

|  |
| --- |
| **PPCBank CoreBanking System**  **Developer’s Guide**  **PPCB\_DEV v0.1.0** |

**Mar 27, 2017**



Innovative Mobile & Fintech Solutions, Ready to Provide

**Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Description | Date | Author | Approver |
| 0.0.1 | Drafted | 2018/02/09 | Lee, DaeRyung | Kim, Byungjin |
| 0.0.2 | Business service implementation guide | 2018/03/08 | Lee, DaeRyung |  |
| 0.1.0 | Fully revised | 2018/03/26 | Lee, DaeRyung |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Agenda**

[1. Introduction 1](#_Toc510179619)

[1.1. Introduction 1](#_Toc510179620)

[1.2. Scope 1](#_Toc510179621)

[2. Naming Standard 2](#_Toc510179622)

[2.1. Service ID 2](#_Toc510179623)

[2.2. Package 2](#_Toc510179624)

[2.3. Class 3](#_Toc510179625)

[2.4. Method 3](#_Toc510179626)

[2.5. Local variable 4](#_Toc510179627)

[2.6. Constant 4](#_Toc510179628)

[2.7. Input parameter 4](#_Toc510179629)

[2.8. Exception 5](#_Toc510179630)

[3. Development Environment 6](#_Toc510179631)

[3.1. Overview 6](#_Toc510179632)

[3.2. Eclipse for Server-side implementation 6](#_Toc510179633)

[3.2.1. Setting 6](#_Toc510179634)

[3.2.2. For Java Cleanup 6](#_Toc510179635)

[3.2.3. For Java Formatter 7](#_Toc510179636)

[3.2.4. For Java Code Template 12](#_Toc510179637)

[4. Business Service Development 14](#_Toc510179638)

[4.1. Introduction 14](#_Toc510179639)

[4.1.1. Understanding on e-link solution 14](#_Toc510179640)

[4.1.2. Understanding on xframe UI framework 14](#_Toc510179641)

[4.1.3. Understanding on basic structure of MARU framework 14](#_Toc510179642)

[4.2. Business service implementation steps 16](#_Toc510179643)

[4.2.1. Message definition 16](#_Toc510179644)

[4.2.2. Client-side logic and UI development 16](#_Toc510179645)

[4.2.3. Server-side logic development 16](#_Toc510179646)

[4.3. Business Logic Implementation with MARU framework 17](#_Toc510179647)

[4.3.1. CPBI/CPBC 17](#_Toc510179648)

[4.3.2. EBI/EBC 23](#_Toc510179649)

[4.3.3. PBI/PBC 23](#_Toc510179650)

[5. Coding Standard 31](#_Toc510179651)

[5.1. Introduction 31](#_Toc510179652)

[5.1.1. Font 31](#_Toc510179653)

[5.2. Structure 31](#_Toc510179654)

[5.2.1. Java source file 31](#_Toc510179655)

[5.2.2. Class 31](#_Toc510179656)

[5.3. Comment 31](#_Toc510179657)

[5.3.1. Common 31](#_Toc510179658)

[5.3.2. File 32](#_Toc510179659)

[5.3.3. Class 32](#_Toc510179660)

[5.3.4. Member variables 32](#_Toc510179661)

[5.3.5. Member method 33](#_Toc510179662)

[5.3.6. SQL Statement 34](#_Toc510179663)

[5.3.7. Etc 38](#_Toc510179664)

[5.4. Code 38](#_Toc510179665)

[5.4.1. Indentation and Spacing 38](#_Toc510179666)

[5.4.2. File ( Package & import) 40](#_Toc510179667)

[5.4.3. Class 40](#_Toc510179668)

[5.4.4. Class method 41](#_Toc510179669)

[5.4.5. Variable 41](#_Toc510179670)

[5.4.6. Statement 42](#_Toc510179671)

[5.4.7. Expression 45](#_Toc510179672)

[5.4.8. Etc 46](#_Toc510179673)

[5.5. Example 46](#_Toc510179674)

[5.5.1. Example code 46](#_Toc510179675)

# 

# Introduction

## Introduction

This document is for unifying programming style between developers.

## Scope

* Naming standards for developing
* Java coding standard

# Naming Standard

## Service ID

Business Code(3) + Service Category Code(2) + Sequence for a Category Code(4) + Function Type(1) + Sequence for a Function(1)

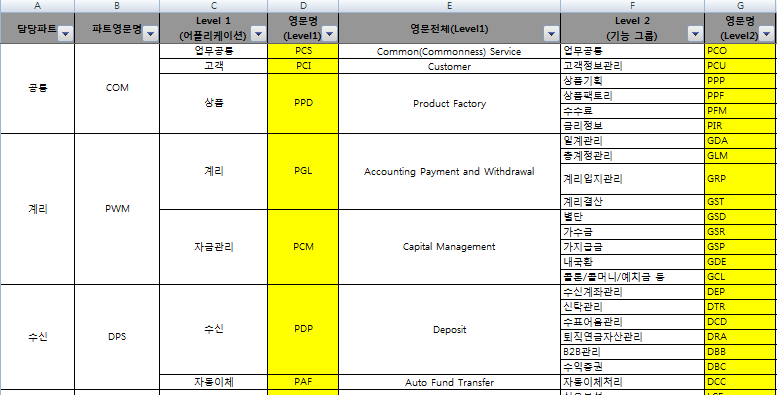
|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | | **Length** | **Description** |
| Business Code | | 3 | Component level code which is defined in … document |
| Transaction Code | Service Category Code | 2 | Level 1 code in 2 digits |
| Sequence for a Category Code | 4 | Sequence code |
| Detail Code | Function Type | 1 | 1 : select  2 : CUD (create, update, delete) |
| Sequence for a Function | 1 | 1~9, A~Z : cases of multi transaction in single service |

## Package

Packages are created with a combination of acronyms based on the terminology dictionary, all in lower case. Create a package with the following structure with reference to Level 1 and 2 of detailed business area code.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **D1** | **D2** | **D3** | **D4** | **D5** | **File** |
| PBC | ppcb | apsara | Level1 Code | Level2 Code | pbi | BusinessNamePbc (n) |
| CPBC | ppcb | apsara | Level1 Code | Level2 Code | cpbi | BusinessNameCpbc (n) |
| EBC | ppcb | apsara | Level1 Code | Level2 Code | ebi | EntityNameEbc (n) |
| Constant | ppcb | apsara | Level1 Code | Level2 Code | cti | FileNameCtc (n) |
| Center-Cut | ppcb | apsara | Level1 Code | Level2 Code | cci | BusinessNameCcc (n) |
| Batch | ppcb | apsara | Level1 Code | Level2 Code | bbi | BusinessNameBbc (n) |
| Deferred | ppcb | apsara | Level1 Code | Level2 Code | dfi | BusinessNameDfc (n) |

In case of package level N + 1 of PBC / CPBC / batch / default, several subpackages can be created depending on the task.



## Class

The name of the class must be a noun, with the first letter capitalized.

If there is a meaningful word in the middle, the beginning of the word is capitalized, and all other characters are in lower case. Pascal Case is used as the notation, and the word uses the word defined in the meta system.

(Note Pascal Case - Capitalize the first letter of each word in compound words without multiple spaces, eg BrnInfoOfrCpbc)

Use suffixes suffixed with suffix for each class as follows.

ex : BrnInfoOfr**Cpbc**

|  |  |
| --- | --- |
| **구분** | **클래스 접미어** |
| PBC | Pbc |
| CPBC | Cpbc |
| EBC | Ebc |
| Constant | Const |
| Centercut | Ccc |
| Deferred | Dfc |
| Batch | Bbc |

Note) The center cut line / post-treatment class is located in the center cut package

## Method

The name of the method must be a verb, the first letter is a lowercase letter, and begins with a word defined in the verb dictionary (modeling). If there is a meaningful word in the middle, the beginning of the word is capitalized, and all other characters are in lower case. It is used in Camel Case notation. Words are written in a combination of standard items or standard words registered in the meta system. (In the main Camel Case - Pascal Case, write the first letter in lowercase instead of uppercase, eg retrieveLastTrstNoGvno)

- public MData retrieveLastTrstNoGvno(MData gvnoInqTrmsEBI) throws LException

## Local variable

Variables use only lowercase letters and are used as words or compound words according to variable naming conventions. Class creation variables (PBC, EBC, etc.) also follow variable naming conventions. For complex words, use Camel Case notation.

Variables are created with standard items or standard word combinations registered in the meta system, and the type abbreviation is attached at the beginning.

- Type abbreviation

|  |  |
| --- | --- |
| **Type** | **약어** |
| String | s |
| Boolean | b |
| BigDecimal | bd |
| Long | l |
| int | i |
| DTO type (MData) | t |
| Input parameter (MData) | ip |
| Return parameter (MData) | rp |

- MData **tPaging** = new MData();

- BigDecimal **bdAccount**;

## Constant

All names of variables declared as class constants are capitalized and "\_" is used in the middle of a meaningful word.

Note) Constants are used only in common, and do not use constants in general business classes. When defining an Enum type, the class name follows the definition in 2.4.

- static final int **MIN\_WIDTH** = 4;

- public enum TopupChannels {

ATM (“6011”),

TELLER(“6012”),

…

}

## Input parameter

The name of the method's input parameter is the same as the local variable naming convention.

## Exception

In the try ~ catch () syntax, variable names for Exception in the catch statement follow naming conventions for Exception as follows.

- LBizException **lbe**

- DevonException **de**

- LInterfaceException **lie**

- LDuplicateException **lde**

# Development Environment

## Overview

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Level 1 | Level 2 | Level 3 | Description | Remark |
| Software | DB | name | PostgreSQL (Community Version) |  |
| url | https://www.postgresql.org/download/ |  |
| version | 10.1 |  |
| WAS | name | JBoss EAP | <http://www.opennaru.com/>  jboss/jboss-eap-vs-wildfly/ |
| url | https://developers.redhat.com/products/eap |  |
| version | 7.1.0 |  |
| Java | name | Oracle Java |  |
| url | <http://www.oracle.com/technetwork/>  java/javase/downloads/jdk8-downloads-2133151.html |  |
| Version | Java SE Development Kit 8u162 |  |
| Eclipse | name | Eclipse EE |  |
| url | http://www.eclipse.org/downloads/eclipse-packages/ |  |
| version | Eclipse Oxygen (4.7.2) |  |
| SVN Plugin | Subversive - SVN Team Provider 4.0.5 |  |
| JBoss Plugin | JBoss Tools 4.5.1.Final |  |
| DB Modeling Tool | name | DA# 5 |  |
| url | http://dataware.kr/da5/?#about5 |  |
| Jenkins | name | Jenkins war |  |
| url | https://jenkins.io/download/ |  |
| version | 2.89.3 |  |
| DBMS (temporary)      For Framework | IP:port | 52.76.249.241:54329 |  |
| superuser ID/PW | postgres / ppcb2018 | Don't use this. |
| pgDevOps | http://localhost:8051 |  |
| ksshin@mcnc.co.kr / mcnc2018 |  |
| DB Name | mrfwdb | tablespace : ts\_maru\_dt, ts\_maru\_ix |
| ID/PW | maruusr / Mcnc@23! |  |
| SVN  DB Modeling Tool | url | http://52.76.249.241:9963/svn/maru | temp ID/pw : mcnc / mcnc!@# |
| name | DA# 5 |  |
| url | http://dataware.kr/da5/?#about5 |  |

## Eclipse for Server-side implementation

### Setting

Target File : {installed directory}\eclipse.ini

Add the following to the last line;

|  |
| --- |
| -Dfile.encoding=utf-8  -Duser.name=Your Name or Nick |

### For Java Cleanup

* run eclipse
* Move Window -> Preferences > java > Code Style > Clean Up
* Click “import” button to import cleanup rule for this project

