

Тема 1.1 Основные операторы языка C

Простой и составной оператор

Операторы описывают те алгоритмические действия, которые должны выполняться в программе. Существует два основных вида операторов: простые и составные.

Простым оператором является оператор, который не содержит в себе других операторов. К ним можно отнести оператор присваивания, оператор процедуры и оператор перехода.

Составным оператором называется конструкция языка программирования, которая состоит из нескольких команд (операторов), но участвует в программе в качестве единого оператора. Сюда можно отнести следующие операторы: `if`, `switch`, `while`, `do while`, `for`. В языке C составной оператор ограничивается фигурными скобками.

1 Операторы принятия решений

В языке программирования C используется несколько конструкций для принятия решений:

1. оператор `if`;
2. оператор `switch`;
3. условный оператор.

1.1 Оператор `if`

Общая форма записи оператора `if`:

```
if (expression)
program statement;
```

В операторе `if` используется результат вычисления условия, заключенного в круглые скобки, на основе которого принимается решение. Результат вычисления условия `expression` может быть арифметическим или логическим. Если результат выполнения условия `expression` будет истинным, то возможно выполнить несколько утверждений типа `program statement`. Для этого следует использовать фигурные скобки, например:

```
if (expression)
{
program1 statement1;
program2 statement2;
...
}
```

Так же можно использовать конструкцию `if-else`:

```
if (expression)
program1 statement1;
else
program2 statement2;
```

Если выполняется условие `expression`, то будет выполняться фрагмент программы `program1 statement1`, в противном случае будет выполняться `program2 statement2`.

1.2 Оператор `switch`

Последовательность операторов `if-else`, когда значения переменных последовательно сравниваются с различными значениями, часто используется, для этого был разработан специальный оператор `switch`.

Формат оператора `switch`:

```
switch (expression)
{
    case value1:
        program statement
        program statement
        ...
        break;
    case value2:
        program statement
        program statement
        ...
        break;
    ...
    case valuen:
        program statement
        program statement
        ...
        break;
    default:
        program statement
        program statement
        ...
        break;
}
```

Выражение `expression`, заключенное в круглые скобки, последовательно сравнивается со значениями `value1`, `value2`, ..., `valuen`, которые должны быть простыми константами или константными выражениями. В том случае, когда одно из этих значений равно значению выражения `expression`, выполняются те утверждения, которые следуют за данным значением.

Утверждение `break` сигнализирует об окончании выполнения утверждений и приводит к выходу из оператора `switch`.

Специальный дополнительный вариант `default` будет выполнен в том случае, когда не будет найдено ни одного совпадения.

1.3 Условный оператор

Оператор условия является тернарным, это означает, что он может иметь три операнда.

Общий формат записи:

```
условие ? выражение_1 : выражение_2
```

Если в результате вычисления условия будет получено значение TRUE, то выполняется выражение_1, и результатом выполнения оператора условия будет значение, полученное при вычислении этого выражения. Если в результате вычисления условия будет получено значение FALSE, то выполняется выражение_2, и результатом выполнения оператора условия будет значение, полученное при вычислении выражения_2.

2 Организация цикла в программе

Циклом называется блок кода, который для решения задачи требуется повторить несколько раз. Каждый цикл состоит из: блока проверки условия повторения цикла; тела цикла. Цикл выполняется до тех пор, пока блок проверки условия возвращает истинное значение. Тело цикла содержит последовательность операций, которая выполняется в случае истинного условия повторения цикла. После выполнения последней операции тела цикла снова выполняется операция проверки условия повторения цикла. Если это условие не выполняется, то будет выполнена операция, стоящая непосредственно после цикла в коде программы.

В языке C существуют следующие виды циклов:

1. while – цикл с предусловием;
2. do...while – цикл с постусловием;
3. for – параметрический цикл (цикл с заданным числом повторений).

2.1 Цикл с предусловием while

Общая форма записи:

```
while (выражение) {  
    Блок операций;  
}
```

Если выражение истинно, то выполняется блок операций, заключенный в фигурные скобки, затем выражение проверяется снова. Последовательность действий, состоящая из проверки и выполнения блока операций, повторяется до тех пор, пока выражение не станет ложным. При этом происходит выход из цикла и производится выполнение операции, стоящей после оператора цикла.

При построении цикла while в него необходимо включить конструкции, изменяющие величину проверяемого выражения так, чтобы в конце концов оно стало ложным, иначе выполнение цикла будет осуществляться бесконечно.

2.2 Цикл с постусловием do...while

Общая форма записи:

```
do {  
    блок операций;  
} while (выражение);
```

Цикл do...while – это цикл с постусловием, где истинность выражения проверяется после выполнения всех операций, включенных в блок, ограниченный фигурными скобками. Тело цикла выполняется до тех пор, пока выраже-

ние не станет ложным, то есть тело цикла с постусловием выполнится хотя бы один раз.

Использовать данный цикл лучше в тех случаях, когда должна быть выполнена хотя бы одна итерация, либо когда инициализация объектов, участвующих в проверке условия, происходит внутри тела цикла.

2.3 Параметрический цикл for

Общая форма записи:

```
for (инициализация параметра; проверка условия окончания; коррекция параметра) {  
    блок операций;  
}
```

For — параметрический цикл (цикл с фиксированным числом повторений). Для организации такого цикла необходимо осуществить три операции:

1. инициализация параметра - присваивание параметру цикла начального значения;
2. проверка условия окончания - сравнение величины параметра с некоторым граничным значением;
3. коррекция параметра - изменение значения параметра при каждом прохождении тела цикла.

Эти три операции записываются в скобках и разделяются точкой с запятой (;). Как правило, параметром цикла является целочисленная переменная. Инициализация параметра осуществляется только один раз — когда цикл for начинает выполняться. Проверка условия окончания осуществляется перед каждым возможным выполнением тела цикла. Когда выражение становится ложным (равным нулю), цикл завершается. Коррекция параметра осуществляется в конце каждого выполнения тела цикла. Параметр может как увеличиваться, так и уменьшаться.