать закрывающую скобку под открывающей для улучшения читаемости программного кода. Для этой же цели текст внутри фигурных скобок необходимо смещать вправо на несколько позиций.

В качестве условия в операторах if-else могут использоваться любые логические выражения, принимающие значения «истинно» или «ложно» (true – false). Ниже приведена таблица с указанием простейших операций сравнения целых и вещественных чисел

|  |  |  |
| --- | --- | --- |
| *операция* | *знак операции* | *пример записи на C/С++* |
| больше | > | A > B |
| меньше | < | C < 2 |
| больше либо равно | >= | k >= 0 |
| меньше либо равно | <= | i <= 10 |
| равно | == | x == y |
| не равно | != | D!= 0 |

**Пример 1.** Требуется написать программу, переводящую температуру по шкале Цельсия TC (°С) в температуру по Кельвину TK (К). Значение TC вводится пользователем с клавиатуры.

Решение. Используем известную формулу преобразования – TK = TC – T0, где T0 = –273 °С – температура абсолютного нуля. Будем считать введенное TC некорректным, если оно меньше T0.

 // – Kelvin vs Celsius –

#pragma hdrstop

#include <iostream.h> // для потокового В/В

#include <conio.h> // для консольного В/В (getch)

#pragma argsused

int main (int argc, char\* argv[])

{

float T0 = -273; // объявляем и инициализируем T0

float Tc, Tk; // объявляем вещественные Tc и Tk

cout<<» Vvedite Tc =»; // выводим приглашение

cin>>Tc; // запрашиваем Tc

**if (Tc<T0)**// проверяем условие Tc<T0

**{**

cout<<» Tc < T0!»; // условие истинно, выводим на

**}** // экран сообщение об ошибке

**else**

**{**

Tk = Tc-T0; // условие ложно, рассчитываем

cout << «Tk =» << Tk; // Tk и выводим на экран

**}**

getch(); // задержка до нажатия на клавишу

return 0; // завершение программы

}

 // –

Наберите приведенный выше код, откомпилируйте его и запустите программу на выполнение. Исследуйте результаты работы при различных значениях TC.

**2. Сокращенные варианты записи**

При программировании обыденной является ситуация, когда требуются некоторое действие в ответ на сложившиеся условия (например, если получены неверные исходные данные от пользователя, то выдать сообщение об ошибке и выйти из программы). В таких случаях в программах на C++ может использоваться сокращенная запись оператора условия с отсутствующим блоком else. Общая форма такой записи

**if (**<условие>**)**

**{**

<блок операторов>;

**}**

Здесь в случае истинности условия управление передается блоку операторов в фигурных скобках. В случае ложности условия этот блок пропускается. Соответствующая блок-схема отличается от предыдущей отсутствием одного «плеча»

блок операторов

условие

true (+)

false (-)

Другой вариант сокращения применяется в случае, когда какой-либо из блоков if или else состоит только из одного оператора. В этом случае допускается отсутствие фигурных скобок, ограничивающих этот блок.

**if (**<условие>**)**

оператор 1;

**else**

оператор 2;

Здесь в качестве операторов 1 и 2 могут быть не только простые однострочные арифметические действия или операторы ввода-вывода, но и сложные многострочные конструкции, например другие (вложенные) операторы условия или операторы цикла, которые будут рассмотрены ниже.

**3. Вложенные операторы**

Операторы условия могут быть вложенными друг в друга, в соответствии с тем программным алгоритмом, который они реализуют. Допускается произвольная степень их «вложенности».

Если один оператор if-else вложен в другой, то первый оператор входит во второй *полностью*, а не только какой-либо одной своей частью if или else. Недопустимым является частичное перекрывание между собой их отдельных блоков.

|  |  |
| --- | --- |
| A) правильная запись  if (a<=b)  {  **if (x!=0)**  **cout<<» x!=0»;**  **else**  **{**  **x = 1;**  **y = 0;**  **}**  }  else  {  a = b;  cout<<a;  } | Б) ошибочная запись  if (a<=b)  {  **if (x!=0)**  **cout<<» x!=0»;**  **else**  **{ // начало else**  **x = 1;**  **y = 0;**  }  else  {  **} // конец else**  a = b;  cout<<a;  } |

В приведенном примере один из операторов (выделен полужирным шрифтом) вложен в другой. Запись Б) является ошибочной в силу того, что блок else внутреннего оператора условия частично пересекается с обоими блоками if и else внешнего оператора.