File : JAVA\_CleanUp\_For\_PPCB\_Core\_Banking\_v0.1.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" standalone="no"?>  <profiles version="2">  <profile kind="CleanUpProfile" name="JAVA\_CleanUp\_For\_PPCB\_Core\_Banking" version="2">  <setting id="cleanup.qualify\_static\_method\_accesses\_with\_declaring\_class" value="false"/>  <setting id="cleanup.always\_use\_this\_for\_non\_static\_method\_access" value="false"/>  <setting id="cleanup.organize\_imports" value="false"/>  <setting id="cleanup.remove\_trailing\_whitespaces\_ignore\_empty" value="false"/>  <setting id="cleanup.format\_source\_code\_changes\_only" value="false"/>  <setting id="cleanup.qualify\_static\_field\_accesses\_with\_declaring\_class" value="false"/>  <setting id="cleanup.add\_generated\_serial\_version\_id" value="true"/>  <setting id="cleanup.qualify\_static\_member\_accesses\_through\_subtypes\_with\_declaring\_class" value="true"/>  <setting id="cleanup.remove\_redundant\_type\_arguments" value="true"/>  <setting id="cleanup.remove\_unused\_imports" value="true"/>  <setting id="cleanup.insert\_inferred\_type\_arguments" value="false"/>  <setting id="cleanup.make\_private\_fields\_final" value="true"/>  <setting id="cleanup.use\_lambda" value="true"/>  <setting id="cleanup.always\_use\_blocks" value="true"/>  <setting id="cleanup.use\_this\_for\_non\_static\_field\_access\_only\_if\_necessary" value="true"/>  <setting id="cleanup.sort\_members\_all" value="false"/>  <setting id="cleanup.remove\_trailing\_whitespaces\_all" value="true"/>  <setting id="cleanup.add\_missing\_annotations" value="true"/>  <setting id="cleanup.always\_use\_this\_for\_non\_static\_field\_access" value="false"/>  <setting id="cleanup.make\_parameters\_final" value="false"/>  <setting id="cleanup.sort\_members" value="false"/>  <setting id="cleanup.remove\_private\_constructors" value="true"/>  <setting id="cleanup.always\_use\_parentheses\_in\_expressions" value="false"/>  <setting id="cleanup.remove\_unused\_local\_variables" value="false"/>  <setting id="cleanup.convert\_to\_enhanced\_for\_loop" value="false"/>  <setting id="cleanup.remove\_unused\_private\_fields" value="true"/>  <setting id="cleanup.never\_use\_blocks" value="false"/>  <setting id="cleanup.add\_missing\_deprecated\_annotations" value="true"/>  <setting id="cleanup.use\_this\_for\_non\_static\_field\_access" value="false"/>  <setting id="cleanup.remove\_unnecessary\_nls\_tags" value="true"/>  <setting id="cleanup.qualify\_static\_member\_accesses\_through\_instances\_with\_declaring\_class" value="true"/>  <setting id="cleanup.add\_missing\_nls\_tags" value="false"/>  <setting id="cleanup.remove\_unnecessary\_casts" value="true"/>  <setting id="cleanup.use\_blocks\_only\_for\_return\_and\_throw" value="false"/>  <setting id="cleanup.format\_source\_code" value="true"/>  <setting id="cleanup.convert\_functional\_interfaces" value="true"/>  <setting id="cleanup.add\_default\_serial\_version\_id" value="false"/>  <setting id="cleanup.remove\_unused\_private\_methods" value="true"/>  <setting id="cleanup.remove\_trailing\_whitespaces" value="true"/>  <setting id="cleanup.make\_type\_abstract\_if\_missing\_method" value="false"/>  <setting id="cleanup.add\_serial\_version\_id" value="true"/>  <setting id="cleanup.use\_this\_for\_non\_static\_method\_access" value="false"/>  <setting id="cleanup.use\_this\_for\_non\_static\_method\_access\_only\_if\_necessary" value="true"/>  <setting id="cleanup.use\_anonymous\_class\_creation" value="false"/>  <setting id="cleanup.add\_missing\_override\_annotations\_interface\_methods" value="true"/>  <setting id="cleanup.remove\_unused\_private\_members" value="false"/>  <setting id="cleanup.make\_local\_variable\_final" value="true"/>  <setting id="cleanup.add\_missing\_methods" value="false"/>  <setting id="cleanup.never\_use\_parentheses\_in\_expressions" value="true"/>  <setting id="cleanup.qualify\_static\_member\_accesses\_with\_declaring\_class" value="true"/>  <setting id="cleanup.use\_parentheses\_in\_expressions" value="false"/>  <setting id="cleanup.add\_missing\_override\_annotations" value="true"/>  <setting id="cleanup.use\_blocks" value="true"/>  <setting id="cleanup.make\_variable\_declarations\_final" value="false"/>  <setting id="cleanup.correct\_indentation" value="true"/>  <setting id="cleanup.remove\_unused\_private\_types" value="true"/>  </profile>  </profiles> |

### For Java Formatter

* run eclipse
* Move to Window -> Preferences > java > Code Style > Formatter
* Click “import” button to import formatter definition for this project

File : JAVA\_Formatter\_For\_PPCB\_Core\_Banking\_v0.2.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" standalone="no"?>  <profiles version="13">  <profile kind="CodeFormatterProfile" name="JAVA\_Formatter\_For\_PPCB\_Core\_Banking" version="13">  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_ellipsis" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_comma\_in\_enum\_declarations" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_new\_line\_in\_empty\_annotation\_declaration" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_comma\_in\_allocation\_expression" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_at\_in\_annotation\_type\_declaration" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.parentheses\_positions\_in\_for\_statment" value="common\_lines"/>  <setting id="org.eclipse.jdt.core.formatter.comment.new\_lines\_at\_block\_boundaries" value="true"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_comma\_in\_constructor\_declaration\_parameters" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.comment.insert\_new\_line\_for\_parameter" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_new\_line\_after\_annotation\_on\_package" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.parentheses\_positions\_in\_method\_invocation" value="common\_lines"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_between\_empty\_parens\_in\_enum\_constant" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.blank\_lines\_after\_imports" value="1"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_closing\_paren\_in\_while" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.comment.insert\_new\_line\_before\_root\_tags" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_between\_empty\_parens\_in\_annotation\_type\_member\_declaration" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_comma\_in\_method\_declaration\_throws" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.parentheses\_positions\_in\_switch\_statement" value="common\_lines"/>  <setting id="org.eclipse.jdt.core.formatter.comment.format\_javadoc\_comments" value="false"/>  <setting id="org.eclipse.jdt.core.formatter.indentation.size" value="4"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_postfix\_operator" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.parentheses\_positions\_in\_enum\_constant\_declaration" value="common\_lines"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_comma\_in\_for\_increments" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_comma\_in\_type\_arguments" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_comma\_in\_for\_inits" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_new\_line\_in\_empty\_anonymous\_type\_declaration" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_semicolon\_in\_for" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.disabling\_tag" value="@formatter:off"/>  <setting id="org.eclipse.jdt.core.formatter.continuation\_indentation" value="2"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_enum\_constants" value="0"/>  <setting id="org.eclipse.jdt.core.formatter.blank\_lines\_before\_imports" value="0"/>  <setting id="org.eclipse.jdt.core.formatter.blank\_lines\_after\_package" value="1"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_binary\_operator" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_comma\_in\_multiple\_local\_declarations" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.parentheses\_positions\_in\_if\_while\_statement" value="common\_lines"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_arguments\_in\_enum\_constant" value="16"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_opening\_angle\_bracket\_in\_parameterized\_type\_reference" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.comment.indent\_root\_tags" value="true"/>  <setting id="org.eclipse.jdt.core.formatter.wrap\_before\_or\_operator\_multicatch" value="true"/>  <setting id="org.eclipse.jdt.core.formatter.enabling\_tag" value="@formatter:on"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_closing\_brace\_in\_block" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.comment.count\_line\_length\_from\_starting\_position" value="false"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_parenthesized\_expression\_in\_return" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_throws\_clause\_in\_method\_declaration" value="16"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_new\_line\_after\_annotation\_on\_parameter" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.keep\_then\_statement\_on\_same\_line" value="false"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_new\_line\_after\_annotation\_on\_field" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_comma\_in\_explicitconstructorcall\_arguments" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_new\_line\_in\_empty\_block" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_prefix\_operator" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.blank\_lines\_between\_type\_declarations" value="1"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_closing\_brace\_in\_array\_initializer" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_closing\_paren\_in\_for" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_opening\_paren\_in\_catch" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_opening\_angle\_bracket\_in\_type\_arguments" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_new\_line\_after\_annotation\_on\_method" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_opening\_paren\_in\_switch" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_parameterized\_type\_references" value="0"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_brace\_in\_anonymous\_type\_declaration" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_closing\_paren\_in\_parenthesized\_expression" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_new\_line\_after\_annotation\_on\_enum\_constant" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.never\_indent\_line\_comments\_on\_first\_column" value="false"/>  <setting id="org.eclipse.jdt.core.compiler.problem.enumIdentifier" value="error"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_and\_in\_type\_parameter" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_comma\_in\_for\_inits" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.indent\_statements\_compare\_to\_block" value="true"/>  <setting id="org.eclipse.jdt.core.formatter.brace\_position\_for\_anonymous\_type\_declaration" value="end\_of\_line"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_question\_in\_wildcard" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_paren\_in\_annotation" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_comma\_in\_method\_invocation\_arguments" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_brace\_in\_switch" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.comment.line\_length" value="120"/>  <setting id="org.eclipse.jdt.core.formatter.use\_on\_off\_tags" value="false"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_between\_empty\_brackets\_in\_array\_allocation\_expression" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_brace\_in\_enum\_constant" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_between\_empty\_parens\_in\_method\_invocation" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_assignment\_operator" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_brace\_in\_type\_declaration" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_opening\_paren\_in\_for" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.comment.preserve\_white\_space\_between\_code\_and\_line\_comments" value="false"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_new\_line\_after\_annotation\_on\_local\_variable" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.brace\_position\_for\_method\_declaration" value="end\_of\_line"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_closing\_paren\_in\_method\_invocation" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_union\_type\_in\_multicatch" value="16"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_colon\_in\_for" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.number\_of\_blank\_lines\_at\_beginning\_of\_method\_body" value="1"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_closing\_angle\_bracket\_in\_type\_arguments" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.keep\_else\_statement\_on\_same\_line" value="false"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_binary\_expression" value="16"/>  <setting id="org.eclipse.jdt.core.formatter.parentheses\_positions\_in\_catch\_clause" value="common\_lines"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_comma\_in\_parameterized\_type\_reference" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_comma\_in\_array\_initializer" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_comma\_in\_multiple\_field\_declarations" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_comma\_in\_annotation" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_arguments\_in\_explicit\_constructor\_call" value="16"/>  <setting id="org.eclipse.jdt.core.formatter.indent\_body\_declarations\_compare\_to\_annotation\_declaration\_header" value="true"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_comma\_in\_superinterfaces" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_colon\_in\_default" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_question\_in\_conditional" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.brace\_position\_for\_block" value="end\_of\_line"/>  <setting id="org.eclipse.jdt.core.formatter.brace\_position\_for\_constructor\_declaration" value="end\_of\_line"/>  <setting id="org.eclipse.jdt.core.formatter.brace\_position\_for\_lambda\_body" value="end\_of\_line"/>  <setting id="org.eclipse.jdt.core.formatter.compact\_else\_if" value="true"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_comma\_in\_type\_parameters" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_paren\_in\_catch" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_paren\_in\_method\_invocation" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.put\_empty\_statement\_on\_new\_line" value="false"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_parameters\_in\_constructor\_declaration" value="16"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_type\_parameters" value="0"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_comma\_in\_method\_invocation\_arguments" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_arguments\_in\_method\_invocation" value="16"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_throws\_clause\_in\_constructor\_declaration" value="16"/>  <setting id="org.eclipse.jdt.core.compiler.problem.assertIdentifier" value="error"/>  <setting id="org.eclipse.jdt.core.formatter.comment.clear\_blank\_lines\_in\_block\_comment" value="true"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_new\_line\_before\_catch\_in\_try\_statement" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_paren\_in\_try" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_new\_line\_at\_end\_of\_file\_if\_missing" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.comment.clear\_blank\_lines\_in\_javadoc\_comment" value="false"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_comma\_in\_array\_initializer" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_binary\_operator" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_unary\_operator" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_expressions\_in\_array\_initializer" value="16"/>  <setting id="org.eclipse.jdt.core.formatter.format\_line\_comment\_starting\_on\_first\_column" value="true"/>  <setting id="org.eclipse.jdt.core.formatter.number\_of\_empty\_lines\_to\_preserve" value="0"/>  <setting id="org.eclipse.jdt.core.formatter.parentheses\_positions\_in\_annotation" value="common\_lines"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_colon\_in\_case" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_ellipsis" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_semicolon\_in\_try\_resources" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_colon\_in\_assert" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_closing\_paren\_in\_if" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_comma\_in\_type\_arguments" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_and\_in\_type\_parameter" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_new\_line\_in\_empty\_type\_declaration" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_opening\_paren\_in\_parenthesized\_expression" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.comment.format\_line\_comments" value="false"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_colon\_in\_labeled\_statement" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.align\_type\_members\_on\_columns" value="false"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_assignment" value="0"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_module\_statements" value="16"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_new\_line\_in\_empty\_method\_body" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.indent\_body\_declarations\_compare\_to\_type\_header" value="true"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_between\_empty\_parens\_in\_method\_declaration" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_opening\_paren\_in\_enum\_constant" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_superinterfaces\_in\_type\_declaration" value="64"/>  <setting id="org.eclipse.jdt.core.formatter.blank\_lines\_before\_first\_class\_body\_declaration" value="1"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_conditional\_expression" value="48"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_new\_line\_before\_closing\_brace\_in\_array\_initializer" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_comma\_in\_constructor\_declaration\_parameters" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.format\_guardian\_clause\_on\_one\_line" value="false"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_paren\_in\_if" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_new\_line\_after\_annotation\_on\_type" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_brace\_in\_block" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.brace\_position\_for\_enum\_declaration" value="end\_of\_line"/>  <setting id="org.eclipse.jdt.core.formatter.brace\_position\_for\_block\_in\_case" value="end\_of\_line"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_closing\_paren\_in\_constructor\_declaration" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.comment.format\_header" value="false"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_arguments\_in\_allocation\_expression" value="16"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_opening\_paren\_in\_method\_invocation" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_paren\_in\_while" value="insert"/>  <setting id="org.eclipse.jdt.core.compiler.codegen.inlineJsrBytecode" value="enabled"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_closing\_paren\_in\_switch" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_method\_declaration" value="0"/>  <setting id="org.eclipse.jdt.core.formatter.join\_wrapped\_lines" value="true"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_between\_empty\_parens\_in\_constructor\_declaration" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.wrap\_before\_conditional\_operator" value="true"/>  <setting id="org.eclipse.jdt.core.formatter.indent\_switchstatements\_compare\_to\_cases" value="true"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_closing\_bracket\_in\_array\_allocation\_expression" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_closing\_paren\_in\_synchronized" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.align\_fields\_grouping\_blank\_lines" value="2147483647"/>  <setting id="org.eclipse.jdt.core.formatter.comment.new\_lines\_at\_javadoc\_boundaries" value="true"/>  <setting id="org.eclipse.jdt.core.formatter.brace\_position\_for\_annotation\_type\_declaration" value="end\_of\_line"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_colon\_in\_for" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_resources\_in\_try" value="80"/>  <setting id="org.eclipse.jdt.core.formatter.use\_tabs\_only\_for\_leading\_indentations" value="false"/>  <setting id="org.eclipse.jdt.core.formatter.parentheses\_positions\_in\_try\_clause" value="common\_lines"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_selector\_in\_method\_invocation" value="16"/>  <setting id="org.eclipse.jdt.core.formatter.never\_indent\_block\_comments\_on\_first\_column" value="false"/>  <setting id="org.eclipse.jdt.core.compiler.source" value="9"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_opening\_paren\_in\_synchronized" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_comma\_in\_constructor\_declaration\_throws" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.tabulation.size" value="4"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_new\_line\_in\_empty\_enum\_constant" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_comma\_in\_allocation\_expression" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_opening\_bracket\_in\_array\_reference" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_colon\_in\_conditional" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.comment.format\_source\_code" value="true"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_brace\_in\_array\_initializer" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_closing\_paren\_in\_try" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_semicolon\_in\_try\_resources" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.blank\_lines\_before\_field" value="0"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_at\_in\_annotation" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.continuation\_indentation\_for\_array\_initializer" value="2"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_question\_in\_wildcard" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.blank\_lines\_before\_method" value="1"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_superclass\_in\_type\_declaration" value="64"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_superinterfaces\_in\_enum\_declaration" value="64"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_parenthesized\_expression\_in\_throw" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.wrap\_before\_assignment\_operator" value="false"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_colon\_in\_labeled\_statement" value="insert"/>  <setting id="org.eclipse.jdt.core.compiler.codegen.targetPlatform" value="9"/>  <setting id="org.eclipse.jdt.core.formatter.brace\_position\_for\_switch" value="end\_of\_line"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_comma\_in\_superinterfaces" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_comma\_in\_method\_declaration\_parameters" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_new\_line\_after\_type\_annotation" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_opening\_brace\_in\_array\_initializer" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_paren\_in\_parenthesized\_expression" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.comment.format\_html" value="true"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_at\_in\_annotation\_type\_declaration" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_closing\_angle\_bracket\_in\_type\_parameters" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.parentheses\_positions\_in\_method\_delcaration" value="common\_lines"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_compact\_if" value="52"/>  <setting id="org.eclipse.jdt.core.formatter.indent\_empty\_lines" value="true"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_type\_arguments" value="0"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_comma\_in\_parameterized\_type\_reference" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_unary\_operator" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_paren\_in\_enum\_constant" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_arguments\_in\_annotation" value="0"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_comma\_in\_enum\_declarations" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.keep\_empty\_array\_initializer\_on\_one\_line" value="false"/>  <setting id="org.eclipse.jdt.core.formatter.indent\_switchstatements\_compare\_to\_switch" value="true"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_new\_line\_before\_else\_in\_if\_statement" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_assignment\_operator" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_paren\_in\_constructor\_declaration" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.blank\_lines\_before\_new\_chunk" value="0"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_new\_line\_after\_label" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.indent\_body\_declarations\_compare\_to\_enum\_declaration\_header" value="true"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_opening\_bracket\_in\_array\_allocation\_expression" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_opening\_paren\_in\_constructor\_declaration" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_colon\_in\_conditional" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_angle\_bracket\_in\_parameterized\_type\_reference" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_comma\_in\_method\_declaration\_parameters" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_closing\_angle\_bracket\_in\_type\_arguments" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_closing\_paren\_in\_cast" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_colon\_in\_assert" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.blank\_lines\_before\_member\_type" value="1"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_new\_line\_before\_while\_in\_do\_statement" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_bracket\_in\_array\_type\_reference" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_closing\_angle\_bracket\_in\_parameterized\_type\_reference" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_arguments\_in\_qualified\_allocation\_expression" value="16"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_new\_line\_after\_opening\_brace\_in\_array\_initializer" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_new\_line\_in\_empty\_enum\_declaration" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.indent\_breaks\_compare\_to\_cases" value="true"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_closing\_paren\_in\_method\_declaration" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_opening\_paren\_in\_if" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_semicolon" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_postfix\_operator" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_opening\_paren\_in\_try" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_angle\_bracket\_in\_type\_arguments" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_opening\_paren\_in\_cast" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.comment.format\_block\_comments" value="false"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_lambda\_arrow" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_opening\_paren\_in\_method\_declaration" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.keep\_imple\_if\_on\_one\_line" value="false"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_brace\_in\_enum\_declaration" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_parameters\_in\_method\_declaration" value="16"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_between\_brackets\_in\_array\_type\_reference" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_angle\_bracket\_in\_type\_parameters" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_semicolon\_in\_for" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_comma\_in\_method\_declaration\_throws" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_bracket\_in\_array\_allocation\_expression" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.indent\_statements\_compare\_to\_body" value="true"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_multiple\_fields" value="16"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_comma\_in\_enum\_constant\_arguments" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_prefix\_operator" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.brace\_position\_for\_array\_initializer" value="end\_of\_line"/>  <setting id="org.eclipse.jdt.core.formatter.wrap\_before\_binary\_operator" value="true"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_brace\_in\_method\_declaration" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_comma\_in\_type\_parameters" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_closing\_paren\_in\_catch" value="insert"/>  <setting id="org.eclipse.jdt.core.compiler.compliance" value="9"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_closing\_bracket\_in\_array\_reference" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_comma\_in\_annotation" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_comma\_in\_enum\_constant\_arguments" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.parentheses\_positions\_in\_lambda\_declaration" value="common\_lines"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_between\_empty\_braces\_in\_array\_initializer" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_colon\_in\_case" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_comma\_in\_multiple\_local\_declarations" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_brace\_in\_annotation\_type\_declaration" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_bracket\_in\_array\_reference" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_paren\_in\_method\_declaration" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.wrap\_outer\_expressions\_when\_nested" value="true"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_closing\_paren\_in\_cast" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.brace\_position\_for\_enum\_constant" value="end\_of\_line"/>  <setting id="org.eclipse.jdt.core.formatter.brace\_position\_for\_type\_declaration" value="end\_of\_line"/>  <setting id="org.eclipse.jdt.core.formatter.blank\_lines\_before\_package" value="0"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_paren\_in\_for" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_paren\_in\_synchronized" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_comma\_in\_for\_increments" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_paren\_in\_annotation\_type\_member\_declaration" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.alignment\_for\_expressions\_in\_for\_loop\_header" value="0"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_opening\_paren\_in\_while" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_closing\_paren\_in\_enum\_constant" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_comma\_in\_explicitconstructorcall\_arguments" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_closing\_paren\_in\_annotation" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_opening\_angle\_bracket\_in\_type\_parameters" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.indent\_body\_declarations\_compare\_to\_enum\_constant\_header" value="true"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_lambda\_arrow" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_brace\_in\_constructor\_declaration" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_comma\_in\_constructor\_declaration\_throws" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.join\_lines\_in\_comments" value="false"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_closing\_angle\_bracket\_in\_type\_parameters" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_question\_in\_conditional" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.comment.indent\_parameter\_description" value="true"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_new\_line\_before\_finally\_in\_try\_statement" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.tabulation.char" value="tab"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_comma\_in\_multiple\_field\_declarations" value="do not insert"/>  <setting id="org.eclipse.jdt.core.formatter.blank\_lines\_between\_import\_groups" value="0"/>  <setting id="org.eclipse.jdt.core.formatter.lineSplit" value="120"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_after\_opening\_paren\_in\_annotation" value="insert"/>  <setting id="org.eclipse.jdt.core.formatter.insert\_space\_before\_opening\_paren\_in\_switch" value="insert"/>  </profile>  </profiles> |

