

# Code Conventions for Java™ Programing Language

Jan. 10, 2001

Original : *Copyright © 1995- 1999, Sun Microsystems, Inc.*  
<http://java.sun.com/docs/codeconv/>

Translated By : [\(raytrust@raytrust.pe.kr\)](mailto:raytrust@raytrust.pe.kr)  
<http://raytrust.pe.kr/>

# Table of Contents

<b>1.</b>	.....	<b>4</b>
<b>2.</b>	.....	<b>4</b>
2.1.	.....	4
2.2.	.....	4
<b>3.</b>	.....	<b>4</b>
3.1.	.....	5
3.1.1.	.....	5
3.1.2.	Package          Import .....	5
3.1.3.	.....	5
<b>4.</b>	.....	<b>6</b>
4.1.	.....	6
4.2.	.....	6
<b>5.</b>	.....	<b>9</b>
5.1.	.....	9
5.1.1.	Block .....	9
5.1.2.	Single-Line .....	10
5.1.3.	Trailing .....	10
5.1.4.	End-Of-Line .....	11
5.2.	.....	11
<b>6.</b>	.....	<b>12</b>
6.1.	.....	12
6.2.	.....	12
6.3.	.....	12
6.4.	.....	13
<b>7.</b>	.....	<b>14</b>
7.1.	.....	14
7.2.	.....	14
7.3.	RETURN .....	14

7.4.	IF, IF-ELSE, IF ELSE-IF ELSE .....	14
7.5.	FOR .....	15
7.6.	WHILE .....	15
7.7.	DO-WHILE .....	16
7.8.	SWITCH .....	16
7.9.	TRY-CATCH .....	16
<b>8.</b>	<b>.....</b>	<b>17</b>
8.1.	.....	17
8.2.	.....	17
<b>9.</b>	<b>NAMING CONVENTIONS .....</b>	<b>18</b>
<b>10.</b>	<b>PROGRAMMING .....</b>	<b>19</b>
10.1.	.....	19
10.2.	.....	20
10.3.	.....	20
10.4.	.....	20
10.5.	가 .....	21
10.5.1.	.....	21
10.5.2.	.....	21
10.5.3.	Conditional Operator '?' Expressions .....	21
<b>11.</b>	<b>CODE EXAMPLES.....</b>	<b>21</b>
11.1.	JAVA SOURCE FILE EXAMPLE.....	22

1.

- code convention 가?
- Code convention 가 :
- lifetime 80%가 .
  - 가 .
  - Code convention , 가 .
  - 가 , 가 .

2.

2.1.

:

	.java
	.class

2.2.

:

GNUmakefile	make . gnumake .
README	.

3.

2000  
 , “ 11.1 Java Source File Example”

3.1.

public 가 . Private  
public , public  
private . Public  
가 .

- 
- Package Import
- 

3.1.1.

, , , C  
/\*  
\*  
\*  
\*  
\*  
\*  
\*  
\*  
\*  
\*/

3.1.2. PACKAGE IMPORT

package  
import :  
package java.awt;  
  
import java.awt.peer.CanvasPeer;

: ASCII  
(com, edu, gov, mil, net, org) 1981 ISO  
Standard 3166

3.1.3.

	/	
1	/ (/**...*/)	“5.2 ”
2	/	
3	, / (/**...*/),	/
4	(static)	public protected package( 가 ) private
5		
6		
7		가 public private

4.

4 가 (

4.1.

80  
:  
70 가

4.2.

Expression 가 ,

- 
- (Operator)
-

- expression

- 8

```
someMethod(longExpression1, longExpression2, longExpression3,  
            longExpression4, longExpression5);
```

```
var = someMethod1(longExpression1,  
                  someMethod2(longExpression2,  
                              longExpression3));
```

가 가

```
longName1 = longName2 * (longName3 + longName4 - longName5)  
            + 4 * longname6; //
```

```
longName1 = longName2 * (longName3 + longName4  
                        - longName5) + 4 * longname6; //
```

, 8

```
//
```

```
someMethod(int anArg, Object anotherArg, String yetAnotherArg,  
            Object andStillAnother) {  
    ...  
}
```

```
// 8
```

```
private static synchronized horkingLongMethodName(int anArg,  
            Object anotherArg, String yetAnotherArg,
```

```
Object andStillAnother) {  
    ...  
}  
  
8- (4 ) :  
  
//  
if ((condition1 && condition2)  
    || (condition3 && condition4)  
    ||!(condition5 && condition6)) { //  
    doSomethingAboutIt(); // 가 .  
}  
  
//  
if ((condition1 && condition2)  
    || (condition3 && condition4)  
    ||!(condition5 && condition6)) {  
    doSomethingAboutIt();  
}  
  
//  
if ((condition1 && condition2) || (condition3 && condition4)  
    ||!(condition5 && condition6)) {  
    doSomethingAboutIt();  
}  
  
ternary expression 가 :  
  
alpha = (aLongBooleanExpression) ? beta : gamma;
```



$$\alpha = (\text{aLongBooleanExpression}) ? \beta : \gamma;$$

```
alpha = (aLongBooleanExpression)
      ? beta
      : gamma;
```

5.

