```
                list.addLine (s);
            }
        }
}
```

Then, at the top programming level, we just create instances of a table and a list using the listBridge class.

```
//add in customer view as list box
        listBridge lList = new listBridge(new productList());
        lList.add Data (prod);
        pleft.add("North", new JLabel("Customer view"));
        pleft.add("Center," lList);

        //add in executive view as table
        listBridge lTable = new listBridge(new productTable());
        lTable.addData (prod);
        pright.add("North", new JLabel("Executive view"));
        pright.add("Center", lTable);
```

The Class Diagram

The UML diagram in Figure 10.2 for the Bridge pattern shows the separation of the interface and the implementation. The Bridger class on the left is the Abstraction, and the listBridge class is the implementation of that abstraction. The visList interface describes the public interface to the list classes productList and productTable. The visList interface defines the interface of the Implementor, and the Concrete Implementors are the productList and productTable classes.

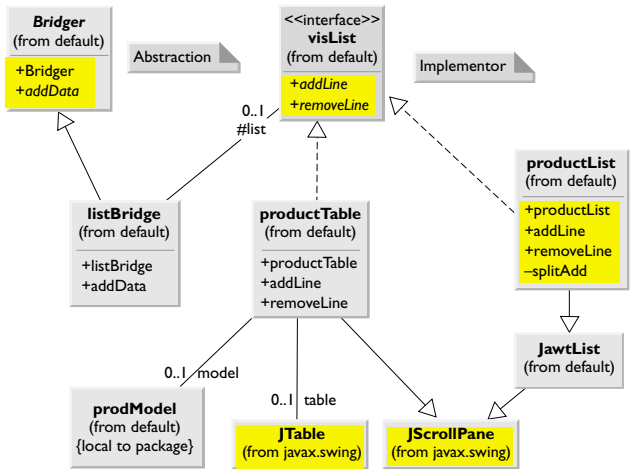

Figure 10.2 The UML diagram for the Bridge pattern used in the two displays of product information.