コーディング規約

目次

コーディング規約

1. はじめに

本書は、総合演習である電子商取引システム開発において、Java 言語でコーディングをする際のルールや指針を示すものです。

本書の目的は、プロジェクトチーム全体で同じルール、指針を共有し、メンテナンス性を重視した、読みやすいプログラムコードを実現し、チーム内やレビュー担当者とのコミュニケーションを円滑にすることです。

2. ソースファイル

2.1. Java ソースファイル名

Java ソースファイルのファイル名は、**トップレベルのクラス名に拡張子[.java]を加えたもの**にしてください。

例：public class TableListView のファイル名は、TableListView.java にする。

2.2. JSP ファイル名

JSP ファイルのファイル名は、**すべて小文字とし、単語の区切りをアンダースコア「 _ 」**にしてください。特に**Servlet クラスを forward した先となる JSP ファイル**の場合、**Servlet クラスを連想できるような名前**にして、**拡張子[.jsp]を加えた**ものにしてください。

例：header.jsp、user_body.jsp

例：TableListViewServlet クラスの forward 先となる JSP ファイルの場合
table_list_view.jsp

2.3. ファイルの文字コード（UTF-8）

ソースファイルの文字コードは **UTF-8** にしてください。

2.4. ソースコードの整形

ソースコードは、**プロジェクトで使用する統合開発環境における、標準で設定されているコードスタイルのフォーマッターに準じて、整形する**ようにしてください。

例：

- 演算子や制御構文の両サイドに半角空白を 1 つ入れる

```
for (int i = 0; i < listArray.size(); i++) {
        System.out.println(listArray.get(i));
}
```

- 中括弧は宣言と同じ行を開始位置にする

```
if (this.isExit() == true) { //  中括弧の開始は宣言と同じ行にする
}
```

特に、Eclipse を利用している場合、**Eclipse の書式のフォーマット機能（Ctrl + Shift + F）を活用**してください。

3. 命名規則

3.1. パッケージ名

パッケージ名は、すべて**小文字**にしてください。また、特別な場合を除き、**記号やダラー「 $ 」、アンダースコア「 _ 」などは使用しない**でください。

例：com.bh.ecsite

パッケージ名は、ユニークなパッケージ名とするため、**組織のトップレベルドメイン名とサブドメインリストを逆順にして始め、その後に、システム名や、パッケージに含まれるソースコードの意味を表す単語を付与**してください。

例：トップレベルドメインが bh.com で、サブドメインが ecsite の場合、パッケージ名は com.bh.ecsite で始める。

例：システム名が xyz で DAO クラスを含むパッケージ名は com.bh.ecsite.xyz.dao とする。

3.1.1. パッケージに含まれるソースコードの意味を表す単語

パッケージに含まれるソースコードの意味を表す単語は次のようにしてください。

DAO クラスを含むパッケージ：dao
DTO クラスを含むパッケージ：DTO
Service クラスを含むパッケージ：service
Servlet クラスを含むパッケージ：servlet
Filter クラスを含むパッケージ：filter
ユーティリティクラス（static な共通処理をまとめたクラス）を含むパッケージ：util

その他のクラスについては、クラスのカテゴリを意味する小文字にしてください。単語から**クラスのカテゴリが想像しにくいような単語は控える**ようにしてください。（others などカテゴリが想像しにくい単語は控える）

例：データベースの接続を管理する
コネクションマネージャクラスを含むパッケージ：database

3.2. クラス名とインターフェース名

クラス名は、**UpperCamelCase（先頭を大文字とし、単語の区切りを大文字）**にしてください。**記号やダラー「 $ 」、ハイフン「 - 」、アンダースコア「 _ 」は使用しない**でください。

例：テーブルリストを表示する（tabel list view）クラスの場合、クラス名は TableListView とする。

3.2.1. DAO クラス名

DAO クラスを表すクラス名は、**末尾を DAO** にしてください。

例：TableListViewDAO

3.2.2. DTO クラス名

DTO クラスを表すクラス名は、**末尾を DTO** にしてください。

例：TableListViewDTO

3.2.3. Service クラス名

Service クラスを表すクラス名は、**末尾を Service** にしてください。

例：TableListViewService

3.2.4. Servlet クラス名

Servlet クラスを表すクラス名は、**末尾を Servlet** にしてください。

例：TableListViewServlet

3.2.5. Filter クラス名

Filter クラスを表すクラス名は、**末尾を Filter** にしてください。

例：TableListViewFilter

3.2.6. ユーティリティクラス名

ユーティリティクラス（共通処理を 1 か所にまとめるためのクラス）を表すクラス名は、**末尾を Util** にしてください。

例：TableListViewUtil

3.2.7. インターフェース名

先頭に I を付けて、以降はクラス名と同様にしてください。

例：テーブルリストを表示する（tabel list view）ことを意味するインターフェースの場合 ITableListView とする。

また、クラスに能力を加えるための mix-in として利用させたい場合、**その能力を形容詞とし、末尾を able** にしてください。

例：Runnable、Serializable など

3.2.8. 抽象クラス名

先頭に Abstract を付けて、実装するサブクラス名を連想させる名前にしてください。

例：TableListView クラス、ImageListView クラスで実装をする抽象クラスの場合 AbstractListView とする。

3.2.9. 例外クラス名

例外クラスは、独自に作成しないようにしてください。やむを得ず例外クラスを独自に作成することとなった場合、**末尾を Exception** にしてください。

3.2.10. テストクラス名

ホワイトボックステストを実施するためのテストクラスを作成する場合は、**テスト対象のクラス名の末尾に Test を加えて**ください。

例：TableListViewService クラスをテストするためのテストクラス名は TableListViewServiceTest とする。

3.3. メソッド名

メソッド名は、**lowerCamelCase（先頭を小文字とし、単語の区切りを大文字）**にしてください。記号やハイフン「 - 」、アンダースコア「 _ 」は使用しないでください。

例：メッセージを送信するメソッドの場合 sendMessage とする。
停止をするメソッドの場合 stop とする。

3.4. 定数名

定数名（static final）は、**すべて大文字とし、単語の区切りをアンダースコア「_」**にしてください。

例：

```
static final int NUMBER = 5;
static final String ERROR_MESSAGE = “エラーメッセージ”;
```

※修飾子が static のみ、もしくは final のみの場合、定数とならないため、後述する定数でないフィールド変数を参照してください。

3.5. （定数でない）フィールド変数名

（定数でない）フィールド変数名は、**lowerCamelCase（先頭を小文字とし、単語の区切りを大文字）**にしてください。記号やハイフン「 - 」、アンダースコア「 _ 」は使用しないでください。

3.6. パラメータ変数名

コンストラクタやメソッドに渡すパラメータの変数名は、**lowerCamelCase（先頭を小文字とし、単語の区切りを大文字）**にしてください。記号やハイフン「 - 」、アンダースコア「 _ 」は使用しないでください。

例：

```
private void updateTableView(int tableViewId ,  TableViewDTO tableViewDTO) {
    ～処理～
}
```

3.7. ローカル変数名

ローカル変数名は、**lowerCamelCase（先頭を小文字とし、単語の区切りを大文字）**にしてください。記号やハイフン「 - 」、アンダースコア「 _ 」は使用しないでください。

ただし、変数のスコープが狭いループカウンタなどには、単純な名前を利用してもよいこととします。

例：for 文の場合

```
for(int i = 0;i < xArray.length; i++) {
  for(int j = 0;j < yArray.length; j++) {
    ～処理～
  }
}
```

例：拡張 for 文の場合

```
List<String> strList = this.getStrList();
for(String s：strList) {
    ～処理～
}
List< TableViewDTO > tableViewList = this.getTableViewList();
for(TableViewDTO list：tableViewList) {
    ～処理～
}
```

以上