Payment Transaction and Report management Soure File Specification

SWEGroup5
Version 1.0

Document Control Page

File Name	:	PTRM_CMSource.doc
Template Version	:	28/05/2009

<table>
<tr><th>Version</th><th colspan="3">Change Description</th><th>Remark</th></tr>
<tr><td rowspan="3"><1.0,
02/08/2009></td><td colspan="3">Original Version</td><td rowspan="3"></td></tr>
<tr><td>Performed by / Date</td><td>Reviewed by / Date</td><td>Approved by / Date</td></tr>
<tr><td>Thawatchai Yernyoob
02/08/2009</td><td>Sirikanya Yomkeot
04/08/2009</td><td>Jirasith Srisupho
05/08/2009</td></tr>
</table>

การพัฒนาระบบ Payment Transaction and Report Management ให้ยึดรูปแบบการพัมนาโปรแกรมและ application ดังต่อไปนี้

1 .ภาษาโปรแกรมที่ใช้ในการพัฒนา คือ ภาษา Java

2. package ต้องเป็นตัวเล็กเท่านั้น เช่น

mypackage, com.company.application.ui

3. type ต้องเป็นคำนาม ตัวอักษรตัวแรกของแต่ละคำต้องเป็นตัวพิมพ์ใหญ่ เช่น

Line, AudioSystem

4. ตัวแปร (variable) คำแรกเป็นตัวอักษรเล็กทั้งหมด คำต่อๆไป ตัวอักษรตัวแรกของแต่ละคำต้องเป็นตัวพิมพ์ใหญ่ เช่น

line, audioSystem

5. ค่าคงที่ (constant) ตัวอักษรทุกตัวต้องเป็นตัวพิมพ์ใหญ่เท่านั้น และใช้เครื่องหมาย _ (Underscore) เป็นตัวเชื่อมคำแต่ละคำ เช่น

MAX_ITERATIONS, COLOR_RED

หรือในกรณีที่ไม่ต้องการกำหนดค่าให้กับตัวแปรตรงๆ สามารถกำหนดได้ ดังนี้

```
int getMaxIterations() // NOT: MAX_ITERATIONS = 25
{
  return 25;
}
```

6. method ต้องเป็นคำกริยา คำแรกเป็นตัวอักษรเล็กทั้งหมด คำต่อๆไป ตัวอักษรตัวแรกของแต่ละคำต้องเป็นตัวพิมพ์ใหญ่ เช่น

getName(), computeTotalWidth()

7. กรณีที่ตั้งชื่อจากคำย่อต่างๆ ให้กำหนดตัวแรกเป็นตัวพิมพ์ใหญ่เพียงตัวเดียว นอกนั้นเป็นตัวพิมพ์เล็ก เช่น

exportHtmlSource(); // NOT: exportHTMLSource();

openDvdPlayer(); // NOT: openDVDPlayer();

8. ตัวแปร (variable) ใน Private class ต้องลงท้ายด้วยเครื่องหมาย _ (Underscore) เช่น

```
class Person
{
private String name_;
  ...
}
```

การใช้ setter

```
 void setName(String name)
 {
   name_ = name;
 }
```

9. ตัวแปร (variable) ของ type ต้องเป็นชื่อเดียวกันกับ type นั้นๆ เช่น

void setTopic(Topic topic) /

void setTopic(Topic value) X

void setTopic(Topic aTopic) X

void setTopic(Topic t) X

หรือ

void connect(Database database) /

void connect(Database db) X

void connect(Database oracleDB) X

10. ชื่อตัวแปรทุกตัวต้องตั้งเป็นภาษาอังกฤษ

11. ตัวแปรที่มี scope การใช้งานกว้าง ต้องตั้งชื่อให้ยาว ตัวแปรที่มี scope การใช้งานแคบ ต้องตั้งชื่อให้สั้น เช่น

integer i, j, k, m, n หรือ characters c, d

12. ชื่อของ Object จะต้องเข้าใจได้ง่าย และไม่นำไปใช้ในการตั้งชื่อ Method เช่น

line.getLength(); /

line.getLineLength(); X

13. ใช้ get/set ในการเรียกใช้งานหรือการกำหนดค่า Attribute เช่น

employee.getName();

employee.setName(name);

matrix.getElement(2, 4);

matrix.setElement(2, 4, value);