### For Java Code Template

* run eclipse
* Move to Window -> Preferences > java > Code Style > Code Templates
* Clieck “import” button

File : JAVA\_CodeTemplates\_For\_PPCB\_Core\_Banking\_v0.2.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" standalone="no"?><templates><template autoinsert="true" context="gettercomment\_context" deleted="false" description="Comment for getter method" enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.gettercomment" name="gettercomment">/\*\*  \* @return the ${bare\_field\_name}  \*/</template><template autoinsert="true" context="settercomment\_context" deleted="false" description="Comment for setter method" enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.settercomment" name="settercomment">/\*\*  \* @param ${param} the ${bare\_field\_name} to set  \*/</template><template autoinsert="true" context="constructorcomment\_context" deleted="false" description="Comment for created constructors" enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.constructorcomment" name="constructorcomment">/\*\*  \* ${tags}  \*/</template><template autoinsert="false" context="filecomment\_context" deleted="false" description="Comment for created Java files" enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.filecomment" name="filecomment">/\*-----------------------------------------------------------------------------------------  \* NAME : ${file\_name}  \* VER : v0.1  \* PROJ : ${project\_name}  \* Copyright ${year} PPCB All rights reserved  \*-----------------------------------------------------------------------------------------  \* H I S T O R Y  \*-----------------------------------------------------------------------------------------  \* DATE AUTHOR DESCRIPTION  \* ---------- -------------- ------------------------------------------------------------  \* ${currentDate:date('yyyy-MM-dd')} ${user} creation  \*---------------------------------------------------------------------------------------\*/</template><template autoinsert="false" context="typecomment\_context" deleted="false" description="Comment for created types" enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.typecomment" name="typecomment">/\*\*  \* &lt;PRE&gt;  \* -- detail description --  \* &lt;/PRE&gt;  \*  \* @logicalName ${type\_name}  \* @version 0.1, ${currentDate:date('yyyy-MM-dd')}  \*/</template><template autoinsert="false" context="fieldcomment\_context" deleted="false" description="Comment for fields" enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.fieldcomment" name="fieldcomment">/\*\*  \* -- field description (LogicalName :: Physical Name) --  \*/</template><template autoinsert="false" context="methodcomment\_context" deleted="false" description="Comment for non-overriding methods" enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.methodcomment" name="methodcomment">/\*\*  \* &lt;pre&gt;  \* -- detail description --  \* &lt;/pre&gt;  \*  \* @serviceID  \* @logicalName  \* ${tags}  \* @exception  \* @fullPath  \*/</template><template autoinsert="true" context="overridecomment\_context" deleted="false" description="Comment for overriding methods" enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.overridecomment" name="overridecomment">/\* (non-Javadoc)  \* ${see\_to\_overridden}  \*/</template><template autoinsert="true" context="delegatecomment\_context" deleted="false" description="Comment for delegate methods" enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.delegatecomment" name="delegatecomment">/\*\*  \* ${tags}  \* ${see\_to\_target}  \*/</template><template autoinsert="false" context="newtype\_context" deleted="false" description="Newly created files" enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.newtype" name="newtype">${filecomment}  ${package\_declaration}  ${typecomment}  ${type\_declaration}</template><template autoinsert="false" context="classbody\_context" deleted="false" description="Code in new class type bodies" enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.classbody" name="classbody">&#13;  </template><template autoinsert="true" context="interfacebody\_context" deleted="false" description="Code in new interface type bodies" enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.interfacebody" name="interfacebody">  </template><template autoinsert="true" context="enumbody\_context" deleted="false" description="Code in new enum type bodies" enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.enumbody" name="enumbody">  </template><template autoinsert="true" context="annotationbody\_context" deleted="false" description="Code in new annotation type bodies" enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.annotationbody" name="annotationbody">  </template><template autoinsert="false" context="catchblock\_context" deleted="false" description="Code in new catch blocks" enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.catchblock" name="catchblock">// ${todo} Auto-generated catch block  //logger.error( "-- fill in information for detail act -- =&gt; ", e );</template><template autoinsert="true" context="methodbody\_context" deleted="false" description="Code in created method stubs" enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.methodbody" name="methodbody">// ${todo} Auto-generated method stub  ${body\_statement}</template><template autoinsert="true" context="constructorbody\_context" deleted="false" description="Code in created constructor stubs" enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.constructorbody" name="constructorbody">${body\_statement}  // ${todo} Auto-generated constructor stub</template><template autoinsert="true" context="getterbody\_context" deleted="false" description="Code in created getters" enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.getterbody" name="getterbody">return ${field};</template><template autoinsert="true" context="setterbody\_context" deleted="false" description="Code in created setters" enabled="true" id="org.eclipse.jdt.ui.text.codetemplates.setterbody" name="setterbody">${field} = ${param};</template></templates> |

# Business Service Development

## Introduction

### Understanding on e-link solution

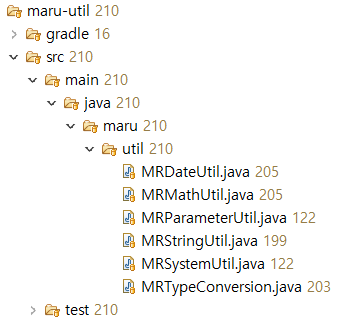
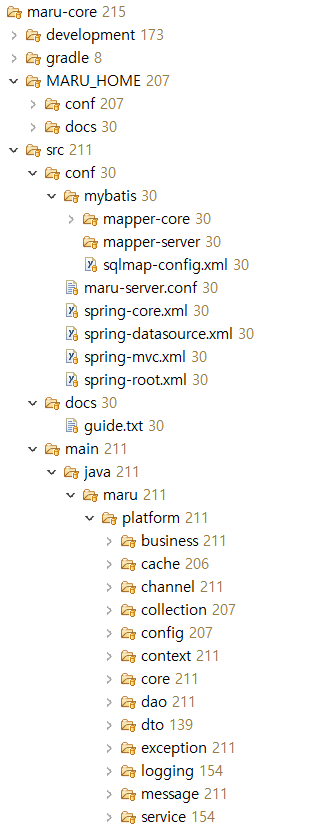
Please refer to e-link reference documents.

