

プロジェクト名 : C++プログラム共通

IT-brainsシステム開発

C++コーディング規約

(バージョン 1.0.0.3)

承認	審査	設計	
		IT-brains 06.12.22 平原	

変更履歴

変更番号	変更ページ	変更内容	担当	承認
1.0.0.0	-	初版	平原 2022/01/10	平原 2022/01/10
1.0.0.1	01 02 05	コーディング規約の使用条件を追加 プロジェクト構成を具体的に記述 コーディング上の制限事項を一部変更	平原 2024/01/11	平原 2024/01/11
1.0.0.2	02	ItbDoubleを『少数型拡張』に変更	平原 2024/08/01	平原 2024/08/01
1.0.0.3	02	日本日時クラス『Jpn~』を追加	平原 2024/12/22	平原 2024/12/22

01 目的と使用条件

01.1 コーディング規約の目的

プログラム製造に携わる開発者全員の意識統一を図ることを目的として作成されており、厳守すること。

コーディング規約を規定することにより、以下の効果を期待する。

- ・開発における情報の共有化
- ・開発成果物の生産性の向上
- ・開発成果物の保守性の向上

01.2 コーディング規約の使用条件

コーディング規約の使用条件を以下とする。

- (1) C++プログラムの開発を行う場合に適用する。
- (2) Itbライブラリを使用して生産性を向上する。

承認	審査	設計	
		IT-brains 06.01.11 平原	


ドキュメント名称
C++コーディング規約 (目的と使用条件)

資料番号		
登録番号	通番	追番
PG-1000	01	01

02 プロジェクト構成

02.1 システム共通部品

ItbLibrary └ ItbLibAttr └ ItbDateTime └ ItbDouble └ ItbString └ ItbLibBaseCtrl └ ItbProcess └ WinService └ TaskTray └ ItbLibComm └ Ping └ SerialPort └ SocketServer └ SocketClient └ FtpClient └ BluetoothServer └ ItbLibDb └ adoDbAdapter └ ItbLibDev └ DeviceInfo └ Clipboard └ RemovableDisk └ Keyboard └ WinShutdown └ ItbLibImage └ Dib	Itbライブラリ	システム共通処理
	属性	Itb属性 日時拡張 小数型拡張 文字列操作
	基本コントロール	Itb基本コントロール プロセス操作 Windowsサービス操作 タスクトレイ雛形
	通信	Itb通信 Ping検査 シリアルポート ソケットサーバ ソケットクライアント FTPクライアント Bluetoothサーバ
	データベース	Itbデータベース ADOデータベースアダプタ
	デバイス	Itbデバイス デバイス情報参照 クリップボード操作 リムーバブルディスク操作 キーボード操作 Windowsシャットダウン操作
イメージ	Itbイメージ デバイス独立ビットマップ操作	

承認	審査	設計	
			

ドキュメント名称
C++コーディング規約 (プロジェクト構成)

