**Автоматическое преобразование типов**

Когда данные одного типа присваиваются переменной другого типа, *неявное* преобразование

типов происходит автоматически при следующих условиях:

• оба типа совместимы;

• диапазон представления чисел целевого типа шире, чем у исходного типа.

Если оба эти условия удовлетворяются, то происходит *расширяющее преобразование.*

Например, тип int достаточно крупный, чтобы вмещать в себя все действительные

значения типа byte, а кроме того, оба типа, int и byte, являются совместимыми целочисленными

типами, и поэтому для них вполне возможно неявное преобразование.

Преобразование типа long в тип double является расширяющим и выполняется автоматически.

Если тип long может быть преобразован в тип double неявно, то обратное преобразование

типа double в тип long неявным образом невозможно, поскольку оно не является расширяющим.

Помимо упомянутых выше ограничений, не допускается неявное взаимное преобразование

типов decimal и float иди double, а также числовых типов и char или

bool. Кроме того, типы char и bool несовместимы друг с другом.

**Приведение несовместимых типов**

*Приведение* — это команда компилятору преобразовать

результат вычисления выражения в указанный тип. А для этого требуется явное

преобразование типов. Ниже приведена общая форма приведения типов.

(*целевой\_тип) выражение*

Здесь *целевой\_тип* обозначает тот тип, в который желательно преобразовать указанное

*выражение.* Рассмотрим для примера следующее объявление переменных.

***double х, у;***

Если результат вычисления выражения х/у должен быть типа int, то следует записать

следующее.

***(int) (х / у)***

Несмотря на то что переменные х и у относятся к типу double, результат вычисления

выражения х/у преобразуется в тип int благодаря приведению. В данном примере

выражение х/у следует непременно указывать в скобках, иначе приведение к типу

int будет распространяться только на переменную х, а не на результат ее деления на

переменную у.

Если приведение типов приводит к *сужающему преобразованию,* то часть информации

может быть потеряна. Например, в результате приведения типа long к типу int

часть информации потеряется, если значение типа long окажется больше диапазона

представления чисел для типа int, поскольку старшие разряды этого числового значения

отбрасываются. Когда же значение с плавающей точкой приводится к целочисленному,

то в результате усечения теряется дробная часть этого числового значения.

**Преобразование типов в выражениях**

Преобразования типов выполняются по принятым в C# *правилам продвижения типов.*

Ниже приведен алгоритм, определяемый этими правилами для операций с двумя

операндами.

ЕСЛИ один операнд имеет тип decimal, ТО и второй операнд продвигается

к типу decimal (но если второй операнд имеет тип float или double, результат

будет ошибочным).

ЕСЛИ один операнд имеет тип double, ТО и второй операнд продвигается

к типу double.

ЕСЛИ один операнд имеет тип float, ТО и второй операнд продвигается к типу

float.

ЕСЛИ один операнд имеет тип ulong, ТО и второй операнд продвигается

к типу ulong (но если второй операнд имеет тип sbyte, short, int или long,

результат будет ошибочным).

ЕСЛИ один операнд имеет тип long, ТО и второй операнд продвигается к типу

long.

ЕСЛИ один операнд имеет тип uint, а второй — тип sbyte, short или int, ТО

оба операнда продвигаются к типу long.

ЕСЛИ один операнд имеет тип uint, ТО и второй операнд продвигается к типу

uint.

ИНАЧЕ оба операнда продвигаются к типу int.