軟體工程實務

Software Engineering Practices - Guided Note 5

組別： **Team Number**	學號： **Student ID**	姓名： **Name**

依照附件 **A** 和課程簡報針對 **sample codes (**附錄 **B)** 進行 **code review**，並記錄於附件 **C&D**

Sample codes 包含一個檔案：「附錄 **B-filesCLI.java**」。這個檔案是從「檔案及目錄操作軟體」中所擷取出來的。顧名思義，**filesCLI** 的功能為提供「使用者介面」，此介面為文字模式的介面，使用 **Console** 做為輸入與輸出。**filesCLI** 本身儘量不執行使用者指定的動作，而是呼叫另外一個類別 **SystemOperation**，去控制與操作檔案系統。

附錄 **B: fileCLI.java** 宣告並定義 **fileCLI** 的各項成員函數和實作
附件 **A** 中的 **Hint** 欄位表示在 **Sample code** 中有該 **bad smell** 或違反 **coding standard** 的個數
Follow Appendix A and the course slides to do code review of the sample code provided in Appendix B, then record the results in appendix C&D.

Sample codes in “Appendix B - filesCLI.java” is extracted from a “file and directory operation software”. As its name suggests, fileCLI is aimed at providing an “Command Line user Interface”, which uses Console to do Input/Output.

“filesCLI” should not do the underlying action specified by the user directly, but call another class “SystemOperation” to control and deal with the file system.

Appendix B: “fileCLI.java” declares and defines every member methods(functions) and their implementations.

“Hint” column in Appendix A is the count of that bad smell or coding standard violation.