資料番号		
登録番号	通番	追番
PG-1000	02	01

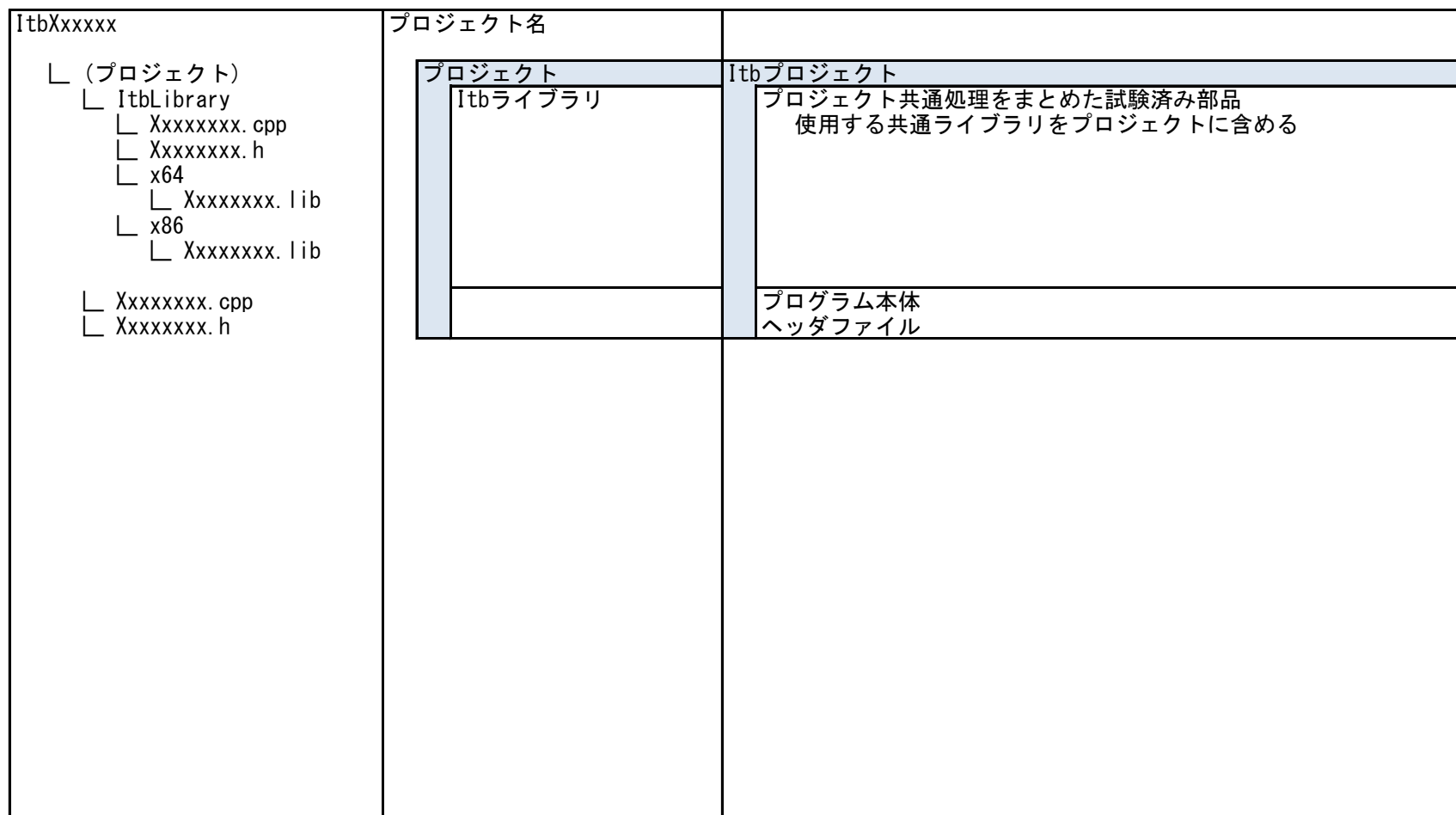
ItbLibrary └ ItbLibFile └ ItbFile └ ItbFolder └ Registry └ ItbLibLog └ ItbLog └ その他 └ ItbBaseFile └ TextFile └ FixedFile └ FixedDataset └ IniFile └ ZipFile └ LzhFile └ SevenZipFile └ RtfFile └ WaveFile └ JpnDateTime └ JpnHoliday └ JpnSixday └ JpnSolarTerm └ SelectFolderDlg └ SharedMemory └ ScreenInfo	Itbライブラリ	システム共通処理
	ファイル	Itbファイル Itb基底ファイル Itb基底フォルダ レジストリ操作
	ログ	Itbログ ログ操作
	その他	Itbファイル拡張、フォルダ選択操作等 Itbファイル拡張 テキストファイル操作 固定長ファイル操作 固定長データセット操作 INIファイル操作 ZIPファイル操作 LZHファイル操作 7-zipファイル操作 RTFファイル操作 WAVEファイル操作 日本日時 日本祝日 日本六曜 日本二十四節気 フォルダ選択ダイアログ 共有メモリ操作 スクリーン情報参照

承認	審査	設計	
			

ドキュメント名称
C++コーディング規約 (プロジェクト構成)

資料番号		
登録番号	通番	追番
PG-1000	02	02

02.2 システム共通部品を使用したプロジェクト構成



承認	審査	設計	
			

ドキュメント名称
C++コーディング規約 (プロジェクト構成)

資料番号		
登録番号	通番	追番
PG-1000	02	03

03 命名規則

03.1 管理番号

プログラムに付加する管理番号体系を以下とする。

$\frac{99}{①} . \frac{99}{②} . \frac{99}{③} . \frac{99}{④}$

- | | |
|-------------|--------------------|
| ①バージョン識別番号 | バージョン番号として識別する番号 |
| ②メジャーリリース番号 | パッケージの種類を表す番号 |
| ③マイナーリリース番号 | 仕様変更を行った回数 |
| ④リビジョン番号 | 仕様変更・不具合改修に対する改修回数 |

03.2 プログラムファイル名

プログラムファイルの命名体系を以下とする。

Xxxxxxx.cpp	.cpp	プログラム本体（実体）
Xxxxxxx.h	.h	ヘッダファイル


先頭文字を英大文字、以降英数大小文字の組み合わせで、機能を表現する名詞の組み合わせで表現する。

03.3 クラス名

クラス名の命名体系を以下とする。

C Xxxxxxxx	C	: クラス接頭辞
I Xxxxxxxx	I	: インタフェース接頭辞

先頭文字を英大文字、以降英数大小文字の組み合わせで、機能を表現する名詞の組み合わせで表現する。

承認	審査	設計	
			

ドキュメント名称
C++コーディング規約 (命名規則)