### Understanding on xframe UI framework

Please refer to xframe reference documents.

### Understanding on basic structure of MARU framework

#### MARU framework package structure



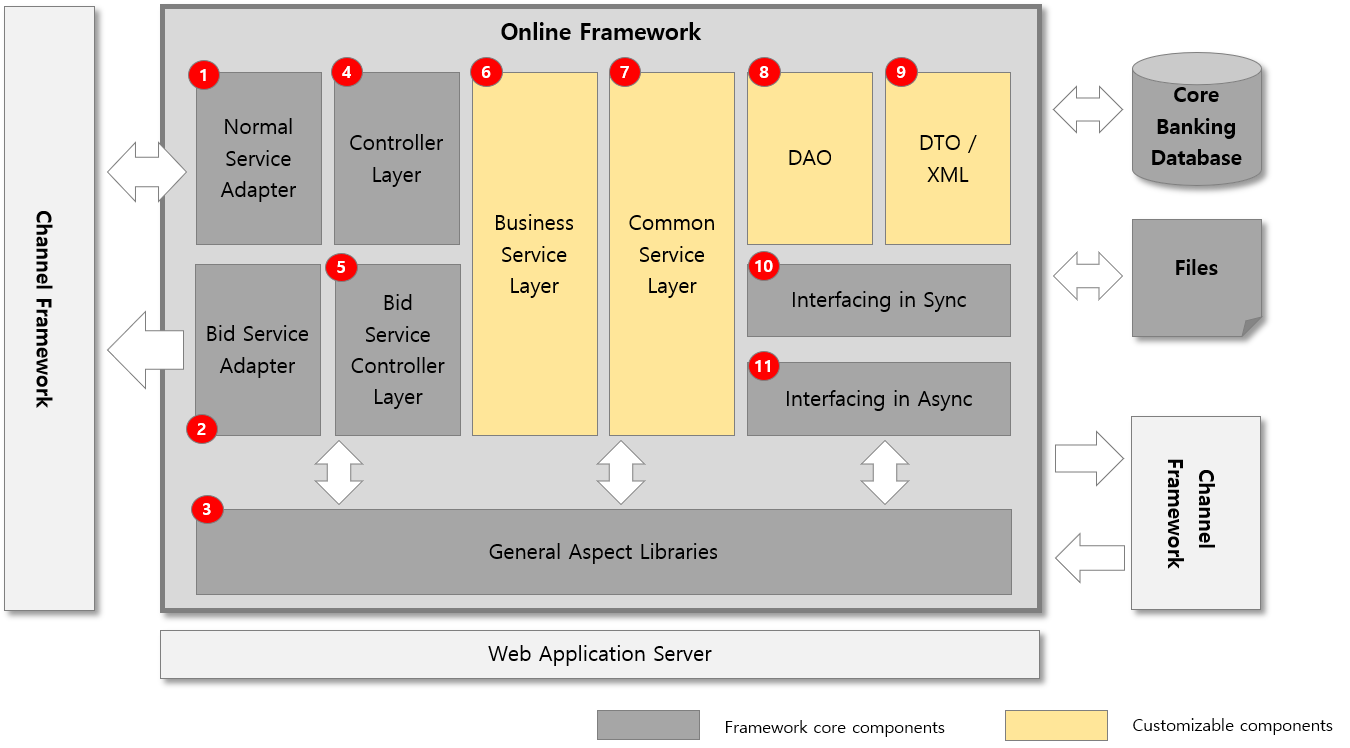
MARU Utility

MARU Server

MARU Core

* It is a core project of MARU framework. It is a base library to run business logic on MARU, and is deployed with a jar file.

#### Framework architecture



1. Normal service adapter

It is running in HTTP protocol with JSON message. Channel framework, specially MCI(multi channel interface) sends a transaction message to online framework through it. It is linked with Normal Service Adapter of MARU framework(online framework).

|  |
| --- |
| Note. URL for normal service invocation  http:// ….. / |

1. Bid service adapter

Bid service is used for propagate a message to teller or teller application from business service or center, so it is bound for outside of core bank system through channel system.

|  |
| --- |
| Note. URL for bid service invocation  http:// … / |

1. General aspect libraries

It is packaged into utility (…..package 경로를 적어주자). It is referenced from other component or business service in a form of internal API.

A

1. Controller layer
2. Bid service controller layer
3. Business service layer
4. Common service layer
5. DAO
6. DTO
7. Interfacing in Sync
8. Interfacing in Async

## Business service implementation steps

### Message definition

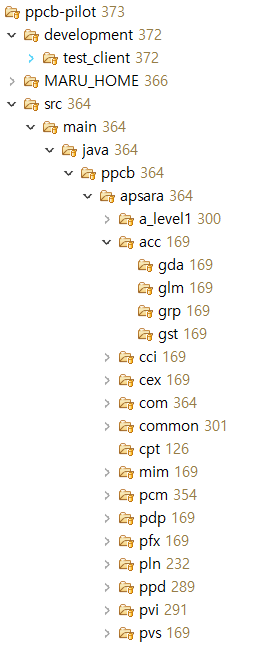
Please refer to e-link reference documents.

### Client-side logic and UI development

Please refer to xframe reference document.

### Server-side logic development

여기에는 서버 개발 방법론의 개론만 기술한다. 대략적인 스텝과 각 스텝에서 필요로 하는 기술요소, 참고자료 등을 기술하여 다음 챕터의 내용을 이해할 수 있도록 준비하도록 한다.

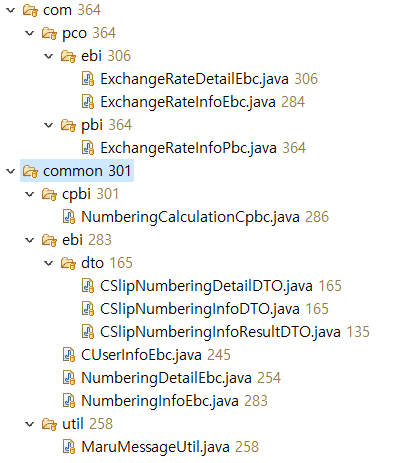
Unit test

Main business logic

Level 2 code

Level 1 code

Project code : APSARA



Util

* Framework utilities which can be used by business logic

CPBI/CPBC (Common PBI/PBC)

* Common business interface and implementation

PBI/PBC (Project Business Interface/Class)

* Business interface and implementation

EBI/EBC (Entity Business Interface/Class)

* Implementation for entity handling

나중에 생성툴을 포함한 개발 방법 개론을 여기에 기술해 준다.