Appendix A -
Generic Code Review Sample Check List

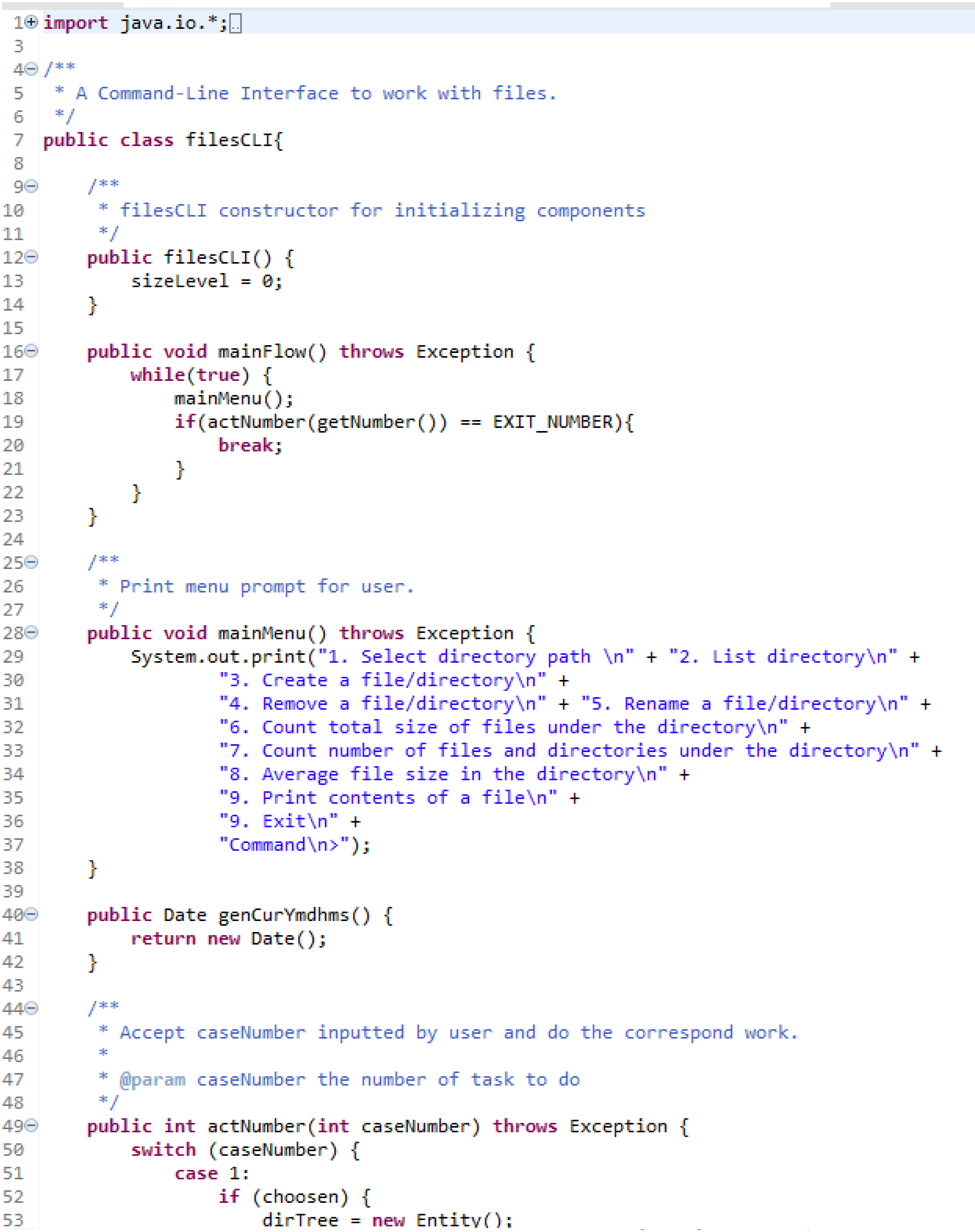
GN 5-1 Code Smell

No	Item to check	Checked	Hint
	Class & Method Definitions		
1	Large class.		1
2	Long method.		1
3	Duplicated code.		1
	Variables (Fields)		
4	Unsuitable naming		8
5	Are there any redundant or unused variables?		2
6	Every variable is properly initialized.		2
	Structures		
7	There are uncalled or unneeded procedures or any unreachable code.		1
	Loops and Branches		
8	Does every switch statement have a default?		1
	Arithmetic Operations (Exception Handling)		
9	Is overflow or underflow possible during a computation?		1
	Documentation		
10	Lack of comments.		4
11	All comments are consistent with the code		1

