СТРУКТУРА ВЫБОРА

4 лекция

Дисциплина «Программирование микроконтроллеров на языке С++»

Управляющие структуры

Все программы могут быть написаны с использованием всего трех управляющих структур:

- Структура следования
- Структура выбора
- Структура повторения

Структура следования встроена в C++. Пока не указано иное, компьютер выполняет операторы один за другим в той последовательности , в которой они записаны.

Типы структур выбора альтернатив

Формат условного оператора	Тип структуры выбора	Комментарий
If (Выражение)Оператор;	Структура с единственным выбором	Если Выражение истинно, то выполняется Оператор, если выражение ложно, то выполняется следующий за if оператор
if(Выражение)Оператор1; else Оператор2;	Структура с двойным выбором	Если Выражение истинно, то выполняется Оператор1, если ложно – Оператор2
switch	Структура с множественным выбором	Реализуется многоальтернативное ветвление

Операторы сравнения

Обозначение	Значение
Операции отношения	
>	больше
>=	больше или равно
<	меньше
<=	меньше или равно
Операции проверки на равенство	
= =	равно (сравнение на равенство)
!=	не равно

Логические операции

Битовые логические операции		Логические операции	
&	and, и	&&	and, и
|	or, или	||	or, или
~	отрицание not, не	!	отрицание not, не
^	исключающее или, or		

// Пример применения операций сравнения

```
#include <iostream>

int main()
{
int x ;
cin >> x;                          // ввести переменную x
if(x>5)                            // если значение переменной x больше 5.
cout << "x>5" << endl;             // вывести об этом сообщение

if(x==5)                           // если x==5
cout << "x==5" << endl;            // вывести сообщение о равенстве
else cout << "x!=5" << endl;       // иначе вывести сообщение о неравенстве
return 0;
}
```

Условная операция присваивания. Операторы if ... else

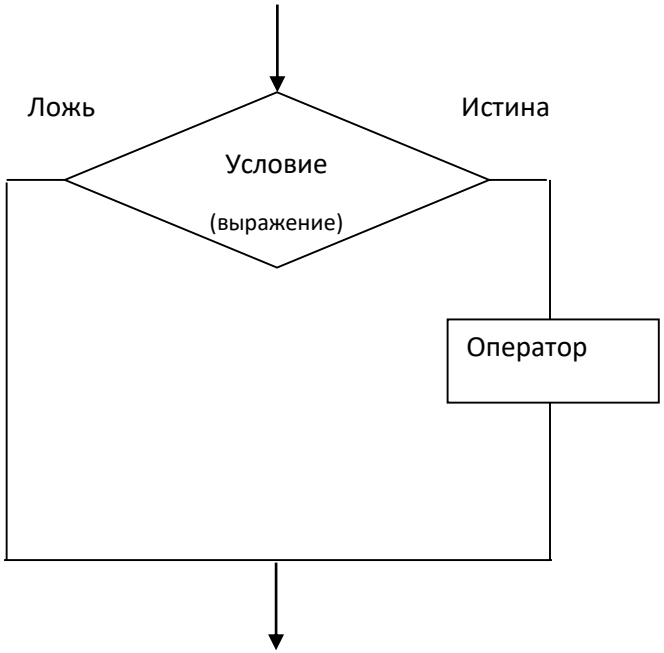


Рисунок 1 – Блок-схема неполной формы условного оператора

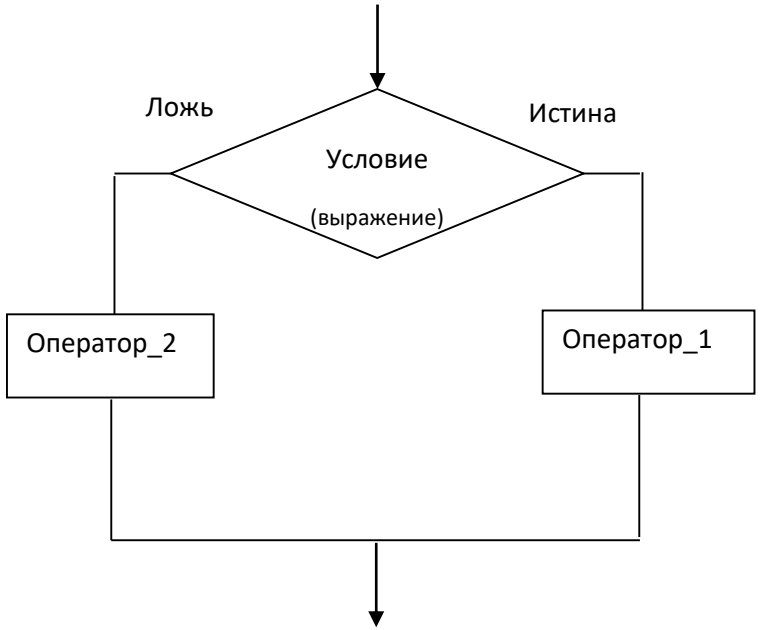


Рисунок 2 – Блок-схема полной формы условного оператора

Структура с единственным выбором

Оператор **if** используется для условного выполнения фрагмента кода. Простейший его вид:

if (условное выражение)
действие_если_истина;

```
if (a != 0)
x = 1/a;
```

Пояснения:

- условное выражение всегда заключается в круглые скобки;
- если оно истинно (true), выполняется действие, указанное после if;
- если оно ложно (false), действие пропускается, и программа продолжает выполнение дальше.

