

№ 27 **Курс:** Процедурное программирование на языке C#
урока:

Средства обучения: Visual Studio 2019 Community Edition

Обзор, цель и назначение урока

Данный урок познакомит вас с новым типом условной конструкции в языке C# - тернарным оператором, как более краткой альтернативой условной конструкции if-else. Вы рассмотрите устройство тернарного оператора, варианты применения тернарного оператора и его работу со ссылочными переменными.

Изучив материал данного занятия, учащийся сможет:

- Понимать, что такое тернарный оператор.
- Уметь применять тернарный оператор на практике.
- Понимать смысл ключевого слова ref и его применения в тернарных операторах.

Содержание урока

1. Устройство тернарного оператора
2. Обратный тернарный оператор
3. Ключевое слово ref и ссылочные переменные

Резюме

- **Тернарный оператор** [? :] — это более краткая альтернатива условной конструкции if-else.
- Тернарный оператор — это просто удобная альтернатива традиционной условной конструкции if-else. If-else всегда может заменить собой тернарный оператор, но в определённых случаях, использование тернарного оператора, делает код короче, красивей и лаконичней.
- В тех случаях, когда логика примитивная, и нет смысла разносить ее на несколько строк, удобно располагать целую условную конструкцию на одной строке. Как вариант – использовать тернарный оператор.
- Тернарный оператор состоит из трёх частей. Первая часть, это условие, вторая часть — это блок истинности, и третья часть — это блок ложности.
- Алгоритм работы тернарного оператора:
(логическое выражение) ? выражение 1 : выражение 2
 1. Вычисляется логическое выражение (условие).

2. Если логическое выражение истинно, то выполняется «выражение 1» (блок истинности), в противном случае – выполняется «выражение 2» (блок ложности).
 3. Вычисленное в п.2 значение возвращается.
- При использовании тернарных операторов, нужно помнить о необходимости преобразования значения выражений к одному типу. Результаты выражений из правой части (блок истинности и блок ложности) должны быть преобразованы к типу, указанному в объявлении переменной слева от знака присвоения.
 - Когда мы используем операторы приведения типов перед тернарным оператором, нам нужно весь тернарный оператор заключать в круглые скобки.
 - Тернарный оператор можно использовать в левой части выражения для выборочного присвоения значения переменным в одном из блоков – истинности или ложности, в зависимости от условия:
(a > b ? ref max : ref min) = 33; // ЕСЛИ: (a > b) ТО: max = 33; ИНАЧЕ: min = 33;
При этом переменные в блоках должны быть проинициализированы заранее и перед ними стоять ключевое слово ref.
 - ref - это сокращение от английского слова reference, которое переводится как ссылка. Поэтому такие переменные, помеченные ключевым словом ref, называют – ссылочными переменными. Оператор ref предоставляет нам ссылку на нужную переменную, и по этой ссылке, мы присваиваем или получаем нужную переменную.
 - Ключевое слово ref перед переменной берёт адрес в памяти, по которому располагается переменная и помещает этот адрес в переменную.
В переменную попадает не само значение, которое хранится в переменной, а именно ее адрес в памяти. Ключевое слово ref, которое стоит перед именем типа, говорит нам о том, что создаваемая переменная, должна в себе хранить только адреса в памяти, но не значения.
 - Ключевое слово ref позволяет нам получить адрес целевой переменной в оперативной памяти. При этом, при обращении к ссылочной переменной, мы получаем не адрес в числовой форме, а значение по этому адресу.
 - Адреса переменных в памяти, ещё называют ссылками.
 - Ссылочные переменные, не могут в себе содержать адреса констант
 - **Стэк** – это определенная служебная область памяти в ОЗУ, в которой мы можем сохранять промежуточные вычисления или значения.
 - Адрес ячейки памяти в 32-х битовых архитектурах **физически** задаётся тридцатью двумя битами или четырьмя байтами. В 64-х битовых архитектурах, адрес ячейки задаётся шестьюдесятью четырьмя битами или восьмью байтами. В C# для

удобства работы тип ссылочной переменной совпадает с типом той переменной, на которую ссылается эта ссылочная переменная.

Закрепление материала

- Что такое тернарный оператор?
- Для чего используется тернарный оператор, чем он лучше/хуже if-else?
- Что можно подставить в качестве выражения истинности/ложности тернарного оператора?
- Как правильно обращаться с разными типами в выражениях тернарного оператора и при объявлении переменной слева от тернарного оператора?
- Что такое оператор ref?
- Для чего может использоваться оператор ref в тернарном операторе?

Самостоятельная деятельность учащегося

- Задание 1

Ознакомьтесь с дополнительными материалами к уроку.

- Задание 2

Напишите программу, в которой получите от пользователя два целых числа в переменные *x* и *y*. Для каждого из полученных чисел, используя тернарный оператор, выполните проверку на то, является ли число четным, и если число является четным – оставьте его значение неизменным, а если число является нечетным – увеличьте его значение на 1.

Создайте строковую переменную *result*, в которую присвойте результат выполнения тернарного оператора. В этом тернарном операторе выполните проверку на то, делится ли сумма чисел, полученная в первой части задачи, на 4 без остатка. Если сумма делится на 4 без остатка – переменной *result* присвойте фразу «Сумма значений {*x*} и {*y*} делится на 4» (используйте интерполяцию строк). Если сумма на 4 не делится – присвойте фразу «Сумма значений {*x*} и {*y*} на 4 не делится». Выведите значение *result* на экран.

- Задание 3

Напишите программу, в которой:

1. Получите от пользователя два целых числа.
2. Используя выражение, в котором тернарный оператор располагается **слева** от оператора присвоения, выполните проверку на то, что первое введенное число больше или равно второму.
3. (в том же выражении) Если проверка возвращает *true* – первому числу должно присвоиться значение 100, иначе 100 должно присвоиться второму введенному числу.
4. Выведите информацию о числах и их итоговых значениях на экран консоли.

Рекомендуемые ресурсы

<https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/operators/conditional-operator>

<https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/operators/conditional-operator#conditional-ref-expression>

<https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/ref#ref-locals>

https://ru.wikipedia.org/wiki/Тернарная_условная_операция