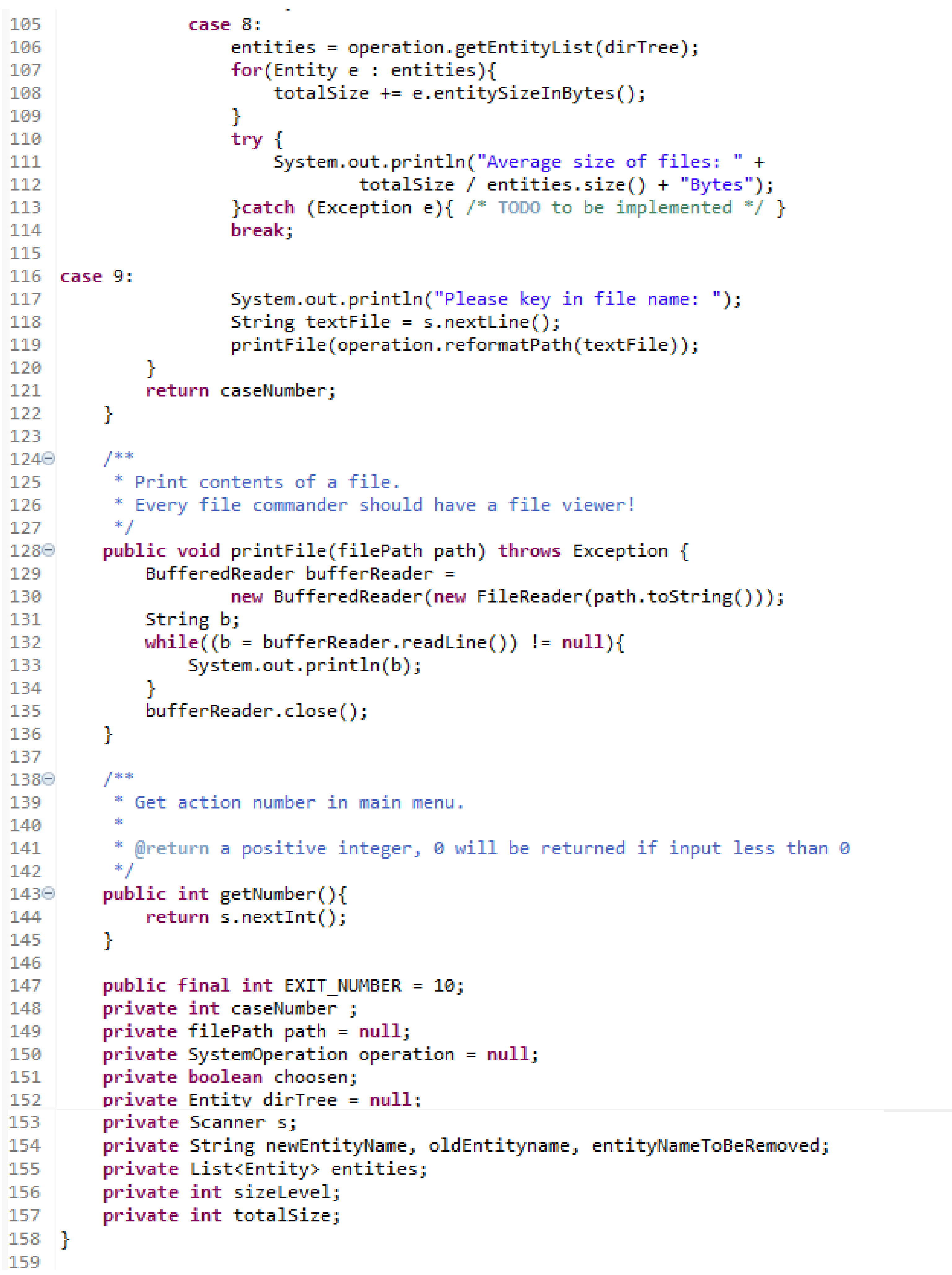
GN 5-2 Coding Standard

No	Item to check	Checked	Hint
1	White space and empty line		1
2	Indention		1
3	Comments		1
4	Naming rules		2