Структура с двойным выбором

Операторы **if - else** используется, когда необходимо выполнить разные действия для истинного и ложного случаев. Простейший его вид:

```
if (условное выражение)
  действие_если_истина;
else
  действие_если_ложь;
```

```
if (a != 0)
x = 1/a;
else
cout<<"Решение отсутствует";
```

Пояснения:

- условное выражение заключается в круглые скобки;
- если условие истинно (true), выполняется действие после if;
- если условие ложно (false), выполняется действие после else;
- выполняется только один из двух вариантов.

Составной оператор

Составной оператор – последовательность операторов, заключенная в фигурные скобки:

```
{
оператор 1;
оператор 2;
оператор 3;
}
```

Выполнение составного оператора заключается в последовательном выполнении составляющих его операторов.

Важно:

- внутри блока каждый оператор заканчивается точкой с запятой
- после закрывающей фигурной скобки } точка с запятой не ставится

Составной оператор

```
If (условное выражение)
{
действие1_если_истина;
действие2_если_истина;
действие3_если_истина;
}
else
{
действие1_если_ложь;
действие2_если_ложь;
действие3_если_ложь;
}
```

```
cout << "Введите два числа : ";
cin >> a >> b;

   if (a > b)
{
      int diff = a - b;
      int sum = a + b;
      double ratio = (double)a / b;
}
   else
{
      int diff = b - a;
      int product = a * b;
      int square = b * b;
}
```

- между закрывающей скобкой **}** и ключевым словом **else** точка с запятой не ставится;
- внутри каждого блока можно размещать любое количество операторов - они выполняются последовательно;
- рекомендуется использовать вид записи с фигурными скобками, даже если в блоке находится только один оператор.

Вложенные операторы

Иногда требуется проверить несколько условий, одно внутри другого. Для этого используются вложенные условные операторы.

```
if (условие1)
{
   if (условие2)
   {
      действия_если_оба_истинны;
   }
   else
   {
      действия_если_условие1_истинно, а условие2_ложно;
   }
}
else
{
   действия_если_условие1_ложно;
}
```

```
cout << "Введите два числа : ";
cin >> a >> b;

    if (a > 0)
    {
        if (b > 0)
            cout << "Обе переменные положительные" << endl;
        else
            cout << "a положительное, b неположительное" << endl;
    }
    else
    {
        cout << "a неположительное" << endl;
    }
```

➢1	Если a = 5, b = -3 → вывод:	a положительное, b неположительное
➢2	Если a = -2, b = -7 → вывод:	a неположительное
➢3	Если a = 3, b = 4 → вывод:	Обе переменные положительные

Составной оператор

Пояснения

- при использовании вложенных операторов **if** текущий **else** всегда относится к последнему **if**, с которым еще не сопоставлен ни один **else**.
- рекомендуется всегда использовать фигурные скобки, даже если в блоке всего один оператор;
- если скобки опустить, может возникнуть так называемая проблема «висячего else»: оператор **else** может быть сопоставлен не с тем **if**;
- вложенные **if** позволяют проверять условия последовательно, уточняя ситуацию;
- каждая пара **if–else** работает только в своей области - важно правильно расставлять фигурные скобки.

Структура с множественным выбором

Оператор **switch** используется, если необходимо сравнить некоторую целочисленную переменную (или выражение, принимающее целочисленное значение) с несколькими значениями. Его синтаксис:

```
switch (выражение)
{
case константа1:
операторы1;
break;
case константа2:
операторы2;
break;
…
default:
операторы;
}
```