資料番号		
登録番号	通番	追番
PG-1000	03	01

03.6 定数名

定数名の命名体系を以下とする。

XXX_ XXX_ XXX

① ② ③

①機能識別子


変数の型を表現する識別子（プリフィックス）を付加する。
クラス特有の定数の場合は、そのクラスを表現する識別子とする。

②機能種別識別子

機能内の種別を表現する識別子を付加する。

③機能分類定数名

種別を分類する名称を付加する。

承認	審査	設計	
			

ドキュメント名称
C++コーディング規約 (命名規則)

資料番号		
登録番号	通番	追番
PG-1000	03	03

04 コーディングスタイル


04.1 コメント

```

//=====
//! @brief      XXXXXXXXXXXXXXXXXXXX
//! @file       XXXXXXXX.xxx
//! @retval     各関数の冒頭説明参照
//! @note       XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
//! @author     _.(XXXXXXX)
//! @date       9999/99/99 新規作成
//! @version    9.9.9.9 9999/99/99 odt 初版
//             9.9.9.9 9999/99/99 _.(XXXXXXX) XXXXXXXXXXXXXXX
//! Copyright (C) 9999 IT-brains Co.,Ltd. All rights reserved.
//=====
class CMyClass {
public:
    //! My関数
    void MyFunction();
}

//=====
//! @brief      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
//! @param      [I/O] axXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
//! @return     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
//! @note       XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
//=====
void CMyClass::MyFunction() {
    //! 説明コメント ({} を使用しない)
    //! 処理取消コメント (1行コメントアウトは"/"/、複数行コメントアウトは"{}"で行う)
    // strcpy(p, "コピー元");
}

```

承認	審査	設計	
			

ドキュメント名称
C++コーディング規約 (コーディングスタイル)

資料番号		
登録番号	通番	追番
PG-1000	04	01


```

void CMyClass::MyFunction() {
  //! コーディング行を追加する場合
  // 9999/99/99 ver9.9.9.9 INS S XXXXX
  :
  // 9999/99/99 ver9.9.9.9 INS E

  //! コーディング行を修正する場合
  // 9999/99/99 ver9.9.9.9 UPD S XXXXX
  :
  // 9999/99/99 ver9.9.9.9 UPD E

  //! コーディング行を削除する場合
  // 9999/99/99 ver9.9.9.9 DEL S XXXXX
  // :
  // 9999/99/99 ver9.9.9.9 DEL E
}

```

承認	審査	設計	
			

ドキュメント名称
C++コーディング規約 (コーディングスタイル)

資料番号		
登録番号	通番	追番
PG-1000	04	02

04.2 if文


- (01) ステートメント構文は、ネストさせること。
- (02) ステートメント構文は、必ず {} でくくること。(1文の場合も必ず {} を使用すること)
- (03) {} は制御文の文末に記入せず、改行して次の行に記入すること。

```
if (0 == ret) {  
    data = text;  
} else {  
    :  
}
```

04.3 switch文

- (01) {} は制御文の文末に記入せず、改行して次の行に記入すること。
- (02) case キーワードは、switch キーワードと同じ桁位置に記入すること。
- (03) case 内のステートメント構文は、ネストさせること。
- (04) default ブロックを必ず指定すること。

```
switch (mode) {  
case 1:  
    a += 1;  
    break;  
default:  
    break;  
}
```

承認	審査	設計	
			

ドキュメント名称
C++コーディング規約 (コーディングスタイル)

資料番号		
登録番号	通番	追番
PG-1000	04	03

04.4 for文

- (01) ステートメント構文は、ネストさせること。
- (02) ステートメント構文は、必ず `{}` でくくること。(1文の場合も必ず `{}` を使用すること)
- (03) `{}` は制御文の文末に記入せず、改行して次の行に記入すること。

```
for (cnt = 0; 10 > cnt; cnt++) {  
    data[cnt] = text[cnt];  
}
```

04.5 while文

- (01) ステートメント構文は、ネストさせること。
- (02) ステートメント構文は、必ず `{}` でくくること。(1文の場合も必ず `{}` を使用すること)
- (03) `{}` は制御文の文末に記入せず、改行して次の行に記入すること。
- (04) ループ内に終了条件処理を必ず記述すること。

```
while (10 > cnt) {  
    if (...) {  
        break;  
    }  
    :  
    :  
    cnt++;  
}
```

承認	審査	設計	
			

ドキュメント名称
C++コーディング規約 (コーディングスタイル)