Appendix B - fileCLI.java



```java
import java.io.*;

/**
 * A Command-Line Interface to work with files.
 */
public class filesCLI{

    /**
     * filesCLI constructor for initializing components
     */
    public filesCLI() {
        sizeLevel = 0;
    }

    public void mainFlow() throws Exception {
        while(true) {
            mainMenu();
            if(actNumber(getNumber()) == EXIT_NUMBER){
                break;
            }
        }
    }

    /**
     * Print menu prompt for user.
     */
    public void mainMenu() throws Exception {
        System.out.print("1. Select directory path \n" + "2. List directory\n" +
                "3. Create a file/directory\n" +
                "4. Remove a file/directory\n" + "5. Rename a file/directory\n" +
                "6. Count total size of files under the directory\n" +
                "7. Count number of files and directories under the directory\n" +
                "8. Average file size in the directory\n" +
                "9. Print contents of a file\n" +
                "9. Exit\n" +
                "Command\n>");
    }

    public Date genCurYmdhms() {
        return new Date();
    }

    /**
     * Accept caseNumber inputted by user and do the correspond work.
     *
     * @param caseNumber the number of task to do
     */
    public int actNumber(int caseNumber) throws Exception {
        switch (caseNumber) {
            case 1:
                if (choosen) {
                    dirTree = new Entity();
                }
                // Set directory path
                System.out.print("Please key in a directory path: ");

   String inputPath;
                inputPath = s.nextLine();
                path = operation.reformatPath(inputPath);
                dirTree = operation.setWorkingPath(path);
                choosen = true;
                break;
            case 2:
                // List all files and folders in the folder chosen
                operation.listAllEntitiesIn(dirTree);
                break;
            case 3:
                // Create a new folder or file
                System.out.print("Please key in Entity name: ");
                newEntityName = s.nextLine();
                System.out.println();
                System.out.print("Please choose Entity type[1:file/2:folder]:");
                int typeNum = s.nextInt();
                operation.createEntity(newEntityName, typeNum, path, dirTree);
                break;
            case 4:
                //Remove an existing folder or file
                System.out.print("Please key in Entity name: ");
                entityNameToBeRemoved = s.nextLine();
                operation.purgeEntity(entityNameToBeRemoved, dirTree, path);
                break;
            case 5:
                //Modify name of an existing folder of file
                System.out.print("Please key in Entity name: ");
                oldEntityname = s.nextLine();
                System.out.print("Please key in the new name: ");
                newEntityName = s.nextLine();
                operation.mn(oldEntityname, newEntityName, dirTree);
                break;
            case 6:
                //Space occupied by the folder
                entities = operation.getEntityList(dirTree);
                for(Entity e : entities){
                    totalSize += e.entitySizeInBytes();
                }
                System.out.println("Total size of files: " + totalSize + "Bytes");
                break;
            case 7:
                //Calculate how many folders and files are there in the folder
                entities = operation.getEntityList(dirTree);
                System.out.println("Files and directories count in this folder: " +
                        entities.size());
                break;
```



```
            case 8:
                entities = operation.getEntityList(dirTree);
                for(Entity e : entities){
                    totalSize += e.entitySizeInBytes();
                }
                try {
                    System.out.println("Average size of files: " +
                            totalSize / entities.size() + "Bytes");
                }catch (Exception e){ /* TODO to be implemented */ }
                break;

case 9:
                System.out.println("Please key in file name: ");
                String textFile = s.nextLine();
                printFile(operation.reformatPath(textFile));
        }
        return caseNumber;
    }

    /**
     * Print contents of a file.
     * Every file commander should have a file viewer!
     */
    public void printFile(filePath path) throws Exception {
        BufferedReader bufferReader =
                new BufferedReader(new FileReader(path.toString()));
        String b;
        while((b = bufferReader.readLine()) != null){
            System.out.println(b);
        }
        bufferReader.close();
    }

    /**
     * Get action number in main menu.
     *
     * @return a positive integer, 0 will be returned if input less than 0
     */
    public int getNumber(){
        return s.nextInt();
    }

    public final int EXIT_NUMBER = 10;
    private int caseNumber ;
    private filePath path = null;
    private SystemOperation operation = null;
    private boolean choosen;
    private Entity dirTree = null;
    private Scanner s;
    private String newEntityName, oldEntityname, entityNameToBeRemoved;
    private List<Entity> entities;
    private int sizeLevel;
    private int totalSize;
}
```

Appendix C - Bad Smells Record (GN 5-1)

編號 No.	分類 Check List No.	位置 (行數) Line #	說明 Description
1	G1		
2	G2		
3	G3		
4	G4		
5	G5		
6	G6		
7	G7		
8	G8		
9	G9		
10	G10		
11	G11		
12	G12		
13	G13		
14	G14		
15	G15		
16	G16		
17	G17		
18	G18		
19	G19		
20	G20		
21	G21		
22	G22		
23	G23		
24			
25			
26			
27			
28			
29			
30			
31			
32			
33			

Appendix D - Coding Standard Violations Record (GN 5-2)

編號 No.	分類 Check List No.	位置 (行數) Line #	說明 Description
1	G1		
2	G2		
3	G3		
4	G4		
5	G5		
6	G6		
7	G7		
8	G8		
9	G9		
10	G10		
11	G11		
12	G12		
13	G13		
14	G14		
15	G15		
16	G16		
17	G17		
18	G18		
19	G19		
20	G20		
21	G21		
22	G22		
23	G23		
24			
25			
26			
27			
28			
29			
30			
31			