

初級 PG 認定試験 Java コーディング規約

目次

1.ファイル構成	2
(1) ファイル名.....	2
(2) ファイルの位置.....	2
2.命名規則	2
(3) パッケージ名	2
(4) ファイル名	2
(5) クラス名.....	2
(6) 例外クラス.....	2
(7) 定数	2
(8) メソッド名	3
(9) 属性(フィールド)の取得メソッド	3
(10) 属性(フィールド)の設定メソッド	3
(11) boolean 値を返すメソッド.....	3
(12) boolean 変数.....	3
(13) 名前の対称性	4
(14) ループカウンタ.....	4
(15) スコープが狭い名前	4
(16) 意味が読み取れる名前.....	4
3.コメント.....	5
(17) javadoc の活用.....	5
(18) javadoc タグ.....	5
(19) クラスコメント.....	6

1.ファイル構成

(1) ファイル名

public クラスはそのクラス名の 1 ファイルにする。

例: public class Point は, Point.java に入れる。

インナークラスは使わないものとする。

(2) ファイルの位置

プロジェクトのルートディレクトリを決め, パッケージ名の “.” をディレクトリ階層に置き換えた位置に入れる。

例: myProject.framework

パッケージは, <ルートディレクトリ>/myProject/framework ディレクトリに配置。

2.命名規則

(3) パッケージ名

“.” で区切られた文字。

myProject.framework

junit.framework

(4) ファイル名

パブリックなクラス名は, コンパイラの規則により, ファイル名と同じでなければならない(大文字小文字の区別を含めて)。

(5) クラス名

先頭大文字。単語の区切りは大文字。

CapitalizedWithInternalWordsAlsoCapitalized

(6) 例外クラス

クラス名の語尾を Exception としたクラス名。

ClassNameEndsWithException

(7) 定数

大文字を”_”でつないだもの。

UPPER_CASE_WITH_UNDERSCORES

(8) メソッド名

先頭小文字、単語の区切りは大文字。

`firstWordLowerCaseButInternalWordsCapitalized()`

(9) 属性(フィールド)の取得メソッド

属性:x メソッド名:`getX()`

属性:y メソッド名:`getY()`

属性:age メソッド名:`getAge()`

(10) 属性(フィールド)の設定メソッド

属性:x メソッド名:`setX()`

属性:y メソッド名:`setY()`

属性:age メソッド名:`setAge()`

(11) boolean 値を返すメソッド

is + 形容詞、can + 動詞、has + 過去分詞、三単元動詞、三単元動詞 + 名詞、助動詞 + 動詞。

`boolean isEmpty()`

`boolean canGet()`

`boolean shouldContinue()`

`boolean hasChanged()`

`boolean contains(Object)`

`boolean containsKey(Key)`

理由:if、while 文等の条件が読みやすくなる。また戻り値の true の意味が分かりやすい。

(12) boolean 変数

形容詞、is + 形容詞、can + 動詞、has + 過去分詞、三単元動詞、三単元動詞 + 名詞、助動詞 + 動詞。

`boolean isEmpty`

`boolean canPlay`

`boolean shouldContinue`

`boolean hasSent`

`boolean contains`

`boolean containsMoreElements`

(13) 名前の対称性

クラス名、メソッド名を付ける際は、以下の英語の対称性に気を付ける。

add/remove
insert/delete
get/set
start/stop
begin/end
send/receive
first/last
get/release
put/get
up/down
show/hide
source/target
open/close
source/destination
increment/decrement
lock/unlock
old/new
next/previous

(14) ループカウンタ

スコープが狭いループカウンタ、イテレータに i、j、k という名前をこの順に使う。

(15) スコープが狭い名前

スコープが狭い変数名は、型名を略したものを使っても良い。

例: `Random rnd = new Random();`

(16) 意味が読み取れる名前

変数名から役割が読み取れる名前を使う。

悪い例: `copy(s1, s2)`

良い例: `copy(from, to)` あるいは `copy(source, destination)`

3.コメント

(17) javadoc の活用

/** コメント */を多いに活用すること。このコメントは、javadoc や同様のツールによって HTML 形式でのドキュメントに変換することができる。

java のコメントは、3 種類存在する。

/** */ javadoc コメント。html 形式でドキュメント出力される。

/* */ 通常のコメント

// 通常のコメント

クラスのコンストラクタ、メソッド、フィールドには必ず /** */コメントを付ける。

(18) javadoc タグ

/** */コメント中、@から始まるキーワード(javadoc タグ)を使う。

@param 引数(パラメーター)名 引数の説明

@return 戻り値の説明

@exception 例外名 例外の説明

@throws 例外名 例外の説明

(19) クラスコメント

```
/**
 * スタックを表現するクラス
 */
public class Stack {
    /** 現在の要素数 */
    private int count;

    /**
     * スタックに要素を一つ入れる
     * @param スタックに入れる要素
     */
    public void push(int item) { ... }

    /**
     * スタックから一番上の要素を取り出す
     * @return スタックの一番上の要素を返す。スタックが空の場合は{@code null}を返す。
     */
    public int pop() { ... }
}
```