資料番号		
登録番号	通番	追番
PG-1000	04	04

05 制限事項

05.1 コーディング上の制限事項


コーディングにおける全般的な制限事項を以下に記述する。

- (01) ネスト(桁あわせ)するタブ幅を4とし、タブ文字の使用はコメント以外禁止する。
- (02) 変数宣言は、1行に1変数を基本とする。
- (03) 文字配列定義について、他言語とのアライメントの関係により文字列の長さが制限されるため、文字列の長さを8の倍数にて定義すること。
- (04) 動的配列を使用する場合、使用後に必ずメモリの開放を行うこと。
※ 例外等、予定外の処理が発生しても必ず開放するように注意する

```
CMyClass clsMyClass = new CMyClass();  
  
try {  
    :  
    delete clsMyClass;  
} catch (...) {  
    delete clsMyClass;  
}
```

- (05) ヘッダファイルについて、二重インクルードを避けるため、以下の定義で囲むこと。
※ define名は、_機能ID+_INCLUDE で構成し、機能IDは先頭文字が英字・以降英数字で表現する

```
#ifndef _機能ID_INCLUDE  
#define _機能ID_INCLUDE  
:  
#endif // _機能ID_INCLUDE
```

承認	審査	設計	
			

ドキュメント名称
C++コーディング規約 (制限事項)

資料番号		
登録番号	通番	追番
PG-1000	05	01

- (06) 演算子の使用方法について、以下の規約に従うこと。
※ [付録B] 参照
- (07) 空白の使用箇所について、以下の場所に空白を挿入することを禁止する。
- ・ メソッドと開き括弧の間
 - ・ ドット演算子(.)の前後
 - ・ 単項演算子とそのオペランドの間
 - ・ キャスト型とキャストされる式の間
 - ・ 開き角括弧{ }の直後、又は閉じ角括弧}の直前
 - ・ セミコロン直前
- (08) 継続行の使用について、複数引数を指定する関数などは1行1引数で記述し、以降次行にネスト(桁あわせ)するタブ幅を4とし記述する。

```
void CMyClass::MyFunction (
    char *apParam1,
    int aiParam2
) {
    char cBuffer[256];
    sprintf(cBuffer, "%s=%d",
        "値",
        100);
}
```

- (09) 演算時の優先順位を明確にするため、括弧「()」を使用してコーディングすること。
- (10) 複合論理式の評価方法について、「&&」「||」等で結合された論理式は左から右に評価され、評価が確定した時点でその後の評価は行われない(短絡評価)。
従って、後の方に関数を含んでもその関数は実行されないのに注意すること。
尚、短絡評価を利用してコーディングするのは許可する。

承認	審査	設計
		

ドキュメント名称
C++コーディング規約 (制限事項)

資料番号		
登録番号	通番	追番
PG-1000	05	02

- (11) リテラル数値・文字について、極力使用せずに名前付定数の使用を推奨する。
 リテラル数値・文字と変数との比較の際には、リテラル数値・文字を先に、変数を後に記述する。

```
#define DATETIME_WEEKDAYS 7

// cnt = DATETIME_WEEKDAYS などの間違いをコンパイル時に防止する
for (cnt = 0; 7 > cnt; cnt++)           → ×
for (cnt = 0; DATETIME_WEEKDAYS > cnt; cnt++) → ○
```

承認	審査	設計
		IT-brains 04.01.10 平原

ドキュメント名称
C++コーディング規約 (制限事項)

資料番号		
登録番号	通番	追番
PG-1000	05	03

付録A 型識別文字規約

プリフィックス	型名	書式	使用例
s	short	sX~	short sNumber;
i	int	iX~	int iNumber;
l	long	lX~	long lNumber;
ll	long long	llX~	long long llNumber;
w	WORD	wX~	WORD wNumber;
u	UINT	uX~	UINT uNumber;
dw	DWORD	dwX~	DWORD dwNumber;
qw	unsigned long long	qwX~	unsigned long long qwNumber;
f	float	fX~	float fNumber;
d	double	dX~	double dNumber;
ld	long double	ldX~	long double ldNumber;
col	COLORREF	colX~	COLORREF colString;
bo	bool	boX~	bool boFlag;
	BOOLEAN		BOOLEAN boFlag;
	BOOL		BOOL boFlag;
c	char	cX~	char cValue;
	TCHAR		TCHAR cValue;
bt	BYTE	btX~	byte btValue;
wc	WCHAR	wcX~	WCHAR wcValue;
p	char*	pX~	char* pValue;
	BYTE*		BYTE* pValue;
	WCHAR*		WCHAR* pValue;
	LPSTR		LPSTR pValue;
	LPTSTR		LPTSTR pValue;
	LPCSTR		LPCSTR pValue;
	LPCTSTR		LPCTSTR pValue;
str	string	strX~	string strValue;
	char []		char [] strValue;
	TCHAR []		TCHAR strValue;
bstr	BSTR	bstrX~	BSTR bstrValue;
vt	VARIANT	vtX~	VARIANT vtValue;
vp	LPVOID	vpX~	LPVOID vpValue;

