C++ コーディング規約

2015/11/18
Version 1.2

1. 命名規則

- **変数名**

小文字で開始し,先頭以外の単語の区切りを大文字としてつなげる(lower camel case).
（例：variableName）

- **変数のプレフィックス（接頭辞）**

メンバ変数にはプレフィックス m をつける.（例：mVariableName）
グローバル変数にはプレフィックス g をつける.（例：gVariableName）

- **定数名**

全て大文字にして, 単語の間には_(アンダーバー)を使用する.（例：CONSTANT_NAME）

- **型名（クラス・構造体・列挙型・typedef 等）**

全ての単語の先頭を大文字としてつなげる（pascal case).
（例：ClassName）

- **関数名**

外部リンケージ（ファイルの外部でも利用できる性質）を持つ関数は pascal case を使用する.（例：FunctionName）

内部リンケージ（ファイルの内部でしか利用できない性質）を持つ関数は lower camel case を使用する.（例：functionName）

- **名前空間名**

全て小文字にする.

- **マクロ名**

定数名と同様.

- **列挙子名**

定数名と同様.

- **ラベル**

変数名と同様.

2. 書式

● インデント

スペースのみを使い，半角スペース 4 文字分とする.
環境によって見え方が変わることを防ぐため，コードにはタブを使用しないようにする.

● 1 行の長さ

最長 120 文字程度とする.

● 1 ファイルの最大行数

特に指定はしないが，できるだけ短くするべきである.

● ″{ ″（中括弧）の書式

{ （中括弧）の始めは改行せずに前方に半角スペースを 1 つ入れて始める.
簡潔な if 文，for 文などであっても{（中括弧）を省略しない.

```
// 例：
function() {
    // do something
}
```

● switch 文

default ラベルは必須とする.

● クラスの書式

public, protected, private の順番でセクションがくるようにする.

● 名前空間の書式

名前空間内では余分なインデントをしない.

● コメントの書式

// のみを使用し，一貫性を持たせる.

● **プリプロセッサディレクティブ**

プリプロセッサディレクティブのシャープはインデントを無視して，行の先頭に置く.

```
// 例:
function() {

#ifdef DEBUG // 行の先頭に置く
    // do something
#endif

}
```

3. クラス

● **メンバイニシャライザ**

全てのスーパークラスとメンバ変数をコンストラクタのメンバイニシャライザで宣言順に初期化する.
また，メンバイニシャライザで初期化しないものは必ずコメントを残すようにする.

● **多重継承**

最初１つを除く，他のすべてのスーパークラスが純粋なインタフェースである場合に限り，多重継承を使うことを許可する.

● **private な継承**

基本的には包含（コンポジション）を優先すべきだが， デザインの方針として有効になる場合は許可する.
（例: boost::noncopyable を継承するとき）

4. ファイル構造

● **拡張子**

原則，次の通りとする.
ヘッダーファイル　　: ".h"
ソースファイル　　: ".cpp"
プリコンパイルヘッダ : ".pch"

5. その他規則

● **goto**

goto文の危険性を考慮した上で,綺麗で簡潔に処理が記述できる場合は使用してもよい.

- **friend**

friend の危険性を考慮した上で，妥当な範囲内であれば， friend クラスや friend 関数を使用してもよい.

- **null pointer**

NULL， 0 ではなく nullptr を使用する.

- **キャスト全般**

static_cast<>() といった，C++のキャストを使用する.
int y = (int)x; といった C スタイルのキャストは使用しない.

- **dynamic_cast**

使用を禁止する.
dynamic_cast が必要な場合には設計に問題がある可能性が高いため.

- **実行時型情報（RTTI）**

使用を禁止する.
実行時に型を問い合わせるというのは，たいていの場合，クラス設計に問題がある可能性が高いため.

6. 参考文献・URL

[1] 「Google C++スタイルガイド 日本語訳 Revision 3.199」
<http://www.textdrop.net/google-styleguide-ja/cppguide.xml>
(2015/1/19 アクセス)

[2] スコット・メイヤーズ・著，小林 健一郎 ・翻訳 (2006)『Effective C++ 原著第3版』ピアソン・エデュケーション

[3] ロベール・著 (2007)『ロベールの C++入門講座』毎日コミュニケーションズ

[4] 「IPA 組込みソフトウェア開発向け コーディング作法ガイド [C++言語版]」
<http://www.ipa.go.jp/files/000005142.pdf>
(2015/1/19 アクセス)