

Условный оператор



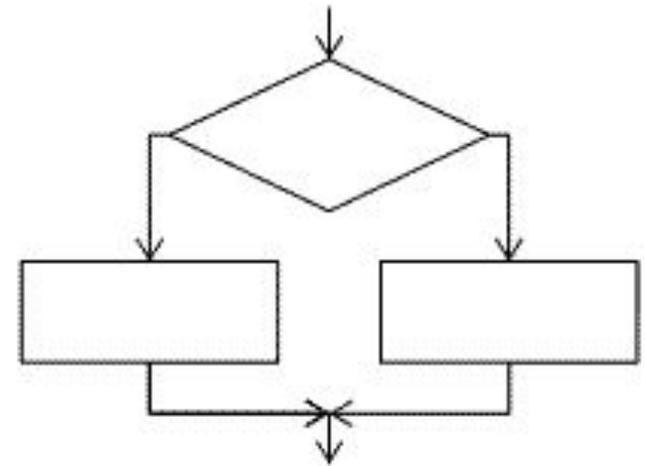
Пойдет направо – песнь
заводит
Налево – сказку говорит...
А. С. Пушкин

Условный оператор

Условный оператор *if* используется для разветвления процесса вычислений на два направления.

Формат оператора:

```
if (выражение) оператор_1;  
[else оператор_2;]
```



Ветвление

Структурная схема оператора



Условный оператор

- ❖ Сначала вычисляется выражение, которое может иметь арифметический тип или тип указателя.
 - ❖ Если оно не равно нулю, выполняется первый оператор, иначе — второй. После этого управление передается на оператор, следующий за условным.
-



Условный оператор

- ❖ Одна из ветвей может отсутствовать.
 - ❖ Если в какой-либо ветви требуется выполнить несколько операторов, их необходимо заключить в блок. Блок может содержать любые операторы, в том числе описания и другие условные операторы.
-



Условный оператор

Знаки сравнения:

==	равно
!=	не равно
<	меньше, чем
>	больше, чем
<=	меньше или равно
>=	больше или равно



Условный оператор

Программа:

```
//Площадь треугольника
#include <iostream.h>
#include <math.h>
main()
{float a,b,c,P,S;
Cout<<"\na="; cin>>a;
Cout<<"\nb="; cin>>b;
Cout<<"\nc="; cin>>c;
If (a>0 && b>0 && c>0 && a+b>c && a+c>b && b+c>a)
{
P=(a+b+c)/2;
S=sqrt(P*(P-a)*(P-b)*(P-c));
Cout<<"\nПлощадь треугольника="<<S;
}
else cout (<<"\n Неверные исходные данные.);
```



Условный оператор

Туристы вышли из леса на шоссе неподалеку от километрового столба с отметкой А км и решили пойти на ближайшую автобусную остановку. Посмотрев на план местности, руководитель группы сказал, что автобусные остановки расположены на километре В и на километре С. Куда следует пойти туристам?



Условный оператор

- ❖ Даны три действительных числа a , b , c . Найти наибольшее из них.
-



Оператор

**множественного
выбора**



SWITCH

Использование оператора SWITCH

Инструкция множественного выбора switch позволяет выполнять различные части программы в зависимости от того, какое значение будет иметь некоторая целочисленная переменная (её называют «переменной-переключателем», а «switch» с английского переводится как раз как «переключатель»).





Использование оператора SWITCH

Оператор switch состоит из двух частей.

- ❖ Первая часть оператора switch представляет собой условие, которое появляется после ключевого слова switch.
 - ❖ Вторая часть представляет собой возможные варианты соответствия. Когда программа встречает оператор switch, она сначала исследует условие, а затем пытается найти среди возможных вариантов тот, который соответствует условию.
-

Использование оператора SWITCH

Схема инструкции такова:

```
switch (переключатель) {  
    case значение1:  
        инструкция1;  
        break;  
    case значение2:  
        инструкция2;  
        break;  
    ...  
    default:  
        инструкция_по_умолчанию;  
}
```



Использование оператора SWITCH

Рассмотрим все элементы оператора:

- переключатель — это целочисленная переменная или выражение дающее целочисленный результат;





Использование оператора SWITCH

□ **значение1, значение2, ...** — это целочисленные литералы, с которыми будет сравниваться значение переключателя.

