ОПЕРАТОРЫ С++

Программа на языке С++ состоит из последовательности **операторов,** каждый из которых определяет законченное описание некоторого действия и заканчивается точкой с запятой. Все операторы можно разделить на четыре группы: операторы следования, ветвления, цикла и операторы передачи управления.

Операторы следования

Операторы следования выполняются компилятором в естественном порядке: начиная с первого до последнего. К операторам следования относятся: *оператор-выражение* и *составной оператор.*
Любое выражение, завершающееся точкой с запятой, рассматривается как оператор, выполнение которого заключается в вычислении значения выражения или выполнении законченного действия. Например:

```
++i;                        // оператор инкремента
x+= y;                      // оператор сложения с присваиванием
f(a, b);                    // вызов функции
x = max(a, b)+a*b;          // вычисление сложного выражения
```

Частным случаем оператора выражения является *пустой оператор* «;». Он используется, когда по синтаксису оператор требуется, а по смыслу — нет. В этом случае лишний символ «;» является пустым оператором и вполне допустим, хотя и не всегда безопасен. Например, случайный символ «;» после условия оператора while или if может совершенно поменять работу этого оператора.
Составной оператор или *блок* представляет собой последовательность операторов, заключенных в фигурные скобки. Блок обладает собственной *областью видимости:* объявленные внутри блока имена доступны только внутри данного блока или блоков, вложенных в него. Составные операторы применяются в случае, когда правила языка предусматривают наличие только одного оператора, а логика программы требует нескольких операторов. Например, тело цикла while должно состоять только из одного оператора. Если заключить несколько операторов в фигурные скобки, то получится блок, который будет рассматриваться компилятором как единый оператор.

Операторы ветвления

Операторы ветвления позволяют изменить порядок выполнения операторов в программе. К операторам ветвления относятся условный оператор if и оператор выбора switch.

Условный оператор if используется для разветвления процесса обработки данных на два направления. Он может иметь одну из форм: *сокращенную* или *полную.*

Формат сокращенного оператора if:

if (B)
s;

где *B* — логическое или арифметическое выражение, истинность которого проверяется; *S* — один оператор: простой или составной.

При выполнении сокращенной формы оператора if сначала вычисляется выражение *B,* затем проводится анализ его результата: если *B* истинно, то выполняется оператор *S;* если *B* ложно, то оператор *S* пропускается. Таким образом, с помощью сокращенной формы оператора if можно либо выполнить оператор *S,* либо пропустить его.

Формат полного оператора if:

if (B)
S1;
else S2;

где *B* — логическое или арифметическое выражение, истинность которого проверяется; SI, S2 — один оператор, простой или составной.

При выполнении полной формы оператора if сначала вычисляется выражение *B,* затем анализируется его результат: если *B* истинно, то выполняется оператор S1, а оператор *S2* пропускается; если *B* ложно, то выполняется оператор *S2,* а S1 — пропускается. Таким образом, с помощью полной формы оператора if можно выбрать одно из альтернативных действий процесса обработки данных.

Рассмотрим несколько примеров записи условного оператора if.

```
if (a > 0)              // Сокращенная форма с простым оператором x = y;
if (++i)                // Сокращенная форма с составным оператором
{
x = y;
y = 2 * z;
}
if (a > 0 || b < 0)   // Полная форма с простым оператором
x = y;
else x = z;
if (i+j-1)            // Полная форма с составными операторами
{
```

```
x = 0;
y = 1;
} else {
x = 1;
y = 0;
}
```

Операторы S1 и S2 могут также являться операторами if. Такие операторы называют вложенными. При этом ключевое слово else связывается с ближайшим предыдущим словом if, которое еще не связано ни с одним else. Рассмотрим несколько примеров алгоритмов с использованием вложенных условных операторов.

Оператор выбора switch предназначен для разветвления процесса вычислений на несколько направлений. Формат оператора:

```
switch ( <выражение> )
{
case <константное_выражение_1>: [«эператор 1>]
case <константное_выражение_2>: [<оператор 2>]
case <константное_выражение_п>: [«эператор п>] [default: <оператор> ]
}
```

Выражение, стоящее за ключевым словом switch, должно иметь арифметический тип или тип указатель. Все константные выражения должны иметь разные значения, но совпадать с типом выражения, стоящим после switch, или приводиться к нему. Ключевое слово case и расположенное после него константное выражение называют также меткой case.
Выполнение оператора начинается с вычисления выражения, расположенного за ключевым словом switch. Полученный результат сравнивается с меткой case. Если результат выражения соответствует метке case, то выполняется оператор, стоящий после этой метки. Затем последовательно выполняются все операторы до конца оператора switch, если только их выполнение не будет прервано с помощью оператора передачи управления break (см. пример). При использовании оператора break происходит выход из switch, и управление передается следующему за switch оператору. Если же совпадения выражения ни с одной меткой case не произошло, то выполняется оператор, стоящий после слова default, а при его отсутствии управление передается следующему за switch оператору.

Упражнения

Дан порядковый номер месяца, вывести на экран его название.
2. Дан порядковый номер дня месяца, вывести на экран количество дней, оставшихся до конца месяца.
3. Дан номер масти m ($1 < m < 4$), определить название масти. Масти нумеруются: «пики» — 1, «трефы» — 2, «бубны» — 3, «червы» — 4.
4. Дан номер карты $к$ ($6 < к < 14$), определить достоинство карты. Достоинства определяются по следующему правилу: «туз» —14, «король» —13, «дама» — 12, «валет» — 11, «десятка» — 10, ..., «шестерка» — 6.
5. Дан номер масти т(1<пг<4)и номер достоинства карты $к$ ($6 < к < < 14$). Определить полное название соответствующей карты в виде «дама пик», «шестерка бубен» и т.д.
6. С 1 января 1990 г. по некоторый день прошло m месяцев, определить название текущего месяца.
7. Дано расписание приемных часов врача. Вывести на экран приемные часы врача в заданный день недели (расписание придумать самостоятельно).
8. Проведен тест, оцениваемый в целочисленный баллах от нуля до 100. Вывести на экран оценку тестируемого в зависимости от набранного количества баллов: от 90 до 100 — «отлично», от 70 до 89 — «хорошо», от 50 до 69 — «удовлетворительно», менее 50 — «неудовлетворительно».
9. Даны два числа и арифметическая операция. Вывести на экран результат этой операции.
10. Дан год. Вывести на экран название животного, символизирующего этот год по восточному календарю.
11. Дан возраст человека мужского пола в годах. Вывести на экран возрастную категорию: до года — «младенец», от года до 11 лет — «ребенок», от 12 до 15 лет — «подросток», от 16 до 25 лет — «юноша», от 26 до 70 лет — «мужчина», более 70 лет — «пожилой человек».
12. Дан пол человека: м — мужчина, ж — женщина. Вывести на экран возможные мужские и женские имена в зависимости от введенного пола.
13. Дан признак транспортного средства: а — автомобиль, в — велосипед, м — мотоцикл, с — самолет, п — поезд. Вывести на экран максимальную скорость транспортного средства в зависимости от введенного признака.
14. Дан номер телевизионного канала (от 1 до 5). Вывести на экран наиболее популярные программы заданного канала.
15. Дан признак геометрической фигуры на плоскости: к — круг, п — прямоугольник, т — треугольник. Вывести на экран периметр и площадь заданной фигуры (данные, необходимые для расчетов, запросить у пользователя).