14. ใช้คำว่า is นำหน้าตัวแปร (variable) หรือ Method ที่มีค่าเป็น boolean เช่น

isSet, isVisible, isFinished, isFound, isOpen

15. ใช้คำว่า compute เพื่อตั้งชื่อ Method ในกรณีที่ Method นั้นๆ ทำงานเกี่ยวกับการคำนวน เช่น

valueSet.computeAverage();

matrix.computeInverse()

16. ใช้คำว่า find เพื่อตั้งชื่อ Method ในกรณีที่ Method นั้นๆ ทำงานเกี่ยวกับการค้นหา เช่น

vertex.findNearestVertex();

matrix.findSmallestElement();

node.findShortestPath(Node destinationNode);

17. ใช้คำว่า initialize เพื่อตั้งชื่อ Method ในกรณีที่ Method นั้นๆ ทำงานเกี่ยวกับการ initial ค่าต่างๆ เช่น

printer.initializeFontSet();

19. กรณีที่ตัวแปรนั้นๆ เป็นตัวแปรสำหรับเก็บชุดของข้อมูล ให้ใส่ s ที่ท้ายคำเพื่อแสดงความเป็นพหูพจน์ (Plural)ด้วย เช่น

Collection<Point> points;

int[] values;

20. ใช้ตัว n นำหน้าชื่อ Object ที่เก็บข้อมูลประเภทตัวเลข เช่น

nPoints, nLines

21. ใช้คำว่า No เป็นคำลงท้ายในการตั้งชื่อตัวแปร(variable) ที่ใช้ในการเก็บหมายเลขต่างๆ เช่น

tableNo, employeeNo

22. ใช้ตัวแปรสั้นๆในการเก็บค่าเพื่อวน loop เช่น i, j, k ดังนี้

```
for (Iterator i = points.iterator(); i.hasNext(); ) {
:
}

for (int i = 0; i < nTables; i++) {
  :
}
```

24. หลีกเลี่ยงการตั้งชื่อโดยใช้คำย่อ เช่น

computeAverage(); /

compAvg(); X

ActionEvent event; /

ActionEvent e; X

ยกเว้นกรณีที่เป็นคำย่อที่รู้จักกันโดยทั่วไป สามารถทำได้ เช่น

HypertextMarkupLanguage เป็น html

CentralProcessingUnit เป็น cpu

หรืออีกกรณีที่เป็นคำย่ออยู่แล้วก็ไม่ควรนำมาขยายให้เป็นคำเต็ม เช่น

cmd มาจากคำว่า command

comp มาจากคำว่า compute

cp มาจากคำว่า copy

e มาจากคำว่า exception

init มาจากคำว่า initialize

pt มาจากคำว่า point

25. หลีกเลี่ยงการตั้งชื่อตัวแปรที่เป็น boolean ให้มีความหมายเป็น Negative เช่น

bool isError; /

isNoError X

bool isFound; /

isNotFound X

26. ตัวแปรประเภท final (final variables) ต้องขึ้นต้นคำด้วย type เช่น

```
final int  COLOR_RED   = 1;
final int  COLOR_GREEN = 2;
final int  COLOR_BLUE  = 3;
```

กรณีที่เป็น Interface

```
 interface Color
 {
   final int RED   = 1;
   final int GREEN = 2;
   final int BLUE  = 3;
 }
```

27. Exception class จะต้องลงท้ายด้วยคำว่า Exception เช่น

```
class AccessException extends Exception
{
  :
}
```

28. Default interface จะต้องขึ้นต้นด้วยคำว่า Default เช่น

```
class DefaultTableCellRenderer
  implements TableCellRenderer
{
  :
}
```