가 : .  
/\*...\*/ // 가 C++ . (doc comments  
) javadoc , /\*\*...\*/ 가 . Doc  
comments HTML .  
comment 가 . Doc  
가 .  
가 가  
가? ' ' .  
, ' 가? ' .  
, ' 가 가  
가 .  
가  
: .  
, .  
form-feed backspace .

### 5.1.

4 가 가 : block, single-line, trailing,  
end-of-line.

### 5.1.1. BLOCK

Diagram illustrating the relationship between blocks and transactions:

- A block contains multiple transactions.
- A transaction is linked to a block.
- A block is linked to a transaction.

```
/*
 *      block comment
 */
```

Block	:	block	/* -
.	:		

```
/* -
 *
 *
 * block
 *
 *
 * one
 *
 * two
 *
 * three
 */
```

### 5.1.2. SINGLE-LINE

“ .) Single-line . ( “ 5.1.1 Block  
single-line :

```
if (condition) {  
    /* Handle the condition. */  
    ...  
}
```

### 5.1.3. TRAILING

| trailing | : |

```
if (a == 2) {
    return TRUE;           /* special case */
} else {
    return isPrime(a);     /* works only for odd a */
}
```

## 5.1.4. END-OF-LINE

```
//
.
,
가
:
.

if (foo > 1) {
    // Do a double-flip.
    ...
}
else {
    return false;        // Explain why here.
}
//if (bar > 1) {
//
//    // Do a triple-flip.
//    ...
//}
//else {
//    return false;
//}
```

## 5.2.

: “ 11.1 Java Source File Example” .

Doc , , , doc . , doc . , : ,

```
/**
 * The Example class provides ...
 */

public class Example { ...
```

doc (/\*\*) ; , , 1 space , 4 space , 5 space . , block (5.1.1 Block .) single-line (5.1.2 Single-Line ) . ,

doc, class block  
doc

## 6.

### 6.1.

```
int level; // indentation level  
int size; // size of table
```

```
int level, size;
```

```
:
```

```
int foo, fooarray[]; //WRONG!
```

```
: space 가
```

```
int level; // indentation level  
int size; // size of table  
Object currentEntry; // currently selected table entry
```

### 6.2.

가 . 가

### 6.3.

```
.( “ {“ “ }”  
;  
가
```

```
void myMethod() {  
    int int1 = 0; //
```

```
    if (condition) {  
        int int2 = 0;    // “ if”  
        ...  
    }  
}
```

for

for

:

```
for (int i = 0; i < maxLoops; i++) { ... }
```

,

:

```
int count;  
...  
myMethod() {  
    if (condition) {  
        int count = 0;    // !  
        ...  
    }  
    ...  
}
```

## 6.4.

,

:

●

“(” 가

●

“{”

●

“}” “}” 가 “{”

null

```
class Sample extends Object {  
    int ivar1;  
    int ivar2;  
  
    Sample(int i, int j) {  
        ivar1 = i;  
        ivar2 = j;  
    }  
  
    int emptyMethod() {}  
  
    ...  
}
```

- 

## 7.

### 7.1.

```
argv++;      // Correct
argc--;      // Correct
argv++; argc--;  // AVOID!
```

### 7.2.

```
“ {      }”
```

- 

- 

- 

```
if-else      ,
for          (
              )
가
```

### 7.3. RETURN

```
가      return      :      return      가
```

```
return;
return myDisk.size();
return (size ? size : defaultSize);
```

### 7.4. IF, IF-ELSE, IF ELSE-IF ELSE

```
if-else      가      :
```

```
if (condition) {
    statements;
}
```

```
if (condition) {  
    statements;  
} else {  
    statements;  
}  
  
if (condition) {  
    statements;  
} else if (condition) {  
    statements;  
} else {  
    statements;  
}
```

: if . 가  
:

```
if (condition) // {} !  
    statement;
```

## 7.5. FOR

for :

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

for ( initialization, condition, update )  
가 :

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

for initialization update , for ,  
(initialization ) . (update  
) .

## 7.6. WHILE

while 가 :

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

while 가 :

```
while (condition);
```

## 7.7. DO-WHILE

do-while 가 :

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

## 7.8. SWITCH

switch 가 :

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

break .  
case .  
switch default case . Default case break  
, case 가 가 .

## 7.9. TRY-CATCH

try-catch 가 :

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



try-catch                  try  
가                          finally    가                  .

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

8.

8.1.

가                          .  
                                :

- 
- 
- 
- 
- 
- block    (5.1.1 Block                          .)    single-line    (5.1.2 Single-Line
- 가

8.2.

