

UbiVelox Java Code Convention

작성일: 2009 년 11 월 12 일

작성자: Kim, Ki-Yong

1. Introduction

Readability, Descriptive, Consistent 의 세 가지 원칙을 가지고 아래 규칙들을 90% 이상 지키려고 노력하며 각자 재량에 따라 적용한다. 이 문서에 나와 있지 않은 항목은 Java 기본 스타일 (<http://java.sun.com/docs/codeconv/html/CodeConvTOC.doc.html>) 을 따른다.

2. General Rule

- ✓ File 은 최대 2000line 을 넘지 않도록 한다.
- ✓ 한 line 에는 가급적 최대 80 characters 를 넘지 않도록 한다.
- ✓ tab : keep tab / tabstop = 4

3. Formatting Conventions

3.1. File Organization

Member 들은 가급적 다음 순서로 정렬한다

- ✓ Class variables (public → protected → private)
- ✓ Instance variables (public → protected → private)
- ✓ Constructors
- ✓ Methods (grouped by functionality)

3.2. Definitions

3.2.1. Class Definitions

```
class MyClass {  
    ...  
}
```

3.2.2. Method Definitions

```
public void doSomething(int withArgument) {  
    ...  
}
```

3.3. Indentation

3.3.1. Wrapping line rule

- ✓ Break after a comma
- ✓ Break before an operator
- ✓ Align the new line with the beginning of the expression at the same level on the previous line

```
method(longExpression1, longExpression2, longExpression3,  
    longExpression4, longExpression5);  
  
longName1 = longName2 * (longName3 + longName4) * (longName5)  
    + 4 * longName6; // PREFERRED  
  
longName1 = longName2 * (longName3 + longName4  
    - longName5) + 4 * longName6; // AVOID  
  
someMethod(int anArg, Object anotherArg, String YetAnotherArg,  
    Object andStillAnother)  
{  
    ...  
}
```

3.4. Whitespaces

Blank space 는 일반적으로 아래의 경우에 적용된다.

- ✓ Keyword 와 "(" 사이, ")" 와 "{" 사이, "}" or "{"와 keyword 사이
- ✓ Binary operator 와 expression 사이

다음에는 적용하지 않는다.

- ✓ identifier 와 "("사이에는 blank space 가 적용되지 않는다.
- ✓ Unary Operator 는 붙여서 쓴다.

```
for (...) {  
    ...  
}  
  
method(...) {  
    ...  
}  
  
Double length = Math.sqrt(x * +y * y);
```

3.5. Statements

다음 항목들에서 제시하는 대로 괄호 () {} 나 띄어쓰기, 줄바꿈 등을 사용하도록 한다.

3.5.1. if, if-else, if else-if else statements

- ✓ {}는 생략하지 않는 것을 기본으로 한다.
- ✓ 예외 : 해당 statement 가 아주 간단할 경우에는 생략해도 무관

```
if (a == b) {  
    ...  
} else if (...) {  
    ...  
} else {  
    ...  
}  
  
if (condition)  
    simple_statement;  
  
if (condition) very_simple_statement;
```

- ✓ Condition 이 너무 길어서 single line 으로 표현이 안될 때는 condition 과 statement 를 구별하기 위해 { 를 아래로 내린다

```
if ((condition1 && condition2)  
    || (condition3 && condition4)  
    || !(condition5 && condition6))  
{  
    doSomethingAboutIt();  
}
```

3.5.2. for, while, do-while statements

- ✓ 다음 형식을 사용한다.

```
for (initialization; condition; update) {  
    ...  
}  
for (initialization; condition; update)  
;  
do {  
    ...  
} while (...);
```

3.5.3. switch statements

- ✓ switch 와 case 는 같은 시작점을 가진다.
- ✓ case 와 case 문 사이는 blank line 를 가진다.
- ✓ break statement 를 가지지 않는 case 일 경우에는 정확한 code 흐름을 파악하기 위해서 /* falls through */ comment 를 add 한다.

```
switch (condition) {  
case ABC:  
    statements;  
    /* falls through */  
case DEF:  
    statements;  
    break;  
  
case XYZ:  
    statements;  
    break;  
  
default:  
    statements;  
    break;  
}
```

3.5.4. try-catch statements

- ✓ 다음 형식을 사용한다.

```
try {  
    statements;  
} catch (ExceptionClass e) {  
    statements;  
} finally {  
    statements;  
}
```

4. Naming Conventions

4.1. Meaningful names

- ✓ 이름으로부터 의미를 알 수 있도록 한다.

```
// 좋지 않은 예제
if (a < 65) {
    y = 65 - a;
} else {
    y = 0;
}

// 올바른 예제
if (age < RETIREMENT_AGE) {
    yearsToRetirement = RETIREMENT_AGE - age;
} else {
    yearsToRetirement = 0;
}
```

