Снежко Лев

Источники:

<https://learn.microsoft.com/ru-ru/cpp/c-language/switch-statement-c?view=msvc-170>

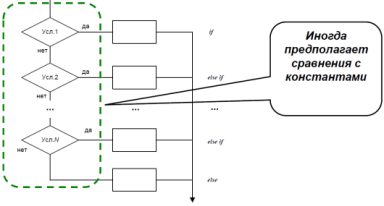
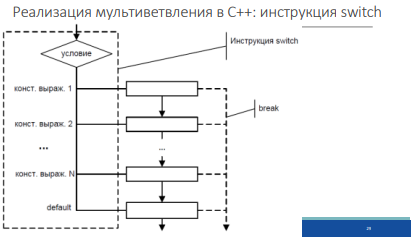
Лекции по С++ Казанцевой О.Г.:

<https://edufpmi.bsu.by/pluginfile.php/7973/mod_resource/content/6/2022-09-15-19-if_statement.pdf>

Code-snippets:

<http://easy-code.ru/lesson/switch-case-cpp>

**Вопрос 18. Оператор выбора switch. Break**

https://learn.microsoft.com/ru-ru/cpp/c-language/switch-statement-c?view=msvc-170

Оператор выбора switch позволяет осуществить выбор выполняемого набора операторов, основываясь на результатах сравнения значения со списком целочисленных выражений. Передает управление одному из нескольких операторов в зависимости от значения условия

**switch (выражение) операторы (until C++17)**

*Выражение* – значение, которое определяет выбор варианта, должно иметь целочисленный тип, тип char или быть переисчлением. В частности, выбор по значению string произвести невозможно.

***switch (выражение)***

***{***

***case метка 1: оператор(ы);***

***case метка 2: оператор(ы);***

***[default: оператор(ы)]***

***}***

***switch ( init-statement(optional); выражение ) statement (since C++17)***

Если используется оператор инициализации, то такой оператор switch эквивалентен:

***{***

***init\_statement***

***switch(выражение) statement;***

***}***

Оператор **break** можно использовать для завершения обработки определенного оператора с меткой в операторе **switch** . Он выполняет ветвление до конца оператора **switch** . Без оператора **break** выполнение программы продолжается до следующего оператора с меткой. Операторы будут выполняться до достижения оператора **break** или конца всего оператора. В некоторых случаях может требоваться именно такое поведение.

Метки **case** и **default** тела оператора **switch** имеют значения только при начальной проверке, которая определяет, с какого места тела оператора начинается выполнение. Операторы **switch** могут быть вложенными. Любые статические переменные инициализируются до выполнения любых операторов **switch**.

switch( c )

{

case 'A':

capital\_a++;

case 'a':

letter\_a++;

default :

total++;

}

В этом примере, если переменная c равна 'A', выполняются все три оператора тела **switch** , так как оператор **break** перед следующим оператором **case** отсутствует. Управления передается первому оператору (capital\_a++;) и продолжается по-порядку во всей остальной части тела. Если переменная c равна 'a', увеличиваются значения переменных letter\_a и total. Если переменная c не равна 'A' или 'a', увеличивается только значение переменной total.

**Примеры:**

***int a=1;***

***switch(a)***

***{***

***case 1:***

***a++;***

***case 2:***

***a++;***

***case 3:***

***a++;***

***}***

***std::cout<<"a= "<<a;***

**Инструкция switch: рекомендации**

➢ Упорядочивайте варианты по вероятности появления значения

➢ Помещайте правильный вариант первым

➢ Не создавайте искусственные переменные, чтобы использовать switch вместо if … else if

➢ Не используйте default для обработки последнего оставшегося варианта

➢ Если ветка default не используется, оставляйте ее для выявления ошибок

➢ Избегайте кода, «проваливающегося» сквозь блоки case. Если он все же применяется, тщательно документируйте его

Определения:

Выражение

Вопросы:

Зачем нужны break в блоках case?

Может ли использовать число типа long в условии switch? А типа short?

Нужно ли заключать операторы в блоке case в фигурные скобки?