

Boost.勉強会 #7 東京 (2011-12-03)

# C++ TIPS 2 インクリメント編

# 概要

- 主に cpp11 ML でご紹介してきた tips を C++ の仕様をより掘り下げた形でまとめ直してみました。
- 今回はインクリメントにフォーカスした内容です。

C++ Tips

# インクリメント

# インクリメントとは？

- プログラミング用語としてのインクリメントは値を+1したりあるいはイテレータなどを一つ進めることを意味し、デクリメントは逆に-1したり一つ戻したりすることを意味します。
- C言語およびC言語を文法のベースにしている多くの言語では++がインクリメント演算子として、--がデクリメント演算子と使えます。

# インクリメントとは？

- インクリメント演算子もデクリメント演算子もひとつの値を対象とする単項演算子です。
- ++iのようにインクリメント演算子を演算対象の値の前に置くと前置インクリメントと呼ばれi++のように値の前に置くと後置インクリメント呼ばれます。

# インクリメントとは？

- ++i であっても i++ であっても i はインクリメントされます。
- 前置と後置の違いは式の評価値が、前置の場合はインクリメント後の値となり、後置の場合はインクリメント前の値となることです。

# 未定義動作

- 関数引数内で同じ変数に大してインクリメントやデクリメントをやっちゃうと未定義動作になってしまうので気をつけましょう。

```
int a = 0;  
++a; // OK  
++a; // OK  
f(++a, ++a); // NG
```

# 未定義動作

- ポインタや参照を使うと分かり難い形でこの未定義動作になってしまうことがあるので要注意です。

```
void f(int &a, int &b) { h(++a, ++b); }  
void g(int &c) { f(c, c); } // NG
```

# インクリメントの定義

- C++ではユーザー定義のクラスであってもインクリメント/デクリメントを定義できます。
- 言語使用上はその中身は別にインクリメント/デクリメントに類する処理である必要はありません。
  - ただ、混乱の元になるだけなのでインクリメント/デクリメントおよびそれに類する処理以外に使うのはやめておきましょう。

# インクリメントの定義

```
class hoge {  
    public:  
        typedef hoge this_type;  
        int value;  
        this_type & operator ++ () { // 前置インクリメント  
            ++value;  
            return *this;  
        }  
        this_type operator ++ (int) { // 後置インクリメント  
            this_type temp(*this);  
            ++value;  
            return temp;  
        }  
};
```

# メンバー関数呼び出し風の使用

- 通常のメンバー関数と同じ形式での呼び出しも可能です。

```
hoge hoge;  
hoge.operator++(); // 前置インクリメント  
hoge.operator++(0); // 後置インクリメント
```

# メンバー関数呼び出し風の使用

- 後置インクリメントの引数は通常のメンバー関数と同様に使おうと思えば普通に使えます。
  - 通常のメンバー関数と同じ形式で呼び出す場合に限る話であり、`i++`のような形式では呼び出し側から任意の値を渡すことはできません。

# 後置で複製を作らないには？

- デストラクタでインクリメントを行う一時オブジェクトを作ること、後置インクリメントの実装で自身の複製を作らないようにすることも可能です。
  - どのみち、なんらかの一時オブジェクトが必要であることには変わりがない。
  - 型が変わってしまうという副作用がある為、あまり推奨できません。
  - 複製のコストが高いクラスやストリームのイテレータなどで有効な手段となります。

# どちらでもよい場合は前置

- 通常、組み込み型以外の後置インクリメントではインクリメント前のコピーである一時オブジェクトを返すことになり、前置インクリメントでも構わない場合に後置インクリメントを使うことは無駄なコストを支払うことになるのでどちらでもよい場合は前置インクリメントを使いましょう。
- 組み込み型の場合はどちらでも気にする必要はありませんが、統一性の観点やコード修正により変数の型が組み込み型でなくなった場合に備えて、組み込み型でも前置インクリメントを使いましょう。

C++ Tips

質疑応答

C++ Tips

ご清聴ありがとうございました。