

Appendix to Mandatory assignment for Camilla Stenberg (camilest) and Esben Slaatto (esbenss)

Sample code from FileReaderWriter.java

Code example from FileReaderWriter.java **before** refactoring:

```
public void openFiletoRead() {
    try {
        {
            if(true) {
                input = new ObjectInputStream(new FileInputStream("players.ser"));
            }
        }
    } catch (IOException ioException) {
        System.err.println("Error opening file.");
    }
}

public void readRecords() {
    Players records;

    try // input the values from the file
    {
        Object obj = null;

        while (!(obj = input.readObject()).equals(null)) {
            if (obj instanceof Players) {
                records = (Players) obj;{
                    myArr.add(records);{
                        System.out.printf("DEBUG: %-10d%-12s\n",
                            records.getScores(), records.getName());
                    }
                }
            }
        }

        /*
        * while (true) { records = (Players) input.readObject();
        * myArr.add(records); System.out.printf("DEBUG: %-10d%-12s\n",
        * records.getScores(), records.getName()); } // end while
        */
    } // end try
    catch (EOFException endOfFileException) {
        return; // end of file was reached
    } catch (ClassNotFoundException classNotFoundException) {
        System.err.println("Unable to create object.");
    } catch (IOException ioException) {
        System.err.println("Error during reading from file.");
    }
}
```

The same Sample Code from FileReaderWriter.java **after** refactoring:

```
public void openFiletoRead() {
    try {
        input = new ObjectInputStream(new FileInputStream("players.ser"));
    } catch (IOException ioException) {
        System.err.println("Error opening file.");
    }
}

// read all objects from 'players.ser' and store objects in array 'myArr'.
public void readRecords() {
    Players records;

    try // input the values from the file
    {
        Object obj = null;

        // as long as there are more Player objects
        while ((!(obj = input.readObject()).equals(null)) && obj instanceof Players ) {

            records = (Players) obj;
            myArr.add(records);
            System.out.println("test");
            System.out.printf("DEBUG: %-10d%-12s\n",
                               records.getScores(), records.getName());
        }

    } // end try
    catch (EOFException endOfFileException) {
        return; // end of file was reached
    } catch (ClassNotFoundException classNotFoundException) {
        System.err.println("Unable to create object.");
    } catch (IOException ioException) {
        System.err.println("What? Error during reading from file.");
    }
}
```

Sample code from Game.java

Code example from FileReaderWriter.java **before** refactoring:

```
// End method
private void menu(String letter) {
    if (letter.equals(Command.restart.toString())) {
        new Game(true);
    } else {
        if (letter.equals(Command.top.toString())) {
            filerw.openFiletoRead();
            filerw.readRecords();
            filerw.closeFileFromReading();
            filerw.printAndSortScoreBoard();
            new Game(true);
        } else {
            if (letter.equals(Command.exit.toString())) {
                System.exit(1);
            }
        }
    }
}
```

Code example from FileReaderWriter.java **after** refactoring:

```
// perform menu-action
private void menu(String letter) {
    if (letter.equals(Command.restart.toString())) {
        new Game(true);
    } else if (letter.equals(Command.top.toString())) {
        filerw.openFiletoRead();
        filerw.readRecords();
        filerw.closeFileFromReading();
        filerw.printAndSortScoreBoard();
        new Game(true);
    } else if (letter.equals(Command.exit.toString())) {
        System.exit(1);
    }
}
```