

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.util.ArrayList;
import ast.Method;
/**
 * Class that has all the information to create an skeleton such as:<br>
 * - methods to create<br>
 * - imports needed<br>
 * - package<br>
 * - class name<br>
 *
 * @author Sergio Alcocer
 */
public class SkeletonMaker {
    private static SkeletonMaker self;
    private ArrayList<Method> methods;
    private ArrayList<String> imports;
    private String classPackage, className;

    /**
     * Constructor of the class
     */
    public SkeletonMaker() {
        methods = new ArrayList<Method>();
        imports = new ArrayList<String>();
    }

    /**
     * Adds an import to the list
     * @param _import import to be added
     */
    public void addImport(String _import) {
        imports.add(_import);
    }

    /**
     * Sets the package of the class
     * @param _package package to be set
     */
}
```

```
public void setPackage(String _package){
    classPackage = _package;
}

/**
 * Adds a method to the class
 * @param _method method to be added
 */
public void addMethod(Method _method){
    methods.add(_method);
}

/**
 * Sets the name of the class
 * @param _className name of the class to be set
 */
public void setClassName(String _className){
    className = _className;
}

/**
 * Creates the skeleton
 * @param fileName full path of the file to use
 */
public void createSkeleton(String fileName){
    try {
        FileWriter fw = new FileWriter(fileName);
        BufferedWriter out = new BufferedWriter(fw);

        if (classPackage != null && !classPackage.equals("")){
            out.write("package " + classPackage + ";\r\n\r\n");
        }

        for (int i = 0; i < imports.size(); i++){
            out.write("import " + imports.get(i) + ";\r\n");
        }
        out.write("public class " + className + "{\r\n");
        out.write("\t// TODO Write your own class properties\r\n\r\n");

        for (int i = 0; i < methods.size(); i++){
```

```

        Method current = methods.get(i);

        String paramString = new String();
        for (int j = 0; j < current.getParamNumber(); j++){
            paramString += current.getParam(j) + " arg" + j + ", ";
        }
        if(paramString.length() > 0){
            paramString = paramString.substring(0, paramString.length() - 2);
        }

        if(current.isConstructor()){
            out.write("\t" + ((current.getAccess() == Method.PRIVATE)?"private ":"public ") + current.getReturnedType
() + "(" + paramString + "){\r\n");
        }else{
            out.write("\t" + ((current.getAccess() == Method.PRIVATE)?"private ":"public ") + current.getReturnedType
() + " " + current.getName() + "(" + paramString + "){\r\n");
        }

        out.write("\t\t// TODO Write body\r\n");
        out.write("\t}\r\n");
    }
    out.write("}");

    out.close();
    fw.close();
} catch (Exception ex) {
    ex.printStackTrace();
}
}
/**
 * Following the Singleton pattern, this function returns the only instance of the class
 * @return the only instance of SkeletonMaker
 */
public static SkeletonMaker getInstance(){
    if (self == null) self = new SkeletonMaker();
    return self;
}

/**

```

```
* Gets an array of String with the definition of the methods (to be shown on the list)
* @return ArrayList<String> with the methods
*/
public ArrayList<String> getMethodsArray() {
    ArrayList<String> returned = new ArrayList<String>();
    for(int i = 0; i < methods.size(); i++) {
        returned.add(methods.get(i).toString());
    }
    return returned;
}
}
```