**practice-5.3**

Code:

import javax.swing.JOptionPane;

import java.util.Date;

// ProductionLineDemo class

public class ProductionLineDemo {

// Item interface

interface Item {

String MANUFACTURER = "OracleProduction";

void setProductionNumber(int number);

void setName(String name);

String getName();

Date getManufactureDate();

int getSerialNumber();

}

// ItemType enum

enum ItemType {

AUDIO("AU"), VISUAL("VI"), AUDIO\_MOBILE("AM"), VISUAL\_MOBILE("VM");

private final String code;

ItemType(String code) {

this.code = code;

}

public String getCode() {

return code;

}

}

// Product class implementing Item interface

abstract static class Product implements Item {

private static int currentProductionNumber = 1;

private int serialNumber;

private String manufacturer;

private Date manufacturedOn;

private String name;

public Product(String name) {

this.name = name;

this.serialNumber = currentProductