Guidance of JAVA Coding Conventions

PARTI Naming

1. Package

   com.XXXX.XXXX

2. Classes/Interfaces

   a. concrete class:

      a) singular noun

      b) first letter with each internal words capitalized

      c) consistent with table names in DB

      d) simple & descriptive

   Example:

   ```
   public class Department{}
   public class Account extends Class {}
   ```

   b. controller class:

      a) singular noun

      b) first letter with each internal words capitalized

      c) simple & descriptive

   Example:

   ```
   public class DepartmentController{}
   public class AccountController implements Controller {}
   ```

3. Methods

   a. constructor

   b. Getters & setters

      a) Start with get/set + member variable name

      b) first letter lowercased, with the first letter of each internal words capitalized

      Example:

      ```
      public String getDept(){return dept;}
      public void setName(String name) { cname = name; }
      ```

   c. Other methods

      a) verb (followed by noun)

      b) first letter lowercased, with the first letter of each internal words capitalized

      c) simple & make sense

Example:

```
public void createNewCourse(int cid, string cname, …){
    Course course = new Course( , , , ,);
    …
}
public Calendar checkTimetable(int fid){
    return timeTable;
}
```

4. Variables
   a) *first letter lowercased, with the first letter of each internal words capitalized*
   b) *simple & make sense*
   c) *temporary variables: i, j, k, m, and n for integers; c, d, and e for characters.*
   d) *member variables & local variables: avoid using single letter*
   Example:

```
int i;
char c;
Account acc;
```

5. Constants
   a) *uppercase with words separated by underscores*
   b) *static final*
   Example:

```
static final int MIN_WIDTH = 4;
```

PARTII Variable Declarations

1. One declaration per line with trailer comments

```
int level; // indentation level
int[] array;    // array declaration
```

2. Initialization of local variables

JAVA does not require initializing local variables immediately after its declaration; however, it is preferred to have the local variables initialized immediately.

3. Class, interface and methods declaration

```
class object extends Object {
    private int ivar1;
    private int ivar2;

    public constructor (int i, int j) {
        ivar1 = i;
        ivar2 = j;
    }

    public int emptyMethod() {
        if(condition)
        {
        ...
        }
    }

    ...
}
```

PART III Comments

1. Doc comments(Beginning comments of a class)

```
/*
* class name
* version info
* author name
*/
```

2. single line comments

```
if (condition) {
    /* Handle the condition. */ OR
    // Handle the condition.
    If(condition)
        return false;
    ...
}
```

3. block comments

```
if (condition) {
    /* This is an
    *example of
    *block comments
    */
    If(condition)
        return false;
    ...
}
```

4. trailing comments

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

5. comment out

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

OR:

```
/*The following section is for test purpose only
if (foo > 1) {
        // Do a double-flip.
        ...
}
*/
```

6. Methods description

```
/**
*The Example method provides ...
*@param1
*@param2
*@returns
  */
public String Example(int *param1, int param2)
{       ...
        return str;
}
```