Если переключатель равен **значениюN**, то программа будет выполняться со строки, следующей за case **значениеN**: и до ближайшего встреченного break, либо до конца блока switch (если break не встретится);

Использование оператора SWITCH

- ❖ Если программа находит соответствие, выполняются указанные операторы.
- ❖ Если же ни один из указанных вариантов не соответствует условию, то выполняется вариант default.

Обратите внимание на использование оператора break в каждом варианте предыдущей программы.



Использование оператора SWITCH

- ❖ Оказывается, если C++ встречает вариант, соответствующий условию оператора switch, то он подразумевает, что все последующие варианты тоже соответствуют условию.



Использование оператора SWITCH

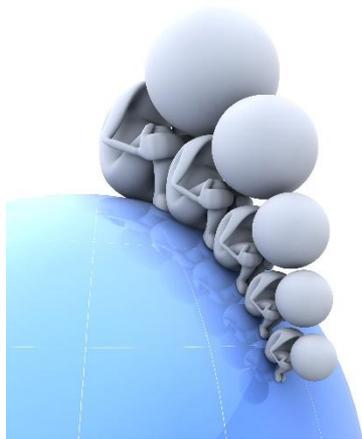
- ❖ Оператор `break` указывает C++ завершить текущий оператор `switch` и продолжить выполнение программы с первого оператора, следующего за оператором `switch`. Если удалить операторы `break` из программы, то программа выведет не только требуемое сообщение, но и сообщение для всех последующих вариантов (потому что если один вариант является истинным, то и все последующие варианты в C++ рассматриваются как истинные).





Использование оператора SWITCH

Такое поведение программы можно использовать во благо. Например, можно разместить несколько подряд идущих меток с разными литералами, для которых будет выполняться один и тот же код:





Использование оператора SWITCH

```
switch(ans) {  
case 'Д':  
case 'д':  
case 'У':  
case 'у':  
    cout << "Продолжаем программу";  
case 'Н':  
case 'н':  
case 'N':  
case 'n':  
    cout << "Останавливаем программу";  
    break;  
default:  
    cout << "Вы ввели неподходящий символ";  
}
```



Использование оператора SWITCH

- **default:** — это метка инструкции после которой будут выполняться в том случае, если выше ни одно из **значенийN** не совпало с переключателем. Метка default — необязательная: можно её не включать в блок switch меток или не выполнять после неё никаких команд;
 - **инструкцияN** — простая или составная инструкция. При этом в случае составной несколько команд не обязательно объединять в блок, можно их просто написать друг за другом разделяя с помощью «;» (и начиная новые строки для удобства).
-



Использование оператора SWITCH

Если мы захотим как-то оповестить пользователя о том, что он ввёл неподходящий символ, то пригодится метка default:

```
switch(ans) {  
    case 'Д':  
        cout << "Продолжаем программу";  
        break;  
    case 'Н':  
        cout << "Останавливаем программу";  
        break;  
    default:  
        cout << "Вы ввели неподходящий символ";  
}
```



Использование оператора SWITCH

Вывод словесного описания оценки, основываясь на текущей оценке ученика

```
#include <iostream.h>
int main()
{
    int grade;
    cout << "Введите оценку" << endl;
    cin >> grade;
    switch (grade)
    { case 5: cout << "Поздравляем, вы получили пять" << endl;
      break;
      case 4: cout << "Хорошо, у вас хорошо" << endl; break;
      case 3: cout << "У вас всего лишь удовлетворительно" <<
      endl; break;
      case 2: cout << "Плохо, у вас двойка" << endl; break;
      default: cout << "Ужасно! Учите лучше!" << endl; break;
    }
}
```



Использование оператора SWITCH

Switch позволяет только сравнивать переключатель с конкретными значениями, но не позволяет для какой-то из веток задать условие в виде целого диапазона значений с использованием операторов сравнения (например, с использованием строгих неравенств и логических операторов «и» или «или»).

В качестве переключателя могут выступать только целочисленные переменные или выражения.