32. ซอร์สไฟล์ ต้องมีนามสกุลเป็น .java เช่น

Point.java

41. Method modifiers ต้องเรียงลำดับ ดังนี้

public static double square(double a); /

static public double square(double a); X

42. การแปลง Type ต้องระบุชนิดของ Type ที่ต้องการเปลี่ยนแปลงด้วย เช่น

floatValue = (int) intValue; /

floatValue = intValue; X

43. การกำหนด Array ต้องกำหนดที่ Type ไม่ใช่กำหนดที่ตัวแปร เช่น

int[] a = new int[20]; /

int a[] = new int[20]; X

double[] vertex; /

double vertex[]; X

int[] count; /

int count[]; X

public static void main(String[] arguments) /

public double[] computeVertex() /

55. การใช้ If ตัว conditional จะต้องอยู่คนละบรรทัด เช่น

```
if (isDone)
  doCleanup(); /
```

if (isDone) doCleanup(); X

62. การกำหนด Block สามารถกำหนดได้ 2 ลักษณะคือ

```
while (!done) {
  doSomething();
  done = moreToDo();
}
```

หรือ

```
while (!done)
{
  doSomething();
  done = moreToDo();
}
```

63. การกำหนด Class จะต้องอยู่ในรูปแบบนี้

```
class Rectangle extends Shape
    implements Cloneable, Serializable
{
  ...
}
```

64. การกำหนด Method จะต้องอยู่ในรูปแบบนี้

```
public void someMethod()
    throws SomeException
{
  ...
}
```

65. การกำหนด if-else จะต้องอยู่ในรูปแบบนี้

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

หรือ

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

หรือ

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

```
else {
statements;
}
```

66. การกำหนด For จะต้องอยู่ในรูปแบบนี้

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

68. การกำหนด While จะต้องอยู่ในรูปแบบนี้

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

69. การกำหนด Do-While จะต้องอยู่ในรูปแบบนี้

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

70. การกำหนด Switch จะต้องอยู่ในรูปแบบนี้

```
switch (condition) {
  case ABC :
    statements;
    // Fallthrough

  case DEF :
    statements;
    break;

  case XYZ :
    statements;
```

```
    break;

  default :
    statements;
    break;
}
```

71. การกำหนด Try-Catch จะต้องอยู่ในรูปแบบนี้

```
try {
  statements;
}
catch (Exception exception) {
  statements;
}
```

หรือ

```
try {
  statements;
}
catch (Exception exception) {
  statements;
}
finally {
  statements;
}
```

73. การใช้ white-space จะต้องเว้นวรรค 1 ตัวอักษร ดังนี้

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

while (true) {   /
while(true){ X
```

doSomething(a, b, c, d); /

doSomething(a,b,c,d); X

case 100 : /

case 100:

for (i = 0; i < 10; i++) { /

for(i=0;i<10;i++){ X

75. Code ที่ทำงานคนละส่วนกัน ให้เว้นบรรทัด 1 บรรทัด เช่น

```
// Create a new identity matrix
Matrix4x4 matrix = new Matrix4x4();

// Precompute angles for efficiency
double cosAngle = Math.cos(angle);
double sinAngle = Math.sin(angle);

// Specify matrix as a rotation transformation
matrix.setElement(1, 1,  cosAngle);
matrix.setElement(1, 2,  sinAngle);
matrix.setElement(2, 1, -sinAngle);
matrix.setElement(2, 2,  cosAngle);

// Apply rotation
transformation.multiply(matrix);
```

77. การกำหนดตัวแปล ต้องกำหนดชิดซ้าย เช่น

```
TextFile  file;
int       nPoints;
double    x, y;
```

80. การเขียน Comment ต้องเขียนเป็นภาษาอังกฤษเท่านั้น

Authorized By:	Received By:	Reviewed By:	Reviewed By:
(Thawatchai Yernyoob)	(Nattawut Riyasan)	(Jirasith Srisupho)	(Sirikanya Yomkoet)
Authorized Date: 02/08/2008	Received Date: 02/08/2008	Reviewed Date: 05/08/2008	Reviewed Date: 05/08/2008
*Configuration Manager	*Tester or CM	*Project Manager	*Project Leader