

手羽先クロニクル　　　コーディングルール

目次

1. 命名規則
2. 条件文
3. クラス、関数
4. パスやファイルについて



|  |  |  |
| --- | --- | --- |
| 命名規則名 | 日本語名 | 内容 |
| PascalCase | キャメルケース | 先頭大文字、それ以降の単語区切りも大文字 |
| camelCase | パスカルケース | 先頭小文字、それ以降の区切りは大文字 |
| snake\_case | スネークケース | すべて小文字、単語区切りをアンダ－スコア\_でつなぐ |
| SNAKE\_CASE | 不明 | すべて大文字、単語区切りをアンダ－スコア\_でつなぐ |

* **命名規則**

**本制作は、以下のルールに則る**

|  |  |
| --- | --- |
| クラス | キャメルケース |
| 関数 | キャメルケース |
| 変数 | パスカルケース |
| 定数 | 大文字スネークケース |
| Bool変数 | is~，has~，can~, enable~ 等 |

動作やクラス、格納する情報にふさわしい名前をつける

ハンガリアン(m\_hoge)禁止: メリットよりデメリットが多いことが知られている

略称禁止: 略称を使うと分からなくなる場合がある

* **条件文**
* 条件文が簡潔である
* Returnが簡潔である
* 意図が明確(初期化エラーチェック、null等)
* 一行if禁止{}で必ずくくる
* 3項演算子は原則使わない
* Nullptrは明記する
* 長い条件文は関数化するか、boolでとる

**3.クラス、関数**

* 引数や関数名は、多少冗長になってもコメントを読まなくても意味が伝わるような名前にする。
* コメントをヘッダの宣言部の直前に付ける

コメントの形式は以下の通り

//クラス、関数の動作(実行、使用することでどのような効果が得られるかを書く)

//引数に指定するもの

//戻り値

//注意事項や補足的なもの(あれば)

* 値を変更しない関数や引数は、constを付ける
* cpp内のコメントに関しては処理についての補足や、自分のメモを書くものとする

4.パス、ファイルについて

* パスやファイル名は全部半角のみにする
* フォルダのファイルを必ず分ける（エディターとエクスプローラー両方）