桁あふれ

桁あふれの監視

数値の演算をコンピュータで行っていると、アナログでは存在しない問題が発生します
それは、演算結果を格納するメモリサイズが得られた結果よりも少ない状態です
となれば、当然オーバーフローして正しい計算結果は得られません

オーバーフローの処理は、一般では溢れた上位ビットを切り捨てます
しかし、重要な計算を行う時に誤った結果をプログラムが表示する危険性もあるため
アプリケーションの種類によっては、オーバーフローの発生を知る必要が生じることもあります

そこで、C# はコンパイラに依存したオーバーフローのチェックだけではなく
プログラマが明示的にオーバーフローの監視を行えるように設計されています
それが **checked** 演算子による演算の監視です

checked (expression)

expression には、桁あふれを監視する式を指定します
checked 演算子は expression を評価して、問題がなければ結果を返します
しかし、オーバーフローが発生すると **System.OverflowException** を発生させます

expression に定数式を指定すれば、これは静的に判断できるので意味がありません
そのため、コンパイルエラーが発生します

```
class Test {
 static void Main() {
  try {
   byte x = 255;
   x = checked((byte)(x * 4));
  }
  catch (System.OverflowException err) {
   System.Console.WriteLine(err.Message);
  }
 }
}
```

このプログラムでは byte 型の 8 ビットの数値を演算しています
ここで、checked を用いなかった場合の動作はコンパイラや開発環境に依存します
明示的ないとがある場合、checked を用いて桁あふれの監視を行います

変数 x には 255 という数値が格納されていて、これは byte 型の最大値です
これに 4 を乗算するわけですから、当然オーバーフローが発生します

しかし、プログラムの大部分で桁あふれが発生すると問題があるという場合
ありとあらゆる演算を checked 演算子で行うのは、大変な手間がかかります
そのような場合は checked **ステートメント**をもちいます

checked block

block には、checked を適用するステートメントブロックを指定します
この点を除けば、checked ステートメントの効果は checked 演算子に等しいです

```
class Test {
 static void Main() {
  checked {
   try {
    byte x = 255;
    x = (byte)(x * 4);
   }
   catch (System.OverflowException err) {
    System.Console.WriteLine(err.Message);
   }
  }
 }
}
```

多くのコンパイラで互換性を取り、コンパイラ結果を同じにするには
このように checked などで桁あふれチェックコンテキストを制御するべきです

因みに、逆に例外を発生させたくない場合はどうしましょう
つまり、桁あふれが発生するとあふれた部分を**切り捨てる**という場合です
これは、一般的にはデフォルトの状態ですが、環境によって異なる可能性もあります
デフォルトに依存したくない場合、やはり明示的に制御するべきです

あふれ部を切り捨てる場合は **unchecked** 演算子を用います
また、広範囲に指定するならば unchecked ステートメントも存在します

unchecked (expression)
unchecked block

expression には、桁あふれをチェックする式を指定します
block はこれをステートメントブロック単位で行います

```
class Test {
```

```
 static void Main() {
  byte x = 255;
  System.Console.WriteLine(unchecked((byte)(x * 4)));
 }
}
```

このプログラムの演算では、オーバーフローが発生しますが
unchecked 演算子を明示的に指定しているため、例外は発生しません
こうすることで、開発環境のデフォルトの処理に依存することを避けることができます
x に4 を乗算したのだから、実行すれば左に 2 回シフトした結果が得られるはずです
ただし、当然あふれた部分は切り捨てられています

1111 1111 -> 11 1111 1100

桁あふれした上位2ビットの赤い部分はカットされます
よって 1111 1100 (255 - 3 = 252) がコンソールに出力されます

---

checked

checked (expression)
checked block

演算の桁あふれをチェックします
桁あふれが発生した場合 System.OverflowException をスローします

expression - チェックする式を指定します
block - チェックするステートメントブロックを指定します

unchecked

unchecked (expression)
unchecked block

演算の桁あふれの報告を行わないことを明示します
桁あふれが発生した場合、あふれた上位ビットを切り捨てます

expression - 報告を行わない式を指定します
block - 報告を行わないステートメントブロックを指定します

前のページへ　戻る　次のページへ