## Business Logic Implementation with MARU framework

### CPBI/CPBC

|  |
| --- |
| package ppcb.apsara.common.cpbi;  import java.math.BigDecimal;  import org.apache.commons.lang3.StringUtils;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Service;  import maru.platform.collection.MData;  import maru.platform.context.util.MContextUtil;  import maru.util.MRDateUtil;  import maru.util.MRMathUtil;  import maru.util.MRStringUtil;  import maru.util.MRTypeConversion;  import ppcb.apsara.common.ebi.NumberingDetailEbc;  import ppcb.apsara.common.ebi.NumberingInfoEbc;  /\*\*  \* Numbering Seq Calculation  \*  \* @param iNumberingDetailDTO  \* @return  \*/  @Service  public class NumberingCalculationCpbc {  @Autowired  NumberingInfoEbc numberingInfoEbc;  @Autowired  NumberingDetailEbc numberingDetailEbc;    /\*\*  \* input  \* numberingTypeCode  \* numberingCombinationCodeDetail  \* tellerID  \* ouput  \*  \*  \* @param param  \* @return  \*/  public String getCmptNumbering(MData iNumberingDetail) {  String sLastNmbrItem = ""; // 최종채번항목명  String snmbrDivNm = ""; // 채번구분명  String resultDigit = "";  int countNmbrDetail = 0;  String sNmbrCtnt = ""; // 채번일련번호자리수 길이만큼 왼쪽에 0을 padding    BigDecimal bdLastNo = BigDecimal.ZERO;    // long lNmbrNoTmcnt = 0L; // #  String sNmbrDscExisYn = "N"; //# // s처리코드(Y:Update 대상, N:Insert 대상)      //@ 채번기본 MData  MData selectNumberingInfo = new MData(iNumberingDetail);  MData selectNumberingDetail = null;    String numberingTypeCode = iNumberingDetail.getString("numberingTypeCode");  String numberingCombinationCodeDetail = iNumberingDetail.getString("numberingCombinationCodeDetail");  // =============================================================================  // ######### 입력값 검증  // =============================================================================  if (StringUtils.isEmpty(numberingTypeCode)) {  throw new IllegalArgumentException("numberingTypeCode string may not be null");  }    if (StringUtils.isEmpty(numberingCombinationCodeDetail)) {  throw new IllegalArgumentException("numberingCombinationCodeDetail string may not be null");  }    try {  BigDecimal bdNxtrNmbrNo = BigDecimal.ZERO; // #  //@ 채번명세 테이블 존재여부 체크  countNmbrDetail = numberingDetailEbc.getCountNmbrDetail( iNumberingDetail );  //@ 채번명세 select  selectNumberingDetail = numberingDetailEbc.selectNumberingDetail( iNumberingDetail );  //@ 채번명세테이블에 미존재시 최종채번번호는 1로 setting  if ( countNmbrDetail == 0 ) {  bdNxtrNmbrNo = BigDecimal.ZERO;  } else {  bdNxtrNmbrNo = selectNumberingDetail.getBigDecimal( "numberingLastNo" );  sNmbrDscExisYn = "Y";  }    try {  //@ 채번기본 select  selectNumberingInfo = numberingInfoEbc.selectNumberingInfo( iNumberingDetail );  //@ 채번시작번호  BigDecimal bdNmbrStrtNo = selectNumberingInfo.getBigDecimal("numberingStartNo");  //@ 채번최종번호 + 채번시작번호  bdLastNo = MRMathUtil.bdAdd(bdNxtrNmbrNo, bdNmbrStrtNo);    //@ 채번일련번호 자릿수  int iDigit = MRTypeConversion.toInteger(selectNumberingInfo.getBigDecimal( "numberingSeqNoDigit" ));  //@ 채번일련번호자리수 길이만큼 왼쪽에 0을 padding  sNmbrCtnt = StringUtils.leftPad(MRTypeConversion.toString(bdLastNo) + "", iDigit, "0");  // =============================================================================  // @ 체크디지트산출 입력셋팅  // =============================================================================  MData iCheckDigit = new MData(iNumberingDetail);    //@ 체크디지트여부에 따라 분기처리  if (StringUtils.equalsAnyIgnoreCase((String)selectNumberingInfo.get("checkDigitUseYN"), "Y")) {    resultDigit = this.verifyChekDigit(iCheckDigit, sNmbrCtnt); //##체크디지트산출    System.out.println( ">>resultDigit =========" + resultDigit );  //@ 최종채번항목내용 = 입력한 채번항목내용 + 가중치를 계산한 값  sLastNmbrItem = numberingCombinationCodeDetail + sNmbrCtnt.toString() + resultDigit.toString();  }  else {  //@ 최종채번항목내용 = 입력한 채번항목내용 + 가중치를 계산한 값  sLastNmbrItem = numberingCombinationCodeDetail + sNmbrCtnt.toString();  }  System.out.println( ">>count =========" + sNmbrCtnt );  System.out.println( ">>채번조합내용 =========" + sLastNmbrItem );      } catch (Exception e) {  System.out.println( ">> Exception=========" );  e.printStackTrace();  }    // =============================================================================  // ######### 채번명세 입력값 셋팅  // =============================================================================  //@ 채번구분코드  iNumberingDetail.put("numberingTypeCode", numberingTypeCode);  //@ 채번조합코드내용  iNumberingDetail.put("numberingCombinationCodeDetail", numberingCombinationCodeDetail);  //@ 채번최종번호  iNumberingDetail.put("numberingLastNo", bdLastNo);  //@ 최종거래일자  iNumberingDetail.put("lastTransactionDate", MRDateUtil.getCurrentDate());      //@ 채번명세및 기본테이블에 미존재시 insert  if (StringUtils.equals(sNmbrDscExisYn, "N")) {  System.out.println( ">>채번명세등록 =========" + sLastNmbrItem );  //@ 채번조합코드내용  iNumberingDetail.put("numberingCombinationCodeDetail", numberingCombinationCodeDetail);  //@ 최초등록직원번호  iNumberingDetail.put("firstRegisterEmpNo", MContextUtil.getTellerID());  //@ 최종변경직원번호  iNumberingDetail.put("lastChangeEmpNo", MContextUtil.getTellerID());    numberingDetailEbc.registerNumberingDetail( iNumberingDetail );  }  else {  System.out.println( ">>채번명세수정 =========" + sLastNmbrItem );  //@ 채번조합코드내용  iNumberingDetail.put("numberingCombinationCodeDetail", numberingCombinationCodeDetail );  //@ 최종변경직원번호  iNumberingDetail.put("lastChangeEmpNo", MContextUtil.getTellerID() );  //@ update  numberingDetailEbc.updateNumberingDetail( iNumberingDetail );  }  } catch (Exception e) {  System.err.println(e);  e.printStackTrace();  }    return sLastNmbrItem;    }  /\*\*  \*  \*  \* @designSeq  \* @logicalName 체크디지트검증  \* @param LData iCkdgValdtn i체크디지트검증 : 체크디지트검증입력Dto  \* @return LData rCkdgValdtn r체크디지트검증 : 체크디지트검증출력Dto  \* @exception LException 에러 발생  \*  \* @modelProject COMMON  \*/  public String verifyChekDigit(MData iCheckDigit, String sNmbrCtnt) throws Exception {    String numberingCombinationCodeDetail = iCheckDigit.getString("numberingCombinationCodeDetail");  long lCheckDigit = 0L;  String sCheckDigit = "";    //@ 채번기본 select  MData selectNumberingInfo = numberingInfoEbc.selectNumberingInfo( iCheckDigit );  if (StringUtils.isEmpty(selectNumberingInfo.getString("weightCompositionNo"))) {  System.out.println( ">>가중치번호 오류===" + selectNumberingInfo.get("weightCompositionNo"));  }  // =============================================================================  // ######### GeneralCodeBlock Weight계산    // =============================================================================  // (2). (1)항에서 구한, 나머지 값들을 모두 합한 후, 10으로 나누어 나머지를 구한다.  // (1). 각 숫자에 (Weight = '137137137')을 곱한 후, 10으로 나누어 나머지를 구한다.  // (3). (2)항에서 구한 값을 9에서 뺀 값을 검증번호로 한다.  //----------------------------------------------------------------------------------  //@ 채번조합항목내용  String sCodeDetail = numberingCombinationCodeDetail + sNmbrCtnt.toString();  //@ loop건수  int iMax = sCodeDetail.length();  System.out.println("<<======= 입력값" + numberingCombinationCodeDetail );  System.out.println("<<======= 가중치번호" + selectNumberingInfo.getString("weightCompositionNo"));  System.out.println("<<======= 채번조합내용" + sCodeDetail);  System.out.println("<<======= 자릿수" + iMax);    for (int i = 0; i < iMax; i++) {  lCheckDigit = ((MRTypeConversion.toInteger(MRStringUtil.mid(selectNumberingInfo.getString("weightCompositionNo"), i, 1)))  \* MRTypeConversion.toInteger((MRStringUtil.mid(sCodeDetail, i, 1)))) + lCheckDigit;  // % 10 ) + lCheckDigit;  }  //@ 체크디지트 정의  long lRmn = 0L;  long lChekcDigitCalc = 0L;    lRmn = lCheckDigit % 11;  if (lRmn == 10 ) {  lChekcDigitCalc = 11;  }  else if (lRmn == 0 ) {  lChekcDigitCalc = 10;  }  else if (lRmn == 1 ) {  lChekcDigitCalc = 11;  }  else {  lChekcDigitCalc = lRmn;  }  System.out.println("<<======= 체크디지트= [" + lCheckDigit);  //@ Weight계산후 최종값  sCheckDigit = MRTypeConversion.toString(11 - lChekcDigitCalc);  System.out.println("<<======= Weight계산후 최종값= [" + sCheckDigit);    // =============================================================================  // ######### GeneralCodeBlock ##체크디지트결과 셋팅  // =============================================================================  return sCheckDigit;  }  } |

### EBI/EBC

|  |
| --- |
| package ppcb.apsara.pcm.pcu.ebi;  import org.springframework.stereotype.Repository;  import maru.platform.collection.MData;  @Repository  public interface IndividualCustomerDetailEbc {    /\*\*  \*  \* @param param  \* @return  \*/  int regtIndividualCustomerDetail(MData param);    /\*\*  \* <pre>  \* input  \* customerNo  \* customerName  \* customerAddress  \* </pre>  \*  \* @param param  \* @return  \*/  MData inquiryIndividualCustomerDetailByCustomerID(MData param);    } |

### PBI/PBC

|  |
| --- |
| package ppcb.apsara.pcm.pcu.pbi;  import java.time.LocalDateTime;  import java.util.Arrays;  import org.apache.commons.lang3.StringUtils;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.transaction.PlatformTransactionManager;  import org.springframework.transaction.TransactionStatus;  import org.springframework.transaction.interceptor.DefaultTransactionAttribute;  import maru.platform.business.annotation.BusinessComponent;  import maru.platform.business.annotation.Operation;  import maru.platform.collection.MData;  import maru.platform.context.util.MContextUtil;  import maru.platform.exception.MOperationException;  import maru.platform.message.CommonBody;  import maru.platform.message.MaruMessage;  import maru.platform.message.MessageBody;  import maru.platform.message.builder.CommonBodyBuilder;  import maru.platform.message.builder.MaruMessageBuilder;  import maru.platform.message.builder.MessageBodyBuilder;  import maru.platform.message.type.MessageHeaderTypeCode;  import maru.util.MRTypeConversion;  import ppcb.apsara.common.cpbi.NumberingCalculationCpbc;  import ppcb.apsara.pcm.pcu.ebi.CustomerContactInfoEbc;  import ppcb.apsara.pcm.pcu.ebi.CustomerInfoEbc;  import ppcb.apsara.pcm.pcu.ebi.IndividualCustomerDetailEbc;  /\*\*  \*  \* @author Kevin  \*  \*/  @BusinessComponent  public class IndividualCustomerInfoPbc {  @Autowired  private CustomerInfoEbc customerInfoEbc;  @Autowired  private CustomerContactInfoEbc customerContactInfoEbc;  @Autowired  private IndividualCustomerDetailEbc individualCustomerDetailEbc;    @Autowired  private NumberingCalculationCpbc numberingCalculationCpbc;    @Autowired  private PlatformTransactionManager txManager;  /\*\*  \* Customer Register 고객등록  \* @param request  \* @return  \*/  @Operation("PCU00000121")  public MData createIndividualCustomerInfo( MData request ) throws MOperationException {    // Input Message Define 표준전문데이터부 - 입력바디  MData inputData = request;    // Output Message Define 표준전문데이터부 - 출력바디  MData outputData = new MData();    TransactionStatus txStatus = txManager.getTransaction( new DefaultTransactionAttribute() );  try {  // Pre Pocessing 선처리  //preProcessing();    // Input Validation 입력항목검증  //inputValidation();  // CustomerID Type Code Check 고객실명번호구분코드 검증  if ( StringUtils.isEmpty( inputData.getString("customerIDNoTypeCode")) ) {  System.out.println( "Input Error : customerIDNoTypeCode is Null!!!" );  throw new MOperationException("000001", "customerIDNoTypeCode is Null!!!");  }  // Customer Full Name Check 고객명 검증  if ( StringUtils.isEmpty( inputData.getString("customerName")) ) {  System.out.println( "Input Error : customerName is Null!!!" );  throw new MOperationException("000001", "customerName is Null!!!");  }  // National Country Code Check 국적국가코드 검증  if ( StringUtils.isEmpty( inputData.getString("nationalityCountryCode")) ) {  System.out.println( "Input Error : nationalityCountryCode is Null!!!" );  throw new MOperationException("000001", "nationalityCountryCode is Null!!!");  }  // Resident Country Code Check 거주지국가코드 검증  if ( StringUtils.isEmpty( inputData.getString("residentCountryCode")) ) {  System.out.println( "Input Error : residentCountryCode is Null!!!" );  throw new MOperationException("000001", "residentCountryCode is Null!!!");  }  // Related Tables Search & Validation 관련 테이블 검색 및 검증  //tableSearchAndValidation();    // Main Processing 본처리  // mainProcess();  // Call Common Module : Compute Numbering  //callNumberingCalculationCpbc();  MData param = new MData();  param.put("numberingTypeCode","PCU001");  param.put("numberingCombinationCodeDetail","0000010");  param.put("tellerID", MContextUtil.getTellerID());    String cmptNumbering = numberingCalculationCpbc.getCmptNumbering( param );  if ( StringUtils.isEmpty(cmptNumbering) ) {  System.out.println( "Error : Customer number is Null!!!" );  throw new MOperationException("000005", "customerIDNoTypeCode is Null!!!");  }    // Related Tables Processing 관련 테이블 생성 및 갱신  //processRelatedTables();  // CustomerInfo Insert Start 고객기본생성 시작  MData customerInfo = new MData( inputData );  customerInfo.put( "customerID", cmptNumbering);  customerInfo.put( "frameworkChangeTimeStamp", LocalDateTime.now() );  customerInfo.put( "customerStatusCode", "01" );  customerInfo.put( "transactionCustomerYN", "Y" );  customerInfo.put( "customerCorpIndTypeCode", "01" );  customerInfo.put( "localCustomerName",  ( customerInfo.getString( "CustomerLocalFirstName" ) + customerInfo.getString( "CustomerLocalLastName" ) ) );  customerInfo.put( "customerNewDate", MContextUtil.getBusinessDate() );  customerInfo.put( "customerNewBranchCode", MContextUtil.getTransactionBranchCode() );  customerInfo.put( "customerNewEmpNo", MContextUtil.getTellerID() );  customerInfo.put( "firstRegisterBranchCode", MContextUtil.getTransactionBranchCode() );  customerInfo.put( "firstRegisterDateTime", MContextUtil.getSystemDate() + StringUtils.mid( MContextUtil.getSystemTime(), 0, 6) );  customerInfo.put( "firstRegisterEmpNo", MContextUtil.getTellerID() );  customerInfo.put( "lastChangeBranchCode", MContextUtil.getTransactionBranchCode() );  customerInfo.put( "lastChangeDateTime", MContextUtil.getSystemDate() + StringUtils.mid(MContextUtil.getSystemTime(), 0, 6) );  customerInfo.put( "lastChangeEmpNo", MContextUtil.getTellerID() );    int resultCustomerInfo = customerInfoEbc.regtCustomerInfo( customerInfo );    // 고객연락처기본 Insert  MData customerContactInfo = new MData( inputData );    customerContactInfo.put( "customerID", cmptNumbering);  customerContactInfo.put( "customerContactTypeCode", "001" );  customerContactInfo.put( "frameworkChangeTimeStamp", LocalDateTime.now() );    int resultCustomerContactInfo = customerContactInfoEbc.regtCustomerContactInfo( customerContactInfo );    // 개인고객상세 Insert  MData individualCustomerDetail = new MData( inputData );    individualCustomerDetail.put( "customerID", cmptNumbering);  individualCustomerDetail.put( "frameworkChangeTimeStamp", LocalDateTime.now() );  individualCustomerDetail.put( "deathYN", "N");  individualCustomerDetail.put( "customerIDNoTypeCode", inputData.getString( "customerIDNoTypeCode1" ));  individualCustomerDetail.put( "realIDNo", inputData.getString( "realIDNo1" ));  individualCustomerDetail.put( "iDExpiryDate", inputData.getString( "iDExpiryDate1" ));  individualCustomerDetail.put( "residentCountryCode", inputData.getString( "residentCountryCode1" ));  individualCustomerDetail.put( "postalCode", inputData.getString( "postalCode1" ));  individualCustomerDetail.put( "cityName", inputData.getString( "cityName1" ));    int resultIndividualCustomerDetail = individualCustomerDetailEbc.regtIndividualCustomerDetail( individualCustomerDetail );  // Output Message Setting 표준전문데이터부 - 출력바디  //outputMessageSetting();  outputData.put( "customerID", cmptNumbering );    // Process Result Message Setting 표준전문메시지부 - 바디(정상종료)  //msgBody.setOutMsgCd( "100001" );    // Post Processing  //postProcessing();    // ================================================  txManager.commit( txStatus );  return outputData;  } catch (Exception e) {  // ================================================  txManager.rollback( txStatus );  throw new MOperationException("000001", "residentCountryCode is Null!!!", e); //????????????????  }  }    /\*\*  \* Customer Inquiry 고객조회  \* @param request  \* @return  \*/  @Operation("PCU00000111")  public MData inquiryIndividualCustomerInfo( MData request ) throws MOperationException {    // Input Message Define 표준전문데이터부 - 입력바디  MData inputData = request;    // Output Message Define 표준전문데이터부 - 출력바디  MData outputData = new MData();    TransactionStatus txStatus = txManager.getTransaction( new DefaultTransactionAttribute() );  try {  // Pre Pocessing 선처리  //preProcessing();    // Input Validation 입력항목검증  //inputValidation();  // CustomerID Type Code Check 고객실명번호구분코드 검증  if ( StringUtils.isEmpty( inputData.getString("customerID")) ) {  System.out.println( "Input Error : customerID is Null!!!" );  throw new MOperationException("000001", "customerIDNoTypeCode is Null!!!");  }  // Related Tables Search & Validation 관련 테이블 검색 및 검증  //tableSearchAndValidation();  // CustomerInfo Inquiry 고객기본조회  MData customerInfo = new MData();  customerInfo.put( "customerID", inputData.getString("customerID") );  MData resultCustomerInfo = customerInfoEbc.inquiryCustomerInfoByCustomerID( customerInfo );  System.out.println( "customerInfoEbc = " + resultCustomerInfo );  // customerContactInfo Inquiry 고객연락처기본조회  MData customerContactInfo = new MData();  customerContactInfo.put( "customerID", inputData.getString("customerID") );  MData resultCustomerContactInfo = customerContactInfoEbc.inquiryCustomerContactInfoByCustomerID( customerContactInfo );  // individualCustomerDetail Inquiry 개인고객상세조회  MData individualCustomerDetail = new MData();  individualCustomerDetail.put( "customerID", inputData.getString("customerID") );    MData resultIndividualCustomerDetail = individualCustomerDetailEbc.inquiryIndividualCustomerDetailByCustomerID( individualCustomerDetail );    // Main Processing 본처리  // mainProcess();    // Related Tables Processing 관련 테이블 생성 및 갱신  //processRelatedTables();  // Output Message Setting 표준전문데이터부 - 출력바디  //outputMessageSetting();  // CustomerInfo Output Setting 고객기본출력셋팅  outputData.put( "customerLastName", resultCustomerInfo.getString( "customerLastName" ) );  outputData.put( "customerMiddleName", resultCustomerInfo.getString( "customerMiddleName" ) );  outputData.put( "customerFirstName", resultCustomerInfo.getString( "customerFirstName" ) );  outputData.put( "customerName", resultCustomerInfo.getString( "customerName" ) );  outputData.put( "customerLocalLastName", resultCustomerInfo.getString( "customerLocalLastName" ) );  outputData.put( "customerLocalFirstName", resultCustomerInfo.getString( "customerLocalFirstName" ) );  outputData.put( "nationalityCountryCode", resultCustomerInfo.getString( "nationalityCountryCode" ) );  outputData.put( "customerIDNoTypeCode", resultCustomerInfo.getString( "customerIDNoTypeCode" ) );  outputData.put( "realIDNo", resultCustomerInfo.getString( "realIDNo" ) );  outputData.put( "iDExpiryDate", resultCustomerInfo.getString( "iDExpiryDate" ) );  outputData.put( "residentCountryCode", resultCustomerInfo.getString( "residentCountryCode" ) );    // customerContactInfo Output Setting 고객연락처기본출력셋팅  outputData.put( "phoneNo", resultCustomerContactInfo.getString( "phoneNo" ) );  outputData.put( "cellPhoneNo", resultCustomerContactInfo.getString( "cellPhoneNo" ) );  outputData.put( "faxNo", resultCustomerContactInfo.getString( "faxNo" ) );  outputData.put( "emailAddress", resultCustomerContactInfo.getString( "emailAddress" ) );  outputData.put( "remark", resultCustomerContactInfo.getString( "remark" ) );  outputData.put( "postalCode", resultCustomerContactInfo.getString( "postalCode" ) );  outputData.put( "cityName", resultCustomerContactInfo.getString( "cityName" ) );  outputData.put( "detailAddress", resultCustomerContactInfo.getString( "detailAddress" ) );    // individualCustomerDetail Output Setting 개인고객상세출력셋팅  outputData.put( "birthDate", resultIndividualCustomerDetail.getString( "birthDate" ) );  outputData.put( "genderCode", resultIndividualCustomerDetail.getString( "genderCode" ) );  outputData.put( "marriageYN", resultIndividualCustomerDetail.getString( "marriageYN" ) );  outputData.put( "jobDutyCode", resultIndividualCustomerDetail.getString( "jobDutyCode" ) );  outputData.put( "companyName", resultIndividualCustomerDetail.getString( "companyName" ) );  outputData.put( "customerIDNoTypeCode1", resultIndividualCustomerDetail.getString( "customerIDNoTypeCode" ) );  outputData.put( "realIDNo1", resultIndividualCustomerDetail.getString( "realIDNo" ) );  outputData.put( "iDExpiryDate1", resultIndividualCustomerDetail.getString( "iDExpiryDate" ) );  outputData.put( "residentCountryCode1", resultIndividualCustomerDetail.getString( "residentCountryCode" ) );  outputData.put( "postalCode1", resultIndividualCustomerDetail.getString( "postalCode" ) );  outputData.put( "cityName1", resultIndividualCustomerDetail.getString( "cityName" ) );  outputData.put( "cityDistrictName", resultIndividualCustomerDetail.getString( "cityDistrictName" ) );  outputData.put( "communeName", resultIndividualCustomerDetail.getString( "communeName" ) );  outputData.put( "suiteName", resultIndividualCustomerDetail.getString( "suiteName" ) );  outputData.put( "groupName", resultIndividualCustomerDetail.getString( "groupName" ) );  outputData.put( "streetName", resultIndividualCustomerDetail.getString( "streetName" ) );  outputData.put( "buildingName", resultIndividualCustomerDetail.getString( "buildingName" ) );    // Process Result Message Setting 표준전문메시지부 - 바디(정상종료)  //msgBody.setOutMsgCd( "100001" );    // Post Processing  //postProcessing();    System.out.println( "Output Data = " + outputData );    // ================================================  txManager.commit( txStatus );  return outputData;  } catch (Exception e) {  // ================================================  txManager.rollback( txStatus );  throw new MOperationException("000001", "residentCountryCode is Null!!!", e); //?????????????  }  }  } |

