1の加減算

インクリメントとデクリメント

プログラムの中では、繰り返し処理やグラフィックス処理などで
頻繁に**変数の現在の値 + 1**という計算を行います
このような計算は、通常の算術演算子で次のように求められます

variable = variable + 1;

これは、variable に 1 を加算した値を variable に保存しています
= 演算子は + 演算子よりも優先度が低いのでこれは正確に計算することができます
しかし、このような処理には専用の演算子**インクリメント**が用意されています

インクリメント演算子は**変数の前か後に ++** を付加します
この違いは、演算子の優先順位に影響します

++variable
variable++

++variable は**前置き**インクリメント演算子と呼ばれ
variable++ を**後置き**インクリメント演算子と呼び区別します
インクリメント演算子は両方とも変数に現在の値 + 1 という結果を与えます

```
class Test {
 static void Main() {
  int i = 10;

  i++;
  System.Console.WriteLine(i);
  ++i;
  System.Console.WriteLine(i);
 }
}
```

コンソールには 11 と 12 という数字が表示されます
この結果からも、i++ と ++i がそれぞれ i に 1 加算していることがわかりますね

前置きと後置きの違いは多項式でインクリメントするとわかります
前置き演算子はインクリメントしてから式を評価しますが
後置き演算子は式を評価してから変数をインクリメントするという違いがあります

つまり、前置きの場合はインクリメントしてから値を返しますが
後置きは値を渡してからインクリメントするため、その結果は異なります

```
class Test {
 static void Main() {
  int i = 10;

  System.Console.WriteLine(++i);
  System.Console.WriteLine(i++);
  System.Console.WriteLine(i);
 }
}
```

この結果は、次のようになります

11
11
12

WriteLine(++i) は、i = i + 1 を計算してから WriteLine() メソッドに i を渡しますが
WriteLine(i++) は、WriteLine() メソッドに i を渡してから i = i + 1 の計算をします
その証拠に、WriteLine(i++) の時点では 11 ですが
その後の変数 i は 12 になっていることが確認できます

同様に、現在の変数の値から**1 減算**する専用の演算子もあります
この扱いはインクリメントと同じで、これを**デクリメント**と呼びます
デクリメントは、変数の前後いずれかに **--** を付加します

--variable
variable--

前置きと後置きの違いは、インクリメント演算子と同じです

```
class Test {
 static void Main() {
  int i = 10;

  System.Console.WriteLine(--i);
  System.Console.WriteLine(i--);
  System.Console.WriteLine(i);
 }
}
```

今度は 10 の値から 1 づつ減算されています
インクリメント同様に i＝i－ 1 または i＝i＋(-1) という計算をしたい場合は
デクリメント演算子を使って計算することができます

前のページへ　戻る　次のページへ