



Diseño de Software

Paralelo 1

Tarea SOLID

GRUPO 4

Integrantes:

Acosta Suquitana Savier Alessandro

Poveda Gorotiza Jeremy Rodrigo

Pacheco Reyes Genesis Gabriela

Moreira Merino Marcos Alejandro

Repositorio 3

Link: <https://github.com/RobertTunn1983/GraphicalUserInterfaces-ModelViewController/tree/main/src/main>

Violación del principio **ISP**:

En el archivo Order.java se encuentra la clase Order, esta implementa la interfaz IOrder que define los métodos addItem(), total(), writeToFile(), print(), sortByName(), sortByPrice() y compareTo(), todos estos métodos se implementan sin ningún problema, sin embargo, el método compareTo() carece de sentido ya que no se le aplica una lógica concreta que justifique su implementación.

```
4
5 package uk.ac.rgu.cm2100;
6
7 import java.io.PrintWriter;
8 import java.io.File;
9 import java.util.ArrayList;
10 import java.util.Collections;
11 import java.util.Comparator;
12 import java.util.Date;
13 import java.util.concurrent.TimeUnit;
14
15 public class Order implements IOrder {
16
17     ArrayList<IMenuItem> items;
18
19     public Order() {...3 lines }
20
21     @Override
22     public void addItem (IMenuItem item) {...}
23
24     public void printItems () {...6 lines }
25
26     @Override
27     public int total() {...10 lines }
28
29     @Override
30     public void writeToFile() {...87 lines }
31
32     @Override
33     public void print() {...19 lines }
34
35     @Override
36     public void sortByName() {...12 lines }
37
38     @Override
39     public void sortByPrice() {...8 lines }
40
41     @Override
42     public int compareTo(IMenuItem o) {
43         return 0;
44     }
45 }
```

```

L */
package uk.ac.rgu.cm2100;

public interface IOrder extends Comparable <IMenuItem> {

    public void addItem(IMenuItem item);

    public int total();

    public void print();

    public void writeToFile();

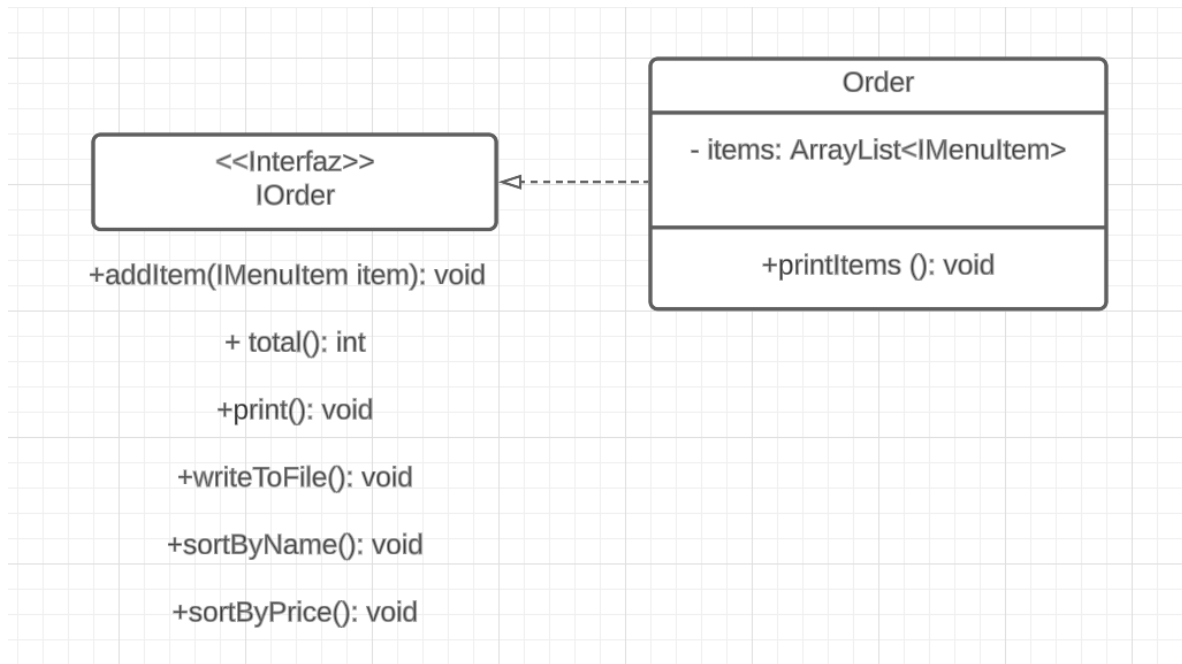
    public void sortByName();

    public void sortByPrice();
}

```

Solución:

Para resolver esta violación del ISP, podríamos eliminar la extensión de Comparable de la interfaz IOrder y moverla a otra interfaz llamada IComparableOrder que las clases que la necesiten las pueden implementar (teniendo la posibilidad de ser comparables entre si). De esta manera, Order ya no tendría que implementar el método compareTo() que no utiliza:



```
*/
package uk.ac.rgu.cm2100;

public interface IOrder {

    public void addItem(IMenuItem item);

    public int total();

    public void print();

    public void writeToFile();

    public void sortByName();

    public void sortByPrice();

}
```

```

    */
    public interface IComparableOrder extends Comparable <IMenuItem> {
        public void addItem(IMenuItem item);

        public int total();

        public void print();

        public void writeToFile();

        public void sortByName();

        public void sortByPrice();
    }

```

Violación del principio ISP:

En este caso podemos observar que la interfaz Initializable está implementada, sin embargo, dicha implementación está literalmente vacía, si consideramos que el método sobrescrito initialize no tiene ninguna utilidad para la clase Controller_1_WelcomePane, podríamos argumentar que esta clase no debería implementar la interfaz Initializable, entonces Controller_1_WelcomePane no tendría que implementar el método initialize lo que podría considerarse una mejora en términos del ISP.

```

public class Controller_1_WelcomePane extends Model implements Initializable {

    @Override
    public void initialize(URL url, ResourceBundle resourceBundle) {
        //Empty
    }

    //Switch to Create Order AnchorPane
    public void goToCreateOrder2(ActionEvent event) throws IOException {...8 lines }

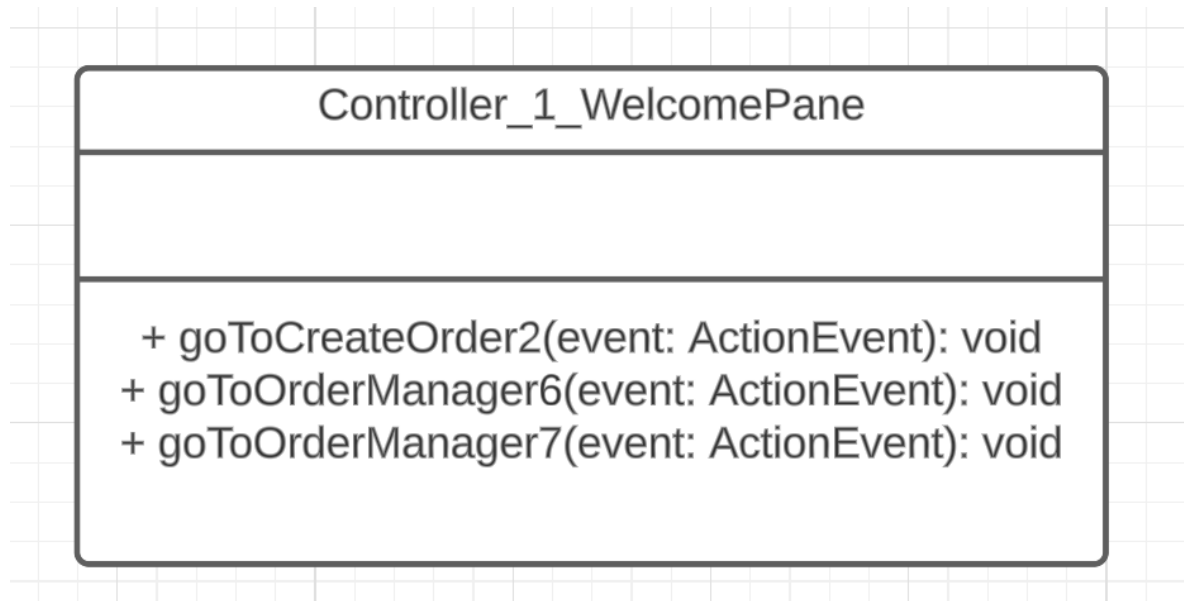
    //Switch to Order Manager AnchorPane
    public void goToOrderManager6 (ActionEvent event) throws IOException {...8 lines }

    //Switch to Create Order AnchorPane
    public void goToOrderManager7(ActionEvent event) throws IOException {...8 lines }
}

```

Solución:

Para resolver este problema simplemente dejamos de implementar la interfaz Initializable y eliminamos el método correspondiente.



```
public class Controller 1 WelcomePane extends Model {  
    //Switch to Create Order AnchorPane  
    public void goToCreateOrder2(ActionEvent event) throws IOException {...8 lines }  
    //Switch to Order Manager AnchorPane  
    public void goToOrderManager6 ((ActionEvent event) throws IOException {...8 lines }  
    //Switch to Create Order AnchorPane  
    public void goToOrderManager7 ((ActionEvent event) throws IOException {...8 lines }  
}
```