# Coding Standard

## Introduction

### Font

Formatter

Courier New

## Structure

### Java source file

Java source file has following structure.

* Beginning comments
* Package declaration
* import strings
* Class or interface description
* Declaration of class or interface

### Class

Java class has these kinds of parts

* Comment of class variables
* Declaration of class variables
* Comment for methods
* Declaration of method(Constructor must be located in advance)

## Comment

### Common

1. All source code must follow guide line to generate Java API Documentation.

(Refer to http://java.sun.com/j2se/javadoc/ )

1. For multi-line comment, ‘/\*\* \*/’ or ‘/\* \*/’ is used. But comment characters and comment content must not be written in same line. To have compliance with Java API Documentation, ‘/\*\* \*/’ is used for commenting, and other case ‘/\* \*/’ is used.

|  |
| --- |
| /\*\*  \* Class constructor specifying number of objects to create.  \*/ |

3) Comment string must be aligned to source code.

|  |
| --- |
| /\*\*  \* Class constructor specifying number of objects to create.  \*/  public class MultipartRequest {  /\*\*  \* Comment  \*/  private HttpServletRequest req; |

### File

For java file comment, general description is required like as create-related information.

It will be located at the top of the file, and details about the function will use the class comment, so focus on the changes, the file overall, and the rights. Use examples are as follows and follow the usage examples faithfully.

|  |
| --- |
| /\*------------------------------------------------------------------------------  \* NAME : TestClass  \* VER : v1.0  \* PROJ : PPCBank Core Banking System  \* Copyright 2018 PPCBank Bank All rights reserved  \*------------------------------------------------------------------------------  \* Update History  \*------------------------------------------------------------------------------  \* DATE AUTHOR DESCRIPTION  \* ----------------- -------------------------- --------------------------------------------------------  \* 2012.01.17 Kevin Ryun Initial version  \* 2012.01.27 Sam Horn add method()  \*-----------------------------------------------------------------------------\*/ |

### Class

1) It is described in front of definition of Class or Interface.

2) ‘/\*\* ... \*/’ is used.

3) Each line of comment is started with ‘\*’.

4) Version must be written with date in following format.

ex) @version 1.0, 2012/01/01

※ If needed, HTML tag can be used for Java Documentation API.

|  |
| --- |
| /\*\*  \* <PRE>  \* comment  \* comment2  \* </PRE>  \*  \* @version 1.0, 2019/01/01  \*/ |

### Member variables

1) It is positioned at upper of member variables.

2) Comment format is ‘/\*\* ... \*/’.

3) It is written with purpose, restriction.

|  |
| --- |
| /\*\*  \* comment  \*/  protected Connection conn; |

### Member method

1) it is defined befor method declaration.

2) Comment format is ‘/\*\* ... \*/’.

3) It is written in simple and clear comment with 1 or 2 lines.

4) serviceID – In case of PBC class, service ID is described in here.

5) logicalName – op name is here.

6) Param – Type, and variable name are briefly described. Write them one line at a time. If not, they should be marked None or not. In the case of MData, the next line describes the name of the Enum class that represents the layout of the MData.

7) Return value - Type and brief description should be written. If not, it should be marked None or not. In the case of MData, the next line describes the name of the Enum class that represents the layout of the MData.

8) Exceptions - List each Exception name and occurrence status, and describe them one line at a time. If it does not exist, it will be marked as None or not.

9) fullPath – The path for the op on the logical model is marked together if necessary, otherwise it is not marked.

|  |
| --- |
| /\*\*  \* comment  \*  \* @serviceID service ID  \* @logicalName OP name  \* @param MData input  \* MData iAdprSeatNoHistListIqry i  \* @return MData rData  \* MData rAdprSeatNoHistListIqry r  \* @exception LException  \* @fullPath ….  \*/  public MData transferUsbk( MData usbkFtrnInp ) throws LException { |

### SQL Statement

1. It is located at the top of the SQL statement.
2. The annotation format uses XML comments (<! - ...-->).
3. Describe the SQL statement outline. (necessary)
4. For conditions that require additional clarification, such as in the WHERE clause, provide additional explanations. (Optional description of constant or complex conditional)
5. Enter the author name, source code path, and statement name immediately after the SELECT, INSERT, DELETE, and UPDATE clauses as comments.

