**JAVA CODE CONVENTIONS**

Java Source Files:

By convention the public classes are listed first among all other classes in the file.

import util.Util;  
  
import java.util.ArrayList;

The beginning comments comprises of basic details about the file like the author, copyright info and it’s current version.

*\** ***@author*** *Manasa  
\** ***@version*** *1.0  
\** ***@since*** *2019-02-27*

Then this is followed by package and import statements. These mark as the initial non-comment statements of the program.

**Class Declaration:**

The first set of lines are comments that describe the basic functionality of a class.

The next is the line which has the class declaration.

Followed by class declaration we have a few lines dedicated to express the implementation details of the class.

The next region is where the actual coding appears where in we have the static variables (incase they are used), the instance variables of the class, the constructor method and other methods implemented by the class in order.

**Line Length & Wrapping Lines:**

**Comments:**

The RiskGame Project has by default the standard implementation comments and documentation comments. The documentation comments come in handy when they are used for Javadoc generation. The implementation comments are used for the programmer’s understanding of the code that is implemented. The implementation comments are further classified as single line comments, block comments, end of line comments and trailing comments.

*Implementation Comments:*

*Block comments:*

The block comments are used to provide details about the block that is coded.

For instance.,

/\*  
 \* Iterate through all Continents and for each continent run bfs on each  
 \*/

*Single line comments:*

These are short statements that are intended only for few set of statements that appear after them. For instance.,

/\* Reinforcement Phase \*/

*Documentation Comments:*

The documentation comments are used to generate Javadoc documentation for the project. Every class in this project has a multitude of documentation comments for creating the API documentation. For instance.,

*/\*\*  
 \* A Scanner instance to read and parse various primitive values.  
 \*/*

**Declaration policy:**

According to the standards we have only one declaration per line and no two declarations are created on the same line. And declarations are placed always at the beginning of a block.

The initialization of local variables are always done after declaration in many cases as a standard.

For instance.,

private String currentPlayer;

private float percentageMap;

Class Declaration:

The class is always declared by making the first letter of class name in upper case. The methods inside the class are separated by a blank line.

For instance.,

public class Player {

---set of statements---

}

**Statements:**

According to the coding conventions each line in the project has atmost one statement per line.

The return statements are always placed at the end of the function and their return value is not enclosed within braces.

The if and if—else statements are created according to the condition below.

if (*condition*) {

*statements*;

}

if (*condition*) {

*statements*;

} else {

*statements*;

}

The for statement has the following syntax:

for (*initialization*; *condition*; *update*) {

*statements*;

}

The while and do...while statements are initialised with the syntax

while (*condition*) {

*statements*;

}

do {

*statements*;

} while (*condition*);

The switch statement is formulated with the below syntax:

switch (*condition*) {

case ABC:

*statements*;

/\* falls through \*/

case DEF:

*statements*;

break;

case XYZ:

*statements*;

break;

default:

*statements*;

break;

}

The try-catch block is of the below format:

try {

*statements*;

} catch (ExceptionClass e) {

*statements*;

}

**Naming Conventions:**

The class name must always begin with a uppercase letter. For instance.,

public class Player {}

The method signature is always implemented as CamelCase. For instance.,

public String getCountryName() {}

The variable names should always be in camel case and short. For instance.,

private String countryName;