**Пример 2.** Пользователь вводит с клавиатуры три целых числа a, b, c. Необходимо вывести на экран наибольшее из этих чисел.

Решение. Один из возможных алгоритмов решения этой задачи показан на следующей блок-схеме.

(a > b)?

true (+)

false (-)

начало

ввод a, b, c

(b > c)?

(a > c)?

вывод c

вывод b

вывод c

вывод a

true (+)

true (+)

false (-)

false (-)

конец

Схема может быть реализована программно с помощью вложенных операторов if-else

 // – Выбор наибольшего из 3 чисел –

#pragma hdrstop

#include <iostream.h>

#include <conio.h>

#pragma argsused

int main (int argc, char\* argv[])

{

float a, b, c; // объявляем три переменные

cout<< «Vvedite a –»; // вводим значения a, b, c

cin>>a;

cout<< «Vvedite b –»;

cin>>b;

cout<< «Vvedite c –»;

cin>>c;

if (a>b) // если a > b

if (a>c) // и если a > c

cout<<» max = «<<a; // то выводим на экран a

else // иначе, т.е. если a <= с

cout<<» max = «<<c; // выводим с

else // иначе, т.е. если a <= b

if (b>c) // и если b > c

cout<<» max = «<<b; // то выводим b

else // иначе, т.е. если b <= а

cout<<» max = «<<c; // выводим с

getch(); // задержка до нажатия любой клавиши

return 0;

}

 // –

Разберите блок-схему этого алгоритма и его программную реализацию. Модифицируйте алгоритм и программный код для поиска наименьшего из трех чисел.

**4. Составные логические выражения**

Условие в операторе if-else может быть выражено не только в виде простого сравнения двух числовых значений. Например, весьма распространены двойные условия, которые в математике записываются в виде «a < b < c». Запись означает, что значение b лежит в диапазоне между значениями a и c. В программе такие условия должны быть переформулированы с использованием простых операций сравнения и логических операций «И», «ИЛИ», «НЕ»

|  |  |  |
| --- | --- | --- |
| *логическая операция* | *знак в С++* | *наименование знака* |
| И | && | двойной амперсанд |
| ИЛИ | || | двойная вертикальная черта |
| НЕ | ~ | тильда |

В частности, выражение «a < b < c» сформулируем как «a меньше b, и b меньше c». На С++ это будет записано как (a<b)&&(b<c). В тексте программы соответствующий оператор будет иметь вид

if((a<b)&&(b<c))

{

… … …;

}

**Пример 3.** На пустой шахматной доске в позиции (n, m) находится белая пешка, в позиции (k, l) – черный слон. Здесь первая координата есть номер столбца шахматной доски, вторая – номер строки (обе изменяются в диапазоне от 1 до 8). Оцените текущую ситуацию в соответствии с тремя вариантами

* пешка находится под ударом,
* слон находится под ударом,
* слон и пешка в «безопасности».

Решение. Примем во внимание, что пешка может атаковать две ближайшие к себе позиции по диагонали вперед, а слон атакует обе свои диагонали полностью. Отсюда могут быть сформулированы условия

* «((k = n+1) ИЛИ (k = n‑1)) И (l = m+1)» – атака пешки на слона,
* «(k+l = n+m) ИЛИ (k-l = n-m)» – атака слона на пешку,
* иначе фигуры в безопасности.

 // – Шахматная композиция –

#pragma hdrstop

#include <iostream.h>

#include <conio.h>

int main (int argc, char\* argv[])

{

int n, m, k, l;

cout<<«Koordinaty beloi peshki:«<<endl;

cout<<» vvedite n –»;

cin>>n;

cout<<» vvedite m –»;

cin>>m;

cout<<«Koordinaty chernogo slona:«<<endl;

cout<<» vvedite k –»;

cin>>k;

cout<<» vvedite l –»;

cin>>l;

if(((k==n+1)||(k==n‑1))&&(l==m+1)) // атакован слон

cout<<endl<< «Atakovan chernyi slon!»;

else

if((k+l==n+m)||(k-l==n-m)) // атакована пешка

cout<<endl<< «Atakovana belaya peshka!»;

else // нет атаки

cout<<endl<< «Figury v bezopasnosti»;

getch();

return 0;

}

 // –