

Java Code Conversions

Presented by



Code Conventions

- Code should be clean and well packaged.
- In most cases, software is not maintained by the original author
- Hence coding standards improves readability and better understanding to programmers who maintain code afterwards.
- Oracle says 80% of the lifetime cost of a software goes to maintenance.
- Code conventions improve code consistency, so that multiple programmers can implement code in similar patterns.
- Bad coding standard may leads to Performance issues.



Code Conventions : Classes

- Class is a template which describes the details of an object.
- Coding convention for class,
 - Class names should use nouns.
 - First letter should be capitalized.
 - First letter of each internal word capitalized.
 - keep the class name simple and descriptive.
 - Avoid acronyms.

Example :

```
public class Employee,  
public class BankAccount
```



Code Conventions : packages

- Coding convention for packages.
- Package declarations should be the first line in java source code.
- Package name should be in entirely lowercase.
- Package name should be meaningful and explain functionality of classes stored with-in.
- Example:

```
package com.mycom.banking.savingsaccount
```

```
package org.mycom.insurance.life
```



Code Conversions : Variables

Coding convention for variables.

- These are in mixed case with first letter in lowercase.
- Internal word should start with capital letter.
- Variable names should not start with underscore(_) operator or with dollar symbol. But both can be used within variable name.
- The names of the variable should have proper meaning.

Examples :

```
int width, basicSalary;  
double simpleInterest;
```



Code Conversions : Constants

Coding convention for constants :

- Name of constants should be all in upper case.
- Word separation can be done with underscore operator.
- Constants should be always static and final.
- Should have value assigned while declaration.

Examples:

```
static final int MAX_LENGTH = 80;
```

```
static final String COMPANY_NAME = "ABC_Consultants";
```



Code Conversions : Methods

Coding convention for methods:

- These are in mixed case with first letter in lowercase.
- Internal word should start with capital letter.
- Methods names should not start with underscore(_) operator or with dollar symbol. But both can be used within variable name.
- The names of the methods should have proper meaning.
- The names of methods are verbs

Examples :

`showEmpDetails()`

`getInterest()`



Code Conventions : Exceptions

Coding convention for exceptions

- The exception class name should start with capital letter.
- The exception class name should be suffixed with 'Exception' word.
- Exception handling should use try, catch ,finally, throw and throws.

• Ex: InvalidUserNameException

 NegativeBalanceException



Code Conventions : if...else

Coding convention for if ... else

- The keyword if and conditional expression must be in the same line.
- Even if there is a single statement within if, it is always suggested to use braces.

Ex:

```
if (expression) {  
    statement;  
} else {  
    statement;  
}
```



Code Conversions : if...else ...

If ... else constructs:

```
if ( <condition> ) {  
    <statements>  
} else {  
    <statements>  
}
```

If ... else if constructs:

```
if ( <condition> ) {  
    <statements>  
} else if ( <condition> ) {  
    <statements>  
} else {  
    <statements>  
}
```

Compare boolean values to true or false implicitly, not explicitly

✓	X
<pre>if (valid) { ... } if (!valid) { ... }</pre>	<pre>if (valid == true) { ... } if (valid == false) { ... }</pre>

When comparing against null always place the variable before the conditional operator

✓	X
<pre>myName != null</pre>	<pre>null != myName</pre>



Code Conversions : return

- return statement constructs
 - return statement with a value should not use parentheses. Method should have only one exit point (return)

✓	X
<pre>public boolean Validate() { If(username.equals(password) { System.out.Println("Valid"); return true; } System.out.Println("Invalid"); return false; }</pre>	<pre>public boolean Validate() { If(username.equals(password) { System.out.Println("Valid"); return true; } else { System.out.Println("InValid"); return false; } return false; }</pre>



Code Conversions : while, do..while

- While statement should start with a keyword while immediately followed by condition expression. There must be a space between while and opening parenthesis.

```
while (condition) {  
    statement;  
}
```

- Do-while statement should appear as follows.

```
do {  
    statements;  
} while (condition);
```



Code Conventions : switch...case

- Switch statement should starts with a keyword switch followed by condition.
- Space must be included between keyword switch and open parenthesis.
- Every switch statement should include a default case.

```
switch (condition) {  
    case {expr}:  
        statements;  
    case {expr}:  
        statements;  
        break;  
    default:  
        statements;  
}
```



Code Conventions : for loop

- For statement should have the following format

```
for (initialization; condition; update) {  
    statements;  
}
```

- An empty for should have the following format

```
for (initialization; condition; update) ;
```

- Integer loop index variables should not be named with a single letter such as i, j, or k, in keeping with common conventions.



Code Conventions : coding formatting

- Each sections should be separated by a blank line
- Avoid more than 2000 line in a single file
- Avoid using line longer than 80 characters
- White Space and Blank Lines
 - Add one space in the following places:
 - between operators
 - after comma in method declarations and invocations
 - after semicolons in for-loops
 - before and after the assignment operator
 - between a keyword and a parenthesis.
 - after the opening and before the closing parentheses in a method call or in the method parameter list. Note that this varies from the Sun standard in order to add more whitespace and make the code easier to read.



Code Conversions : spacing

- Spacing Guidelines

- Do not use spaces between a function name and parenthesis.

✓	✗
Create()	Create ()

- Do use a single space after a comma between function arguments.

✓	✗
Read(myChar, 0, 1);	Read(myChar,0,1);

- Do not use spaces inside brackets.

✓	✗
x = dataArray[index];	x = dataArray[index];



Code Conversions : spacing ...

- Do use a single space before and after comparison operators

✓	✗
if (x == y)	if (x==y)



Code Conventions : indentation

- Four spaces should be used as the unit of indentation.
- Tab must be set exactly 8 spaces.
- Avoid lines longer than 80 characters.

Wrapping Lines:

- When an expression will not fit on a single line, break it according to those general principles.
- Break after a comma and Align the new line with the beginning of the expression at the same level on the previous line.

Example:

```
function(longExpression1, longExpression2, longExpression3,  
        longExpression4, longExpression5);//PREFER
```




Code Conventions : declarations

- Never put different types of declaration in a single line.

```
int mark, marks[]; //wrong
```

- Avoid declaring same variable name in an inner code block.

```
int score;  
for(int index = 0; index<array.length; index++ ) {  
  
    int score;   
    statements;  
  
}
```



Code Conventions : few best practices

- More number of code or too much complex code will lead to problem for testing and maintenance.
- Divide the complex logic into smaller chunks with which maintenance becomes easier.
- Always try to give very appropriate error messages which is understood by the user who is working on the application.
- When appropriate use the proper Exception handling.
- If required make use of stack trace also which shows sequence of method calls which led up to the exception.



Code Conventions : indentation

- Four spaces should be used as the unit of indentation.
- Tab must be set exactly 8 spaces.
- Avoid lines longer than 80 characters.

Wrapping Lines:

- When an expression will not fit on a single line, break it according to those general principles.
- Break after a comma and Align the new line with the beginning of the expression at the same level on the previous line.



