**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:

