Автор: Андрей
27.06.2009 21:52 - Обновлено 13.09.2010 21:22

Предисловие

Благодаря использованию инструкций for, while и do вы можете реализовывать циклы, в которых набор некоторые инструкций, называемый *телом цикла* выполняется многократно, если выполнено некое условие. В общем случае инструкции циклов имеют следующий вид:

for(*инициализирующая_инструкция*$_{opt}$;*условие*$_{opt}$;*выражение*$_{opt}$) *инструкция*
while(*условие*) *инструкция*
do *инструкция* while(*выражение*)

Напомню, что индекс opt ("optional") означает, что данный фрагмент является необязательным и в зависимости от конкретной ситуации может быть опущен.

for-инструкция

for-инструкция предназначена для реализации регулярных циклов. При ее использовании инициализация переменной цикла, условие завершения и модификация переменной могут быть произведены в пределах одной строки - внутри скобок, следующих сразу за ключевым словом for. Возможность собрать все элементы, обслуживающие цикл в одном месте способствует созданию ясной и логичной структуры кода и облегчает его читабельность.

Рассмотрим теперь подробнее каждый из этих элементов.

Инициализирующая инструкция задает начальное значение переменной цикла. Если эта переменная была объявлена выше инструкции for, то ей присваивается константное выражение, значение другой переменной или результат вычисления некоторого выражения. Если же переменная цикла не была объявлена, то инициализирующая инструкция также включает объявление. В этом случае область видимости переменной будет распространяться до конца for-инструкции. Например, если в своей программе вы несколько раз используете for для перебора массивов, то может оказаться логичным в каждом из них в качестве счетчика цикла использовать переменные с одним и тем же именем, например i, каждый раз заново ее инициализируя строкой int i = 0;. При этом никакой ошибки не произойдет, т.к. каждый цикл будет знать о своей собственной переменной. Если вам не нужно инициализировать переменную - например, она была не только объявлена, но и инициализирована вне цикла - то инициализирующая инструкция может быть опущена. Точка с запятой, однако, должна остаться - в for-инструкции их

всегда должно быть две.

После первой точки с запятой следует *условие*, при выполнении которого цикл будет исполняться. Если в начале очередной итерации
условие
примет значение false, цикл завершится.
Условие
также может быть опущено (но вторая точка с запятой, означающая конец условия, должна присутствовать). При этом тело цикла будет выполняться "вечно", либо же до тех пор, пока не будет явно прервано другими средствами языка, такими как break, return или другими.

Выражение модифицирует переменную цикла. Чаще всего это делается с помощью оператора инкремента ++ (увеличение на единицу). Если *выражение* опущено, то переменную цикла нужно изменять где-то внутри тела цикла.

Типичными примерами цикла, реализуемого с помощью for-инструкции, являются

```
for(int i = 0; i
for(int j = i; j
if (a[i][j] != a[j][i])
{
//Матрица не симметрична
}
```

в котором пробегается некоторый диапазон значений (например, от нуля до числа, равного размеру массива); а также

```
for(;;)
{
//...
}
```

в котором тело цикла выполняется "вечно" (а точнее, прерывается конструкциями внутри тела цикла).

while-инструкция

Если отсутствует очевидная переменная цикла, или же естественным местом для ее модификации является середина цикла, то вместо инструкции for полезно использовать while. Если для циклов for в условиях характерно отношение меньше-больше, то while как правило проверяет переменные или выражения на равенство - друг другу, какой-то константе или (в частности) нулю (false). Например, в середине тела цикла при определенных условиях некоторая логическая переменная принимает значение true или false, происходит чтение из стандартного потока ввода (если в потоке ничего нет, оператор >> вернет ноль) или же ищется конец строки (С-строки ограничены нулем) - все это типичные примеры применения цикла while.

do-инструкция

Цикл do во многом похож на цикл while, за одним исключением - в случае использования инструкции do тело цикла обязательно выполняется хотя бы один раз,

вне зависимости от результата проверки условия. По логике вещей это должно означать, что эту инструкцию следует применять в тех случаях, когда при первой итерации цикла условие гарантированно выполняется, однако на самом деле может оказаться очень трудно гарантировать, что это и впрямь каждый раз будет так. И если вопреки вашим ожиданиям условие при первом прохождении не будет выполнено, то при работе программы возникнет ошибка, могущая иметь очень неприятные последствия. Я рекомендую вам при реализации циклов ограничиться инструкциями for и while, тем более, что их вполне хватает для нужд программиста.

Отличия в старых версиях

В Borland C++ 3.1 и других старых реализациях C++ область видимости переменных, объявляемых в инструкции for, распространялась до конца области видимости, в которую входила for. Соответственно, если в одной области видимости несколько циклов объявляли переменную с одним и тем же именем, это являлось ошибкой; при этом, однако, можно было после выхода из цикла использовать последнее значение этой переменной. Например, возможен был такой код:

```
    void f(char[] s, int size, char c)
{
for (int i = 0; i                if (i != size) cout << "символ найден в " << i <<"-й позиции";
}
```

Аналогичного результата в современных версиях языка можно добиться просто поместив объявление переменной перед циклом.