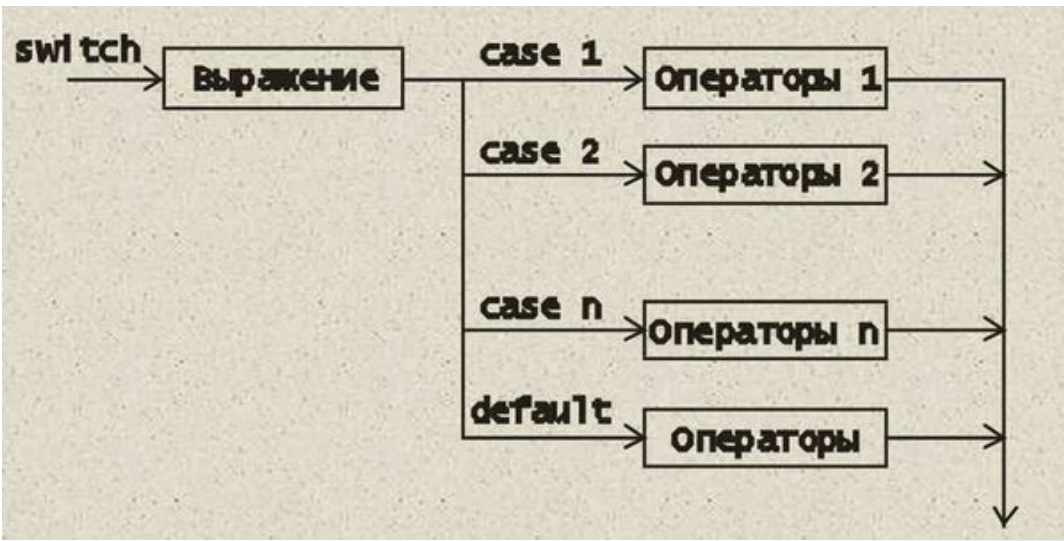


# Оператор выбора



Итак мы рассмотрели оператор с одиночным выбором `if` и оператор с двойным выбором `if else`, но в C++ еще имеется оператор множественного выбора `switch`, который мы сейчас детально рассмотрим.

```
// форма записи оператора множественного выбора  
switch
```

```
switch (/*переменная или выражение*/)
```

```
{
```

```
case /*константное выражение1*/:
```

```
{
```

```
/*группа операторов*/;
```

```
break;
```

```
}
```

```
case /*константное выражение2*/:
```

```
{
```

```
/*группа операторов*/;
```

```
break;
```

```
}
```

```
//. . .
```

```
default:
```

```
{
```

```
/*группа операторов*/;
```

```
}
```

На начальном этапе анализируется выражение или переменная. После чего осуществляется переход к той ветви программы, для которой значение переменной или выражения совпадает с указанным константным выражением. Далее выполняется оператор или группа операторов пока не встретится зарезервированное слово **break** или закрывающая фигурная скобочка. Если значение переменной или выражения не совпадает ни с одним константным выражением, то передается управление ветви программы содержащей зарезервированное слово **default**. После чего выполняется оператор или группа операторов данной ветви.

Варианты:

```
int c=3;
```

```
int a=5;
```

```
float g=0.5;
```

```
char s='c';
```

```
switch (c) //Правильно (целая переменная)
```

```
switch ((a+c)+2) //Правильно (выражение целого типа)
```

```
switch (g) //Неправильно (вещественная переменная)
```

```
switch (s) //Правильно (символьная переменная)
```

Отметим, что в теле оператора **switch** можно использовать вложенные операторы **switch**, при этом в ключевых словах **case** можно использовать одинаковые константные выражения.

```
switch (a)
{
  case 1: b=c; break;
  case 2:
    switch (d)
    {
      case 0: f=8; break;
      case 1: f=5; break;
      case 2: f-=9; break;
    }
  case 3: b-=0; break;
}
```

# Задани

е:

Задание 1: написать программу, которая складывает, вычитает, умножает, делит два числа введенных с клавиатуры (использовать оператор выбора)

Задание 2: Определить номер четверти плоскости, в которой находится точка с координатами X и Y , причем XY != 0. Координаты вводить с клавиатуры.

Задание 3: Составить программу-тест на языке программирования C++: на экране по очереди появляются вопросы (3 любых вопроса), с вариантами ответов. В конце работы программа выдает количество заработанных баллов по результатам ответа. (Пример ввода вопроса с результатами ответов: cout << "1)How many cat lives? \n answers: \n-> 5 lives \n-> 7 lives \n-> 9 lives\n your answer: "; )