:

- .                          :
- while (true) {  
                                ...  
                                }  
                                .
- 가
- .
- .                          binary

unary ( 가 ++ -- )

```
a += c + d;
a = (a + b) / (c * d);

while (d++ = s++) {
    n++;
}
printSize("size is " + foo + " \n");
```

- for expression :

```
for (expr1; expr2; expr3)
```

- Cast :
- ```
myMethod((byte) aNum, (Object) x);
myMethod((int) (cp + 5), ((int) (i + 3)) + 1);
```

# 9. NAMING CONVENTIONS

Naming convention identifier - , .

| Packages | ASCII , 가<br>com, edu, gov, mil, net, org, 1981<br>ISO Standard 316<br>가 .<br>naming convention<br>convention<br>.<br>, , ,<br>, . | com.sun.eng<br>com.apple.quicktime.v2<br>edu.cmu.cs.bovik.cheese |
|----------|------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|
| Classes  | ,<br>.                                                                                                                             | class Raster;<br>class ImageSprite;                              |

|            |                                                                          |                                                                                                           |
|------------|--------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
|            | (가 URL HTML<br>).                                                        |                                                                                                           |
| Interfaces | .                                                                        | interface RasterDelegate;<br>interface Storing;                                                           |
| Methods    | ,<br>.                                                                   | run();<br>runFast();<br>getBackground();                                                                  |
| Variables  | ,<br>.<br>.<br>.<br>.<br>integer i, j, k,<br>m, m, character c,<br>d, e. | Int i;<br>char c;<br>float myWidth;                                                                       |
| Constants  | ANSI<br>(" _ ")<br>( ANSI<br>.).                                         | static final int MIN_WIDTH = 4;<br>static final int MAX_WIDTH = 999;<br>static final int GET_THE_CPU = 1; |

10. PROGRAMMING

10.1.

public  
가 가 .  
가 public  
data structure . class  
( Java 가 struct ), class struct  
behavior 가  
public

## 10.2.

(static)

```
classMethod();           //OK
AClass.classMethod();    //OK
anObject.classMethod();  //AVOID!
```

## 10.3.

for - 1, 0, 1

## 10.4.

가

```
fooBar.fChar = barFoo.lchar = 'c'; // AVOID!
```

|          |      |            |     |
|----------|------|------------|-----|
| Equality | (==) | assignment | (=) |
| .        | :    |            |     |

```
if (c++ = d++) { // AVOID! ( 가 .)
    ...
}
```

```
if ((c++ = d++) != 0) {
    ...
}
```

assignment

assignment

```
d = (a = b + c) + r; // AVOID!
```

```
a = b + c;
d = a + r;
```

## 10.5. 가

### 10.5.1.

expression

```
if (a == b && c == d)    // AVOID!  
if ((a == b) && (c == d)) // RIGHT
```

### 10.5.2.

```
if (booleanExpression) {  
    return true;  
} else {  
    return false;  
}
```

```
return booleanExpression;
```

```
if (condition) {  
    return x;  
}  
return y;
```

```
return (condition ? x : y);
```

### 10.5.3. CONDITIONAL OPERATOR '?' EXPRESSIONS

Ternary operator ?:      ?      binary operator      expression ,  
가      .      :

```
(x >= 0) ? x : -x;
```

## 11. CODE EXAMPLES

## 11.1. JAVA SOURCE FILE EXAMPLE

```

public class 가
.
" " 5.2 "
.
" 3.1.3

/*
 * @(#)CodeConvention.java      0.82 2000/1/17
 *
 * Copyright (c) 2000 Kwang Shin OH.
 * Shin Ra APT. 401 - 1501 KwanYang - DONG, DongAn - GU, AnYang - SI, KOREA
 * All rights reserved.
 *
 * This software is the confidential and proprietary information of Kwang Shin
 * OH ("Confidential Information"). You shall not
 * disclose such Confidential Information and shall use it only in
 * accordance with the terms of the license agreement you entered into
 * with Kwang Shin OH.
 */

package kwangshin.codeconvention;

import kwangshin.*;

/**
 *
 *
 * @version      0.82 17 Jan 2000
 * @author      Firstname Lastname
 */
public class CodeConvention extends Convention{
    /*
     *
     */

    /** classVar1
     *
     */
    public static int classVar1;

    /**
     * classVar2
     *
     *
     *
     */
    private static Object classVar2;

    /** instanceVar1
     *
     */
    public Object instanceVar1;

    /** instanceVar2
     *
     */
    protected int instanceVar2;

    /** instanceVar3
     *
     */
    private Object[] instanceVar3;

    /**

```

```

    * ...      DepositCommodity      .(      )...
    */
public CodeConvention() {
    // ...      ...
}

/**
    * ...      doSomething      .(      )...
    */
public void doSomething() {
    // ...      ...
}

/**
    * ...      doSomethingElse      .(      )...
    * @param someParam
    * @return String
    * @exception exception
    */
public String doSomethingElse(Object someParam) {
    // ...      ...
}
}
```