

# キャスト

## 型の変換

異なる型どうしの算術演算は「2項数値昇格」のアルゴリズムで  
暗黙のうちにサイズの大きな型にあわせて演算が実行されました

ところが、暗黙の型変換による恩恵が受けられない場合もあります  
基本的に short から int へ変換する拡張変換は暗黙の型変換が発生します

```
short x = value;  
int y = x;
```

value は、何らかの数値リテラルや short 型の変数と考えれば  
この代入式は問題なくコンパイルができます  
short から int への型変換は、数値表現範囲が広がるだけなので  
暗黙のうちに変換してもデータが失われるようなことはありません

しかし、int を short に変換するような**縮小変換はできない**のです  
もし int 型の変数を short に代入しようするとコンパイルエラーが発生します  
これは、縮小変換によって上位ビットのデータが失われる可能性があるためです  
そのため、縮小変換には**キャスト式**を使って明示的型キャストを行います

**( type ) expr**

type には変換する型名を、expr には型変換を行う式を指定します  
こうすれば、暗黙に変換できない場合に明示的に変換を指定できます

```
class Test {  
    static void Main() {  
        ushort us = 0xA5FF;  
        byte sb = (byte)us;  
  
        System.Console.WriteLine("ushort = " + us);  
        System.Console.WriteLine("byte = " + sb);  
    }  
}
```

このプログラムは、ushort 型変数 us を byte 型に縮小変換します  
us の値は符号なし 16bit で 1010 0101 1111 1111 となっています  
これを 8bit の byte 型にキャストすると上位 8bit が失われます  
ということは、下位 8bit が代入され 1111 1111、すなわち sb は10進数 255 となります

キャスト式の注意点としては、**(type) の後に演算子を置かない**ということです  
たとえば、以下のキャスト式は非常にあいまいといえます

**( type ) - variable**

これは、type と variable の差を求めている算術演算にも受け取れますし  
-variable を type にキャストしようとしているとも読み取れます

この場合 type が既定のキーワード(int や long)ならばキャストとなります  
キーワード以外の型の識別子だった場合はキャスト式ではないと判断されます  
安全にキャストするには (type)(-variable) と記述することが推奨されます

[前のページへ](#)

[戻る](#)

[次のページへ](#)