

## ОСОБЕННОСТИ ПЕРЕГРУЗКИ ОПЕРАТОРА НЕЯВНОГО ПРИВЕДЕНИЯ ТИПОВ В ЯЗЫКЕ C#.

Неявное преобразование типов – операция, выполняемая всякий раз, когда требуется один фундаментальный тип данных, но предоставляется другой, и пользователь не указывает компилятору, как выполнить конвертацию.

В основном неявное приведение, которое гораздо легче читается в коде по сравнению с явным, следует использовать тогда и только тогда, когда нет риска потерять данные или пропустить исключения. Явное приведение должно использоваться во всех других ситуациях. Например, при преобразовании большого целого числа со знаком в меньшее целое число без знака может оказаться так, что значение потеряет свой знак или будет слишком большим для преобразования, поэтому более подходящим является явное приведение.

Есть 2 основных способа неявного преобразования типов:

1. числовое расширение;
2. числовая конверсия.

Числовые расширения – это преобразование значения из меньшего по размеру типа данных в более большой. Числовые расширения абсолютно безопасны, поэтому неявное преобразование в этом способе используется повсеместно.

Числовые конверсии – это преобразование значения из более крупного типа данных в аналогичный, но более мелкий тип данных, или конвертация между разными типами данных.

В отличие от расширений, которые всегда безопасны, конверсии могут (но не всегда) привести к потере данных. Поэтому в любой программе, где выполняется неявная конверсия, компилятор будет выдавать предупреждение.

Во всех случаях, когда происходит конвертация значения из одного типа данных в другой, который не имеет достаточного диапазона для хранения конвертируемого значения, результаты будут неожиданные. Конвертация возможно и будет проходить успешно, но только если значения будут подходить по диапазону (например `Int >> double`).

Синтаксис для оператора неявного преобразования:

**`public static implicit operator result-type(op-type operand)`**

Тип результата - это тип данных для возвращаемого значения операции. Тип операции - это тип данных для операнда, который должен быть преобразован. Один из двух типов данных должен совпадать с классом, в котором выполняется объявление.