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <!DOCTYPE mapper SYSTEM "http://mybatis.org/dtd/mybatis-3-mapper.dtd" PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN">  -<mapper namespace="ppcb.apsara.pcm.pcu.ebi.CustomerManagementEbc">  -<select resultType="MData" id="retrieveCustomerInfo">  -<![CDATA[  SELECT /\* ksshin / PCM/PCU/CustomerManagementEbc.xml - retrieveCustomerInfo \*/  CSTNO AS customer\_id  , FRMW\_CHNG\_TMST AS framework\_change\_timeStamp  , CUST\_STCD AS customer\_statusCode  , TX\_CUST\_YN AS transactionCustomer\_yn  , CUST\_IDVD\_CRPT\_DVCD AS customer\_corp\_ind\_type\_code  , CUST\_RNNO\_DVCD AS customer\_id\_no\_type\_code  , RNNO AS real\_id\_no  , RLNM\_DVN\_NO AS real\_name\_divide\_no  , CUST\_LOC\_NM AS local\_customer\_name  , CUST\_LOC\_LAST\_NM AS customer\_local\_last\_name  , CUST\_LOC\_FRST\_NM AS customer\_local\_first\_name  , CUST\_NM AS customer\_name  , CUST\_LAST\_NM AS customer\_last\_name  , CUST\_MID\_NM AS customer\_middle\_name  , CUST\_FRST\_NM AS customer\_first\_name  , CHTR\_CD AS person\_characteristic\_code  , NTNT\_CNTRY\_CD AS nationality\_country\_code  , RSDN\_CNTRY\_CD AS resident\_country\_code  , RSDNT\_DVCD AS resident\_type\_code  , CUST\_PSML\_RCPP\_DVCD AS customer\_postal\_address\_type\_code  , OJBB\_SLRY\_TRSF\_YN AS our\_bank\_salary\_auto\_transfer\_yn  , SLRY\_PRVS\_DD AS payroll\_pay\_date  , MNTR\_BANK\_CD AS main\_transaction\_bank\_code  , CUST\_NEW\_DT AS customer\_new\_date  , CUST\_NEW\_BRCD AS customer\_new\_branch\_code  , CUST\_NEW\_EMPNO AS customer\_new\_emp\_no  , EXRS\_MGMT\_BRCD AS dedicate\_branch\_code  , EXRS\_MGMT\_EMPNO AS dedicate\_mgmt\_emp\_no  , EXRS\_CHOIC\_RSCD AS dedicate\_assign\_reason\_code  , NXTR\_EXRS\_MGMT\_BRN\_STDD AS next\_dedicate\_mgmt\_branch\_start\_date  , NXTR\_EXRS\_MGMT\_BRCD AS next\_dedicate\_mgmt\_branch\_code  , NXTR\_EXRS\_CHOIC\_RSCD AS next\_dedicate\_choice\_reason\_code  , SAME\_CSTNO AS same\_customer\_no  , SQ2\_PSPT\_NO AS bank2nd\_passport\_no  , PSPT\_VLD\_DT AS passport\_expiry\_date  , CHNL\_DTLS\_CLCD AS channel\_detail\_type\_code  , CUST\_SGST\_EMPNO AS customer\_referral\_emp\_no  , SGST\_CUST\_CSTNO AS recommend\_customer\_no  , CUST\_EMAIL\_NOTI\_DVCD AS customer\_email\_notify\_type\_code  , FRST\_REG\_BRCD AS first\_register\_branch\_code  , FRST\_REG\_DTTM AS first\_register\_date\_time  , FRST\_REG\_EMPNO AS first\_register\_emp\_no  , LAST\_CHNG\_BRCD AS last\_change\_branch\_code  , LAST\_CHNG\_DTTM AS last\_change\_date\_time  , LAST\_CHNG\_EMPNO AS last\_change\_emp\_no  FROM  COPCU\_CUSMT\_M C  INNER JOIN COPCU\_CUSMT\_M M  ON C.CSTNO = M.CSTNO  INNER JOIN (  SELECT  orders.user\_id  , COUNT(orders.id) AS orders\_count  FROM  orders  ) L  ON C.CSTNO = M.CSTNO  WHERE  CSTNO = #{customerID}  AND  GROUP BY  orders.user\_id  HAVING  ABC = {CSNO}    ]]>  </select>  -<insert id="registerCustomerInfo">  -<![CDATA[  INSERT INTO /\* ksshin / PCM/PCU/CustomerManagementEbc.xml - registerCustomerInfo \*/  INTO COPCU\_CUSMT\_M  (  CSTNO /\* customer\_id \*/  , FRMW\_CHNG\_TMST /\* framework\_change\_time\_stamp \*/  , CUST\_STCD /\* customer\_status\_code \*/  , TX\_CUST\_YN /\* transaction\_customer\_yn \*/  , CUST\_IDVD\_CRPT\_DVCD /\* customer\_corp\_ind\_type\_code \*/  , CUST\_RNNO\_DVCD /\* customer\_id\_no\_type\_code \*/  , RNNO /\* real\_id\_no \*/  , RLNM\_DVN\_NO /\* real\_name\_divide\_no \*/  , CUST\_LOC\_NM /\* local\_customer\_name \*/  , CUST\_LOC\_LAST\_NM /\* customer\_local\_last\_name \*/  , CUST\_LOC\_FRST\_NM /\* customer\_local\_first\_name \*/  , CUST\_NM /\* customer\_name \*/  , CUST\_LAST\_NM /\* customer\_last\_name \*/  , CUST\_MID\_NM /\* customer\_middle\_name \*/  , CUST\_FRST\_NM /\* customer\_first\_name \*/  , CHTR\_CD /\* person\_characteristic\_code \*/  , NTNT\_CNTRY\_CD /\* nationality\_country\_code \*/  , RSDN\_CNTRY\_CD /\* resident\_country\_code \*/  , RSDNT\_DVCD /\* resident\_type\_code \*/  , CUST\_PSML\_RCPP\_DVCD /\* customer\_postal\_address\_type\_code \*/  , OJBB\_SLRY\_TRSF\_YN /\* our\_bank\_salary\_auto\_transfer\_yn \*/  , SLRY\_PRVS\_DD /\* payroll\_pay\_date \*/  , MNTR\_BANK\_CD /\* main\_transaction\_bank\_code \*/  , CUST\_NEW\_DT /\* customer\_new\_date \*/  , CUST\_NEW\_BRCD /\* customer\_new\_branch\_code \*/  , CUST\_NEW\_EMPNO /\* customer\_new\_emp\_no \*/  , EXRS\_MGMT\_BRCD /\* dedicate\_branch\_code \*/  , EXRS\_MGMT\_EMPNO /\* dedicate\_mgmt\_emp\_no \*/  , EXRS\_CHOIC\_RSCD /\* dedicate\_assign\_reason\_code \*/  , NXTR\_EXRS\_MGMT\_BRN\_STDD /\* next\_dedicate\_mgmt\_branch\_start\_date \*/  , NXTR\_EXRS\_MGMT\_BRCD /\* next\_dedicate\_mgmt\_branch\_code \*/  , NXTR\_EXRS\_CHOIC\_RSCD /\* next\_dedicate\_choice\_reason\_code \*/  , SAME\_CSTNO /\* same\_customer\_no \*/  , SQ2\_PSPT\_NO /\* bank2nd\_passport\_no \*/  , PSPT\_VLD\_DT /\* passport\_expiry\_date \*/  , CHNL\_DTLS\_CLCD /\* channel\_detail\_type\_code \*/  , CUST\_SGST\_EMPNO /\* customer\_referral\_emp\_no \*/  , SGST\_CUST\_CSTNO /\* recommend\_customer\_no \*/  , CUST\_EMAIL\_NOTI\_DVCD /\* customer\_email\_notify\_type\_code \*/  , FRST\_REG\_BRCD /\* first\_register\_branch\_code \*/  , FRST\_REG\_DTTM /\* first\_register\_date\_time \*/  , FRST\_REG\_EMPNO /\* first\_register\_emp\_no \*/  , LAST\_CHNG\_BRCD /\* last\_change\_branch\_code \*/  , LAST\_CHNG\_DTTM /\* last\_change\_date\_time \*/  , LAST\_CHNG\_EMPNO /\* last\_change\_emp\_no \*/  )  VALUES  (  #{customer\_id}  , NOW()  , #{customer\_status\_code}  , #{transaction\_customer\_yn}  , #{customer\_corp\_ind\_type\_code}  , #{customer\_id\_no\_type\_code}  , #{real\_id\_no}  , #{real\_name\_divide\_no}  , #{local\_customer\_name}  , #{customer\_local\_last\_name}  , #{customer\_local\_first\_name}  , #{customer\_name}  , #{customer\_last\_name}  , #{customer\_middle\_name}  , #{customer\_first\_name}  , #{person\_characteristic\_code}  , #{nationality\_country\_code}  , #{resident\_country\_code}  , #{resident\_type\_code}  , #{customer\_postal\_address\_type\_code}  , #{our\_bank\_salary\_auto\_transfer\_yn}  , #{payroll\_pay\_date}  , #{main\_transaction\_bank\_code}  , #{customer\_new\_date}  , #{customer\_new\_branch\_code}  , #{customer\_new\_emp\_no}  , #{dedicate\_branch\_code}  , #{dedicate\_mgmt\_emp\_no}  , #{dedicate\_assign\_reason\_code}  , #{next\_dedicate\_mgmt\_branch\_start\_date}  , #{next\_dedicate\_mgmt\_branch\_code}  , #{next\_dedicate\_choice\_reason\_code}  , #{same\_customer\_no}  , #{bank2nd\_passport\_no}  , #{passport\_expiry\_date}  , #{channel\_detail\_type\_code}  , #{customer\_referral\_emp\_no}  , #{recommend\_customer\_no}  , #{customer\_email\_notify\_type\_code}  , #{first\_register\_branch\_code}  , #{first\_register\_date\_time}  , #{first\_register\_emp\_no}  , #{last\_change\_branch\_code}  , #{last\_change\_date\_time}  , #{last\_change\_emp\_no}  )  ]]>  </insert>  -<update id="updateCustomerInfo">  -<![CDATA[  UPDATE /\* ksshin / PCM/PCU/CustomerManagementEbc.xml - updateCustomerInfo \*/  COPCU\_CUSMT\_M  SET  FRMW\_CHNG\_TMST = #{framework\_change\_time\_stamp}  , CUST\_STCD = #{customer\_status\_code}  , TX\_CUST\_YN = #{transaction\_customer\_yn}  , CUST\_IDVD\_CRPT\_DVCD = #{customer\_corp\_ind\_type\_code}  , CUST\_RNNO\_DVCD = #{customer\_id\_no\_type\_code}  , RNNO = #{real\_id\_no}  , RLNM\_DVN\_NO = #{real\_name\_divide\_no}  , CUST\_LOC\_NM = #{local\_customer\_name}  , CUST\_LOC\_LAST\_NM = #{customer\_local\_last\_name}  , CUST\_LOC\_FRST\_NM = #{customer\_local\_first\_name}  , CUST\_NM = #{customer\_name}  , CUST\_LAST\_NM = #{customer\_last\_name}  , CUST\_MID\_NM = #{customer\_middle\_name}  , CUST\_FRST\_NM = #{customer\_first\_name}  , CHTR\_CD = #{person\_characteristic\_code}  , NTNT\_CNTRY\_CD = #{nationality\_country\_code}  , RSDN\_CNTRY\_CD = #{resident\_country\_code}  , RSDNT\_DVCD = #{resident\_type\_code}  , CUST\_PSML\_RCPP\_DVCD = #{customer\_postal\_address\_type\_code}  , OJBB\_SLRY\_TRSF\_YN = #{our\_bank\_salary\_auto\_transfer\_yn}  , SLRY\_PRVS\_DD = #{payroll\_pay\_date}  , MNTR\_BANK\_CD = #{main\_transaction\_bank\_code}  , CUST\_NEW\_DT = #{customer\_new\_date}  , CUST\_NEW\_BRCD = #{customer\_new\_branch\_code}  , CUST\_NEW\_EMPNO = #{customer\_new\_emp\_no}  , EXRS\_MGMT\_BRCD = #{dedicate\_branch\_code}  , EXRS\_MGMT\_EMPNO = #{dedicate\_mgmt\_emp\_no}  , EXRS\_CHOIC\_RSCD = #{dedicate\_assign\_reason\_code}  , NXTR\_EXRS\_MGMT\_BRN\_STDD = #{next\_dedicate\_mgmt\_branch\_start\_date}  , NXTR\_EXRS\_MGMT\_BRCD = #{next\_dedicate\_mgmt\_branch\_code}  , NXTR\_EXRS\_CHOIC\_RSCD = #{next\_dedicate\_choice\_reason\_code}ode}  , SAME\_CSTNO = #{same\_customer\_no}  , SQ2\_PSPT\_NO = #{bank2nd\_passport\_no}  , PSPT\_VLD\_DT = #{passport\_expiry\_date}  , CHNL\_DTLS\_CLCD = #{channel\_detail\_type\_code}  , CUST\_SGST\_EMPNO = #{customer\_referral\_emp\_no}  , SGST\_CUST\_CSTNO = #{recommend\_customer\_no}  , CUST\_EMAIL\_NOTI\_DVCD = #{customer\_email\_notify\_type\_code}  , FRST\_REG\_BRCD = #{first\_register\_branch\_code}  , FRST\_REG\_DTTM = #{first\_register\_date\_time}  , FRST\_REG\_EMPNO = #{first\_register\_emp\_no}  , LAST\_CHNG\_BRCD = #{last\_change\_branch\_code}  , LAST\_CHNG\_DTTM = #{last\_change\_date\_time}  , LAST\_CHNG\_EMPNO = #{last\_change\_emp\_no}  WHERE  CSTNO = #{customer\_id}  ]]>  </update>  </mapper> |

### Etc

* 1. When describing a special item inside a code, use a line-by-line comment (//), but use it as much as possible.
  2. If you want to annotate after a very short statement inside your code, you can write a line-by-line comment (//) on the same line as the statement. If you define a member variable, You can write one on a line and a short line on the same line. In this case, the comments are spaced a lot from the code so that the code is not distracting.
  3. The start of a comment always starts at the same position as the line to be annotated.
  4. Avoid using annotations in code blocks, unless you want to emphasize specific logic or parts that are easy to mistake.
  5. If the annotations occur too often, the quality of the code drops. Therefore, it is a good idea to rethink once again that there is no way to make the program code in that part cleaner in the sense that you need annotations at the time of coding.
  6. Big Decimal operation

- If there is an operation execution logic using Big Decimal, the expression for the operation expression is annotated directly above the logic.

|  |
| --- |
| // Total Increase ratio = unit of increase ratio \* number of routine  taWagImpRt = gdPriImpRt.multiply( tMul ); |

## Code

### Indentation and Spacing

#### Introduction

1. Except for indentation as specified in the program, the source code starts at the first column without spaces.
2. The size of the column on one line should be set according to the standard of each project so that it is not more than 100 Characters. If possible, the line will be longer so that it will not automatically jump to the next line.
3. All indentations are 4 spaces and must be space.
4. Write only one statement per line if possible.
5. Reserved Word Make space after the reserved word.
6. Place a space before and after the operator.
7. Place space after Semicolon (;), comma (,).
8. Do not put a space between parentheses if it is an expression concatenated with nested parentheses.

ex) result = (a \* (b + c + d) \* (e + f));

1. The Unary operation does not place a space.
2. The Cast Operator does not place a space before or after the parentheses.
3. Do not put spaces in [],., ->, &, \*.
4. Place a space before and after the logical operator.

