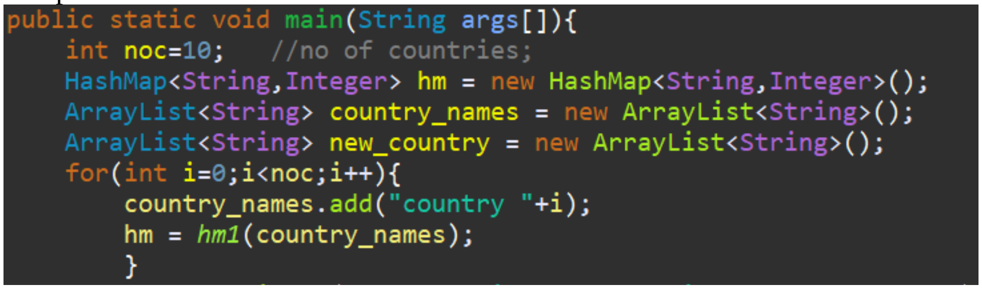
CODING STANDARDS

- All the files are placed in a particular format in a module package named SOEN_6441.
- Files containing map related are in same folder, similarly for the reinforcement and fortification.
- The common elements such as countries, continent, players are common throughout the code.
- Their testcase are in different folder.

1) Coding Convention:

- Code readability is been consider throughout the coding.
- After each function/method **blank line** is used for better readability and sectioning the code.
- It is also used between class declaration in file, between public, protected, private section of code.
- **Minimum code length** is been consider by appending the open curly brace to the statement that precedes it.

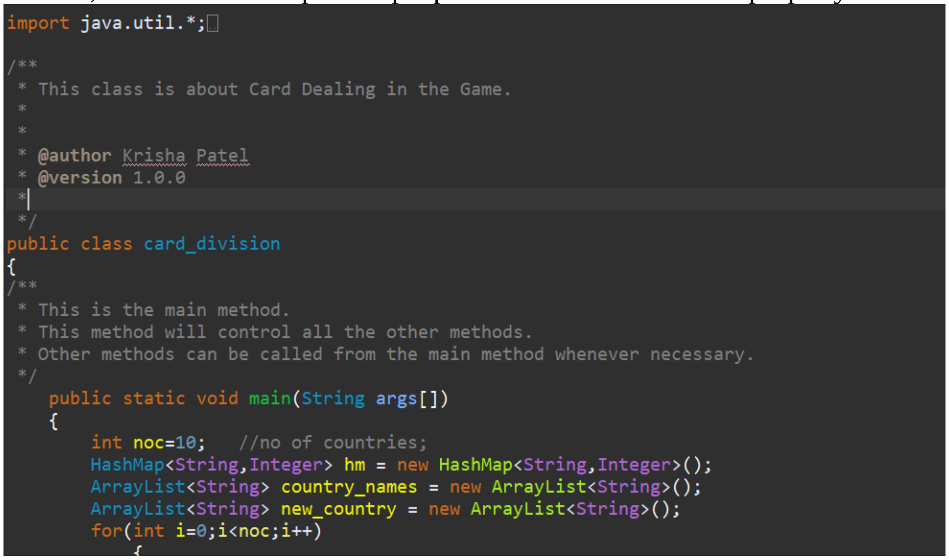


```
public static void main(String args[]){
    int noc=10;   //no of countries;
    HashMap<String,Integer> hm = new HashMap<String,Integer>();
    ArrayList<String> country_names = new ArrayList<String>();
    ArrayList<String> new_country = new ArrayList<String>();
    for(int i=0;i<noc;i++){
        country_names.add("country "+i);
        hm = hm1(country_names);
        }
```

- Blank Line is added before every “API Documentation - JavaDoc”.

2) Commenting Conventions

- ➢ Efficient use of API Documentation – **JavaDoc**
- ➢ Used for all classes and methods to describe there functionality, their parameters, author, version, etc. with will help other people to understand the code properly.

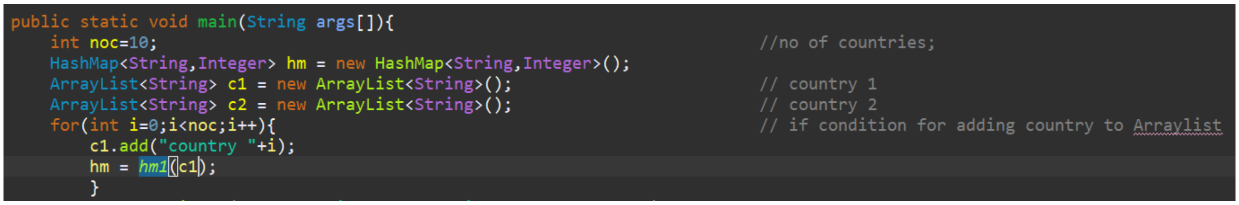


```
import java.util.*;□

/**
 * This class is about Card Dealing in the Game.
 *
 *
 * @author Krisha Patel
 * @version 1.0.0
 *
 */
public class card_division
{
/**
 * This is the main method.
 * This method will control all the other methods.
 * Other methods can be called from the main method whenever necessary.
 */
    public static void main(String args[])
    {
        int noc=10;   //no of countries;
        HashMap<String,Integer> hm = new HashMap<String,Integer>();
        ArrayList<String> country_names = new ArrayList<String>();
        ArrayList<String> new_country = new ArrayList<String>();
        for(int i=0;i<noc;i++)
            {
```

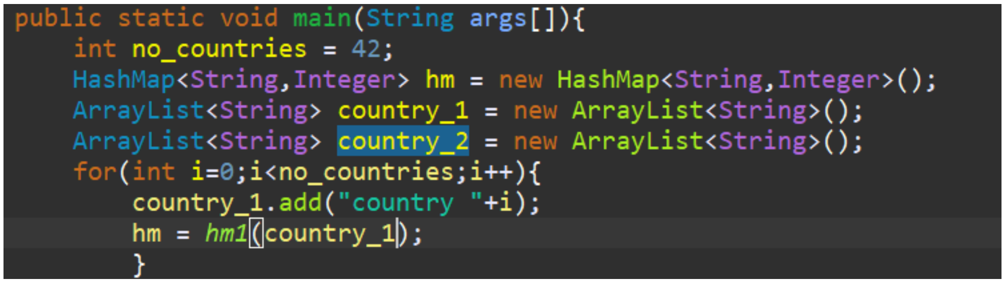
- ➢ Elimination of pointless comments.
- ➢ Refactor code. To eliminate comments in each and every line.

Before



```
public static void main(String args[]){
    int noc=10;                                                              //no of countries;
    HashMap<String,Integer> hm = new HashMap<String,Integer>();
    ArrayList<String> c1 = new ArrayList<String>();                          // country 1
    ArrayList<String> c2 = new ArrayList<String>();                          // country 2
    for(int i=0;i<noc;i++){                                                  // if condition for adding country to Arraylist
        c1.add("country "+i);
        hm = hm1(c1);
        }
```

After



```
public static void main(String args[]){
    int no_countries = 42;
    HashMap<String,Integer> hm = new HashMap<String,Integer>();
    ArrayList<String> country_1 = new ArrayList<String>();
    ArrayList<String> country_2 = new ArrayList<String>();
    for(int i=0;i<no_countries;i++){
        country_1.add("country "+i);
        hm = hm1(country_1);
        }
```

3) Naming Convention

- Class Name:
  - Each class name will start with RGP(Risk Game Player) and then the class name will start with lower case and the next word will start with upper case to bifurcate between the words.

    Eg:

    ```
    RGPmapCreation{
       //code
    }
    ```

- Function Name:
  - Each function name will start with lower case and the next word will have upper case to bifurcate within the words.

    Eg: checkConnection(), armyPerCountry(),

- Variable Name:
  - Each variable will start with lower case, next word will start after a underscore(_) and so on.

    Eg: army_per_country, country_list_for_player

- Object Name:
  - Each object will start with "o" and then name of the function with a underscore.

    Eg: o_RGPmapCreation, o_RGPreinforcement,

References:

Amin Ranj Bar: soen_6441/Lecture 4.pdf