

Оператор ветвления (условная конструкция, условный оператор) — оператор, конструкция языка программирования, обеспечивающая выполнение определённой команды (набора команд) только при условии истинности некоторого логического выражения, либо выполнение одной из нескольких команд (наборов команд) в зависимости от значения некоторого выражения.

В C# существует три основные формы условной конструкции:

1. условный оператор (if-else)
2. тернарный оператор (? :)
3. оператор многозначного выбора (переключатель, switch-case).

Условный оператор if реализует выполнение определённых команд при условии, что используемое логическое выражение в условии, принимает значение true.

Если использовалась конструкция if-else, и результатом условия было значение true, то выполнится только тело оператора if, а тело блока else останется не выполненным.

После выполнения оператора if управление передается следующему оператору. Оператор, выполняемый после проверки условия, может быть любого типа, включая другой оператор if, вложенный в оригинальный оператор if. Во вложенных операторах if предложение else принадлежит к последнему оператору if, у которого нет соответствующего else.

Если тело блока if или else состоит из одного выражения, то операторные скобки можно опустить.

Тернарная условная операция (записывается как ? :) - операция, возвращающая свой второй или третий операнд в зависимости от значения логического выражения, заданного первым операндом.

Тернарный оператор [? :], является сокращенной формой конструкции if... else.

Тернарный оператор состоит из следующих операндов:

(условие) ?(блок истинности или то) : (блок иначе) ;

Алгоритм работы тернарного оператора:

(логическое выражение) ? выражение 1 : выражение 2

1. Вычисляется логическое выражение (условие).
2. Если логическое выражение истинно, то вычисляется значение выражения выражение 1 (блока истинности), в противном случае — значение выражения выражение 2 (блока иначе).
3. Вычисленное значение возвращается.

Тернарный оператор обязательно должен возвращать значение, иначе будет ошибка.

· Либо блок истинности и блок иначе должны быть одинакового типа, либо должно существовать неявное преобразование из одного типа в другой.

· Конструкция переключателя switch-case имеет несколько (две или более) ветвей. Переключатель выполняет одну заданную ветвь в зависимости от значения вычисляемого ключевого выражения. Принципиальным отличием этой конструкции от условного оператора является то, что выражение, определяющее выбор исполняемой ветви, допускает использование не логических значений.

Для пустых операторов case разрешено "проваливание" от одного оператора к другому.

В каждом операторе case указывается постоянное значение. Выполняется тело того оператора case, постоянное значение которого, соответствует значению выражения селектора оператора switch.

Если постоянное выражение оператора case не содержит соответствующего значения, выполняется блок default, если таковой имеется. Если блок default отсутствует, происходит выход за пределы оператора switch.

Каждый блок case, как и блок default, в котором содержатся выполняемые операторы, должен завершаться оператором перехода break, или goto case, return или throw

Выполнение сравнения значения выражения селектора с постоянными значениями операторов case с первого оператора и продолжается по списку, обычно до достижения оператора перехода. В этой точке управление передается за пределы оператора switch или переходит к другому оператору case, если операторы перехода или тело оператора case отсутствовали, и так до того оператора case, у которого будет присутствовать тело и оператор перехода. Такая техника называется проваливанием.

Блок default может быть создан в любом месте тела переключателя switch-case. Исключением является тело операторов case.

Использование Convert.To ...() преобразует значение одного базового типа данных к другому базовому типу данных.

Методы класса Convert используются для поддержки преобразования в базовый тип данных и из него в .NET Framework. Поддерживаются следующие базовые типы: Boolean, Char, SByte, Byte, Int16, Int32, Int64, UInt16, UInt32, UInt64, Single, Double, Decimal, DateTime и String.