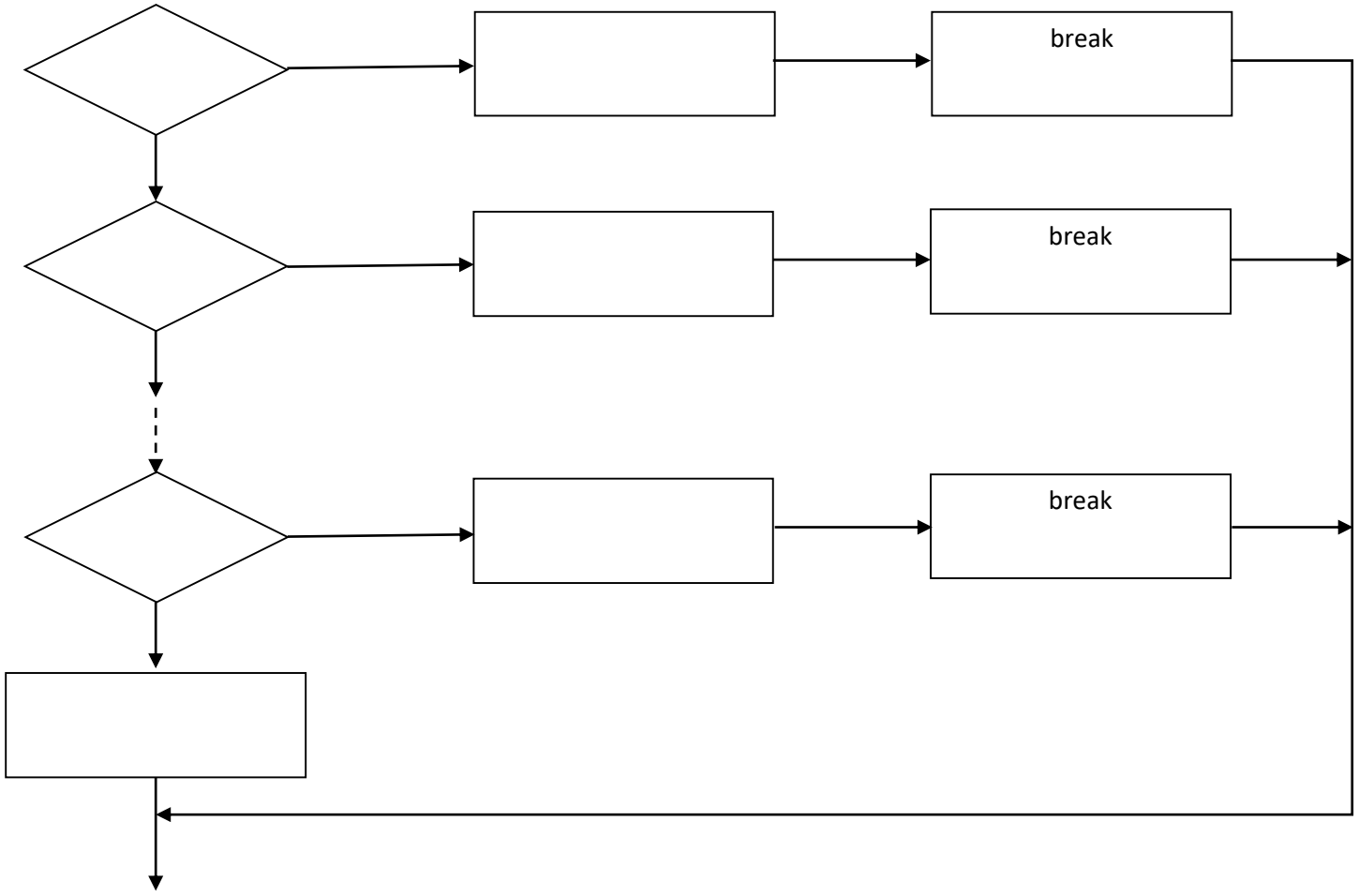
```
cout << "Введите целое число ";
cin >> i;
  switch(i)
{
    case 1:
    cout<<" Число равное 1 ";
    break;
    case 2:
    cout<<" Число равное 2 ";
    break;
    case 3:
    cout<<" Число равное 3 ";
    break;
    default:
    cout<<" Завершение работы "
}
```

Схема выполнения оператора switch следующая:

- вычисляется значение выражения в круглых скобках;
- вычисленные значения последовательно сравниваются с константными выражениями, следующими за ключевыми словами **case**;
- если одно из константных выражений совпадает со значением выражения, то управление передается на оператор, помеченный соответствующим ключевым словом **case**;
- после передачи управления по какой-либо одной из меток дальнейшие операторы выполняются подряд. Поэтому, если необходимо выполнить только часть из них, нужно позаботиться о выходе из **switch**. Это делается с помощью оператора break, который осуществляет выход из тела оператора **switch**;
- если ни одно из константных выражений не равно выражению, то управление передается на оператор, помеченный ключевым словом **default**, а в случае его отсутствия управление передается на следующий после **switch** оператор.

Оператор выбора switch

Блок –схема структуры множественного выбор



Выводы

Структуры выбора в языке C++ являются важнейшими инструментами управления ходом выполнения программы. Они позволяют задавать различные варианты действий в зависимости от условий.

- **Оператор if** – основа всех конструкций выбора. С его помощью программа выполняет блок операторов только при истинности заданного логического условия.
- **Оператор if–else** расширяет возможности выбора, позволяя задавать альтернативные действия для случая, когда условие ложно.
- **Вложенные операторы if** применяются для последовательной проверки нескольких условий.
- **Оператор switch** удобен, когда нужно выбирать из множества вариантов, основанных на конкретных значениях выражения.

Таким образом, конструкции выбора обеспечивают гибкость и логичность программ, позволяют разрабатывать алгоритмы, учитывающие различные варианты развития событий.