4.2. Acronyms

- ✓ Capitalize only the first letter in acronyms

```
setDSToffset → setDstOffset
```

4.3. Type Names

- ✓ Class 이름을 구성하는 각 단어의 첫 글자가 대문자가 되도록 한다.
- ✓ Class 이름은 영어 명사로 한다
- ✓ Abc 를 상속하면서 Def 속성을 갖는 것은 DefAbc 라 이름짓는 것을 원칙으로 한다.

```
FeedXmlDownloader, Request, DetailActivity
```

4.4. Interface Names

- ✓ 영어 명사 혹은 형용사로 한다.
- ✓ 대문자 I 로 시작한다.

```
IHmcpMessage, IDrawable, IContentDownloadNotification
```

4.5. Method Names

- ✓ 영어 동사로 소문자로 시작하며, 따라오는 각 단어의 첫자는 대문자로 한다.

```
void withdraw(int amount);  
Image getScaledInstance();  
bool isValid();  
String getName();
```

- ✓ 명사, 전치사, 형용사, 부사 등을 덧붙여 이름으로 하는 일을 알 수 있게 한다.

```
createTable(), makevoiceCall(), ...
```

- ✓ 함수 인자의 이름으로 의미를 알 수 있게 한다.

```
switchTask(String name), startswith(char ch), ...
```

- ✓ 함수 인자로 의미를 알 수 있는 경우 함수 이름에 중복해서 쓰지 않는다.

```
bindFeatureAndDevice(Feature feature, Device device); // 좋지 않음  
bind(Feature feature, Device device); // 좋음
```

- ✓ bool 상태를 얻어 오는 함수는 3 인칭 현재시제를 사용한다.

```
isAvailable(), hasJob(), endsWith(char ch), canShowIcons(), ...
```

- ✓ Notification method 는 명사+과거분사 형태로 사용한다.

```
messageReceived(Msg msg), ringOccurred(String callerId), ...
```

4.6. Variable Names

- ✓ 영어 명사로 소문자로 시작하며, 따라오는 각 단어의 첫자는 대문자로 한다.
- ✓ Collection type 의 경우 영어 복수형을 사용한다.

```
String name;  
XmlHandler xmlHandler;  
ListView coverflowEpisodeListView;  
ArrayList<ContentRequest> allRequests;  
String[] targetNames;
```

- ✓ Parameter 에 member 와 같은 이름이 사용되는 경우 this-> 로 구분한다.
하지만, 혼동의 우려가 있으니 가급적 다른 이름을 사용하도록 한다.

```
public void setName(String name) {  
    this->name = name;  
}  
  
public void setName(String aName) {  
    name = aName;
```

```
}
```

4.7. Constant Names

- ✓ 영어 명사를 사용한다.
- ✓ 대문자로 구성하며, 각 연결되는 단어에는 "_"를 이용하여 구분한다.
- ✓ enum 에 대해서도 동일한 규칙을 사용한다.

```
public static final int ALIGN_LEFT = 0;  
public static final int ALIGN_RIGHT = 2;  
public static final String CONTENT_SMS_URL = "content://sms";
```

5. Documentation Conventions

5.1. Synchronization

- ✓ Comment 가 Code 와 항상 Sync 가 맞아야 한다.
- ✓ Abc 를 상속하면서 Def 속성을 갖는 것은 DefAbc 라 이름짓는 것을 원칙으로 한다.

5.2. Comment types

5.2.1. Documentation Comment

- ✓ Programming interface 를 설명할 때 사용하며, JavaDoc 사용법에 준한다.

```
/**  
 * Document Comment (JavaDoc Comment)  
 */
```

5.2.2. Standard Comment

- ✓ Code 를 지우지 않고 숨길 때 사용


```
/*  
    if (smsManager.getSmsListSize() < 2 ) return;  
    currentPage++;  
    if(currentPage >= smsManager.getSmsListSize()) currentPage=0;  
    refreshwidget();  
*/
```

5.2.3. One-line or end-line Comment

- ✓ 상세 구현 설명을 할 때 사용한다

```
if (availableSize <= contentLength) {  
    // not enough SD card memory  
    this.dnbase.notifyNotEnoughMemory();  
    isNotEnoughMemory = true;  
  
    continue;  
}
```

6. References

- ✓ The Elements of Java Style (sigs reference library)
- ✓ <http://java.sun.com/docs/codeconv/html/CodeConvTOC.doc.html>