#### Line delimition

If you can not describe everything on one line, the rules for dividing the rows are as follows

1. Split after comma (,).
2. Divide before operator.
3. Divide at the higher level.
4. If the divided line is the same step, place it in the same column as the above line.

(ex 1) Method definition or invocation - If there are many parameters and can not be used on one line, write them on several lines, one per line. If you can write on one line, write it on one line.

|  |
| --- |
| public DamageInfo getDamageDetail( String pDisasterMgmtNo,  String pOccurAddrCode,  String pDmgCauseCode,  Connection pConn ) throws SQLException, InformException { |

(ex 2) A long String connection uses the StringBuffer class.

|  |
| --- |
| StringBuffer sql1 = new StringBuffer( “SELECT COLUMN1, COLUMN2, COLUMN3, COLUMN4” );  sql1.append( “FROM TABLE1, TABLE2, TABLE3” );  sql1.append( “WHERE COLUMN1 = ‘ ” + VALUE1 + “’” );  sql1.append( “AND COLUMN2 = `” + VALUE2 + “’” );  sql1.append( “ORDER BY COLUMN1” ); |

※ Note: Do not use the concatenation operator (+) when connecting long strings.

(ex3 ) Comparisons with complicated conditions are used for multiple lines divided into appropriate condition units.

|  |
| --- |
| If ( (condition1 && condition2)  || (condition3 && condition4)  || !(condition5 && condition6) ) {  doSomethingAboutIt();  } |

#### Blank line adding

Insert a blank line if:

1) Between Annotation and Package Name

2) Between package declaration and first import statement

3) between different import statements (java related, lafj related, user api related etc.)

4) Between the local variables in the method and the first statement

5) In front of the annotation

6) Before and after if-else, loop, switch, and try-catch-finally statements

7) In order to improve the readability

Insert two blank lines in the following cases:

1) When separating source code into semantic sections

2) When one or more classes or interfaces are defined in a file

3) When distinguishing each class or interface

### File ( Package & import)

#### Package

1) The package statement is located at the top of the program logic under the file comment.

2) Put one blank line around the front and back.

### Class

1) The declaration of the class starts at the first column.

2) If the declaration spans several lines, it is divided into appropriate criteria and indentation is used.

|  |
| --- |
| public class LoginView extends Frame  implements ActionListener, TextListener {  private Label lblTitle;  private Label lblID;  ... |

3) Class declaration order is as follows.

① Member variable

- constant member variable

- static variable

- non-static variable) : primitive, reference in order etc

② Member method

- Class declaration method : ( Class TestClass { } )

- Constructor method (default constructor)

- main method

4) The class (static) variable should be used only when it is shared by all instances.

5) If the package structure is changed, the class name including the program to package is not used for easy modification. In the import statement, specify the package name of all the classes used, and use only the class name.

|  |
| --- |
| sf.work010.entity.Tbcif10Entity cif = null ... ( X )  import sf.work010.entity.Tbcif10Entity;  Tbcif10Entity cif = serarchCustomer(custNo); ( O ) |

### Class method

1. The method declaration starts with four columns in the column where the class is declared.
2. Do not put a space between the name of the method and the opening parenthesis following it.
3. Place at least one blank after the opening parenthesis and before the closing parenthesis.
4. If the method has no arguments, nothing is written. (showMsgbox ())
5. You can describe multiple parameters on a single line, in which case separate each parameter by comma. Do not put a space before comma, but at least one space after it.
6. If the declaration spans multiple lines, sort the first parameter of each line and align the opening and closing parentheses.
7. If necessary, perform a cleanup operation in the finally.

Ex)

close the open file

close the open cursor

close open database session

close of open socket

1. Declare the return type of each method explicitly. If no value is returned, declare the return type of the method as void.

### Variable

1. The order of description of member variables is constant -> static -> primitive -> reference variable.
2. The member variable declaration of a class indents four spaces from the first column in which the class is declared.

|  |
| --- |
| public class LoginView {  private Label lblTitle;  private Label lblID;  ...  private void confirmRentalInfo() {  String memberNameStr = null;  String rentalItemStr = null;  ...  }  } |

1. Declare only one variable in a line, and do not put a space before the last semicolon.
2. Declare groups of the same type as possible.

|  |
| --- |
| Label lblTitle;  Label lblPasswd;  Button btnSend;  Button btnReset; |

1. Loop variables are declared in the loop.

Use the inx, jnx, and knx formats. Do not use I, j, or k.

|  |
| --- |
| for ( int inx = 0 ; inx < MAX ; inx++ ) {  ...  } |

It is recommended to use for loop form using object

|  |
| --- |
|  |

1. Do not declare unused member variables or local variables.
2. Member variables are initialized in the constructor or method when necessary.

|  |
| --- |
| private Label lblTitle; |

1. Automatic (= local) variables must be initialized when declaring

|  |
| --- |
| Label lblStatus = new Label( "Status" ); |

1. It is necessary to initialize variables such as flag variable, cumulative counter, and return code
2. Array declaration is declared in type [] arrayname format.

|  |
| --- |
| private String[] address = new String[5]; |

### Statement

#### Common

1) Do not code two sentences on one line.

2) In nested sentences, indent at the beginning of the parent sentence.

3) Open braces Indent the next line.

4) Place open parentheses ('(') and closing parentheses (')') on the same line. If the closing parenthesis can not be placed on the same line, it is placed in the same column as the opening parenthesis and placed at the beginning of the new line.

|  |
| --- |
| if ( ((inputFile = getDatafile( "input.dat" )) != null) &&  ((outputFile = getDatafile( "output.dat" )) != null) ) {  …  } |

#### Compound statement

1) All declarations or statements in the method body are indented.

2) The opening brace is at the end of the place you want to use, and the closing brace is in the indentation position in the sentence that starts the opening brace.

|  |
| --- |
| private void showSerial() {  int Count = 0;  if ( stTape.getSerialNo() == null ) {  parentFrame.showWarningDialog( "Error in Opening" );  tfSerialNo.selectAll();  } else {  tfDailyPrice.setText( stCtlgItem.getRentPrice() );  tfRentalTerm.setText( "2" );  }  } |

#### The for statement

1) For the next space, open the parentheses, write the space, write the contents, and put a space before closing the parenthesis.

2) Put at least one space on both sides of the semicolon used between the conditional expressions in the for statement.

3) The sentence attached to the for loop is indented on the next line.

4) If the conditional expression in the for statement does not fit in a single line, divide and arrange the line by conditional expression.

|  |
| --- |
| for ( inx = 0 ; inx < nSamples ; inx++ )  accum += samples[inx];    for ( int inx = 0 ;  inx < lstRentalList.getItemCount() ;  inx++  ) {  rentalItem = new RentalItemEntity();  rentalItemVect.addElement( rentalItem );  } |

#### The while Statement

1) Put a space after a while, open a parenthesis, write a space, write a space, and put a space before closing the parenthesis.

2) The sentence composing the while sentence indents the next line.

|  |
| --- |
| while ( result.next() ) {  member = new ClubMemberShipEntity( result.getString(1).trim(),  result.getString(2).trim()  );  vect.addElement( member );  memberCount++;  } |

#### The do statement

1) Put the do keyword on one line itself.

2) The sentence constituting the do statement indents the next line.

3) Put a space after the while, open a parenthesis, write a space, write the contents, and put a space before closing the parenthesis.

4) At the end of the while statement, use terminating semicolon (;) after the closing parenthesis.

|  |
| --- |
| do {  ...  } while ( isOk ); |

#### The if and if-else Statement

1) Put a space after the if, open the parentheses, write a space, write the contents, and put a space before closing the parenthesis.

2) The sentence constituting the if statement is indented on the next line.

3) Describe the contents of the else statement in the same format as the contents of the if statement.

4) Use a brace if the sentence consists of one line.

|  |
| --- |
| if ( stTape.getSerialNo() == 0 ) {  parentFrame.showWarningDialog( "Error in Serial number" );  tfSerialNo.selectAll();  } else if ( stTape.getSerialNo() == 1 ) {  parentFrame.showWarningDialog( "Not allowed" );  tfSerialNo.notifyAll();  } else {  tfDailyPrice.setText( stCtlgItem.getRentPrice() );  tfRentalTerm.setText( "2" );  } |

#### The switch Statement

1) Write a switch keyword and a Test Expression on the same line, and then write a sentence consisting of Compound Expression from the next line.

2) After the switch keyword, put a space, followed by an open parenthesis, a space, the contents of a switch statement, and a space before the closing parenthesis.

3) Sort the case and default labels by indenting after the switch keyword.

4) Do not put more than one case or default label on one line.

5) The expression of case or default label indents from the beginning of the case keyword.

6) The default label is recommended to increase the readability of the program.

7) If possible, give break in default case. This prevents developers from making mistakes when new conditions are added in the future.

8) It is also possible to use a case without break for use in various cases.

|  |
| --- |
| switch ( respCode ) {  case 1:  add();  break;  case 2:  delete();  break;  default:  System.out.println( err\_msg );  break;  } |

#### try-catch statement

1) Put the try keyword on one line itself.

2) The sentence constituting the try statement is indented on the next line.

3) The catch keyword is written to the next line after the contents of the try statement, but it is arranged in the same column as the try keyword.

4) Put a space after the catch, open a parenthesis, write a space, an Exception name and a variable name, and put a space before closing the parenthesis.

5) At the end of the catch statement, open the parentheses and place the sentences for Exception on the next line with 4 or more spaces.

6) If a finally is needed, finally keyword is written to the next line after the contents of the catch statement.

7) Finally, open the parentheses next and indent the necessary sentences on the next line.

|  |
| --- |
| try {  dailyPrice = Integer.parseInt( tfDailyPrice.getText() );  } catch ( NumberFormatException e ) {  System.out.println( "NumberFormat Exception" );  parentFrame.showWarningDialog( "error" );  tfDailyPrice.selectAll();  return;  } finally {  ...  } |

### Expression

#### Binary or Ternary Arithmetic or Logical Operators

Binary or Ternary or using arithmetic or logical operators, put at least one space before and after the operator. This helps distinguish the elements of expression.

|  |
| --- |
| fahren = celsius \* 1.8 + 32 ;  xPix = xCoo \* xCcale + xAxisPix ; |

#### Unary Arithmetic or Logical Operator

Do not use spaces between operators and operands when using unary or logical operators.

|  |
| --- |
| inx = -jnx;  newStatus = !oldStatus; |

#### Cast Operator

- The Cast operator always displays the range of the effect in parentheses.

- You should minimize the use of whitespace when using Cast.

- In Cast operation, type specifier should be separated from parentheses.

- The Cast operator itself should not be separated from the supplied value.

|  |
| --- |
| xScale = (float)( xCooEnd / xAxisLin );  xScale = ( (float)xCooEnd ) / xAxisLin; |

#### Comma Operator

When using the Comma operator, do not put a space before it, but a space on the right. It is also easy to read because it matches the comma usage in English grammar.

|  |
| --- |
| for ( inx = 3, jnx = 17 ; inx < X\_END ; inx ++, jnx++ ) {  mapPoint( inx, jnx ) ;  } |

#### Complete Data References

Indexing ([]) and Dot Notation (.) Should be grouped together with operands without spaces.

|  |
| --- |
| initial = midName[0];  accum += test1.samples; |

#### Method call

1) There must be no space between the method name and the left parenthesis after it.

2) If there are no arguments passed to the method, there must be no space between the opening and closing parentheses.

3) If there is an argument passed to the method, there must be at least one space between the argument and the leading and trailing parentheses.

4) The argument passed to the method ensures that whitespace does not precede comma. However, make sure that at least one space follows comma.

|  |
| --- |
| choice = getChoice();  System.out.println( "Enter two floating point values: " );  listMember.setBounds( xPos1, yPos2, xPos1+150, yPos2+250 ); |

#### Constant

1) Generally, the program documentation is enhanced and constants are expressed using static final so that the constant value can easily be changed in the future.

2) Use when the meaning of a character is clear and easily readable, and the value does not change during program execution.

### Etc

1) Only one assignment operator is used for one expression. Do not do the following.

|  |
| --- |
| a = b = 2;  or  if ( ((x = method1()) + (y = method2())) == 10 ) |

2) Logical operation

Do not use more than one logical AND or logical OR operator in a single expression.

Use the NOT operator only when it does not interfere with reading code.

## Example

### Example code

|  |
| --- |
| 추후 추가 |