

## LAB ASSIGNMENT 9

### ADAPTER DESIGN PATTERN

SHRUTI MISHRA

21BCP110

DIV-2 G4

CODE:

❖ **Inside adapterDP package: -**

- Client.java //main file

```
package adapterDP;

public class Client {
    public static void main(String args[]) {

        Instrument_Shop ishop = new Instrument_Shop();

        //Selling New instruments
        ishop.play("Electric", "Guitar");
        ishop.play("Bass", "Xylophone");

        //Selling Old instruments
        ishop.play("Acoustic", "Violin");
    }
}
```

➤ **2 interfaces:**

- Old\_Instrument\_Interface.java

```
package adapterDP;

public interface Old_Instrument_Interface {
    public void play(String audioType, String filename);
}
```

- New\_Instrument\_Interface.java

```
package adapterDP;

public interface New_Instrument_Interface {
    public void playElectric(String fileName);
    public void playBass(String fileName);
}
```

➤ **2 new Instrument types to get in the shop:**

- Electric\_Instrument.java

```
package adapterDP;

public class Electric_Instrument implements New_Instrument_Interface{

    public void playElectric(String filename) {
        System.out.println("Selling NEW-Generation Electric " + filename);
    }

    public void playBass(String fileName) {
        //do nothing
    }
}
```

- Bass\_Instrument.java

```
package adapterDP;

public class Bass_Instrument implements New_Instrument_Interface{

    public void playBass(String filename) {
        System.out.println("Selling NEW-Generation Bass " + filename);
    }

    public void playElectric(String filename) {
        //do nothing
    }
}
```

➤ **InstrumentAdapter.java:**

```
package adapterDP;

public class InstrumentAdapter implements Old_Instrument_Interface {
    New_Instrument_Interface new_instrument_interface;

    public InstrumentAdapter(String audioType) {
        if (audioType.equalsIgnoreCase("Bass")) {
            new_instrument_interface = new Bass_Instrument();
        } else if (audioType.equalsIgnoreCase("Electric")) {
            new_instrument_interface = new Electric_Instrument();
        }
    }
}
```

```

    }
}

@Override
public void play(String audioType, String fileName) {

    if (audioType.equalsIgnoreCase("Electric")) {
        new_instrument_interface.playElectric(fileName);
    } else if (audioType.equalsIgnoreCase("Bass")) {
        new_instrument_interface.playBass(fileName);
    }
}
}
}

```

➤ **Instrument\_Shop.java:**

```

package adapterDP;

public class Instrument_Shop implements Old_Instrument_Interface {

    InstrumentAdapter instrumentadapter;

    @Override
    public void play(String audioType, String fileName) {

        if (audioType.equalsIgnoreCase("Acoustic")) {
            System.out.println("Selling OLD-Generation Acoustic " + fileName);
        }

        // instrumentadapter is providing support to sell other type of
instruments
        else if (audioType.equalsIgnoreCase("Electric") ||
audioType.equalsIgnoreCase("Bass")) {
            instrumentadapter = new InstrumentAdapter(audioType);
            instrumentadapter.play(audioType, fileName);
        }

        else {
            System.out.println("Invalid Instrument Type. " + audioType + "
Product Not Available currently");
        }
    }
}
}

```

## OUTPUT:

```
"C:\Users\Shruti Mishra\.jdk\openjdk-18.0.2.1\bin\java.exe" "-javaagent:  
Selling NEW-Generation Electric Guitar  
Selling NEW-Generation Bass Xylophone  
Selling OLD-Generation Acoustic Violin  
  
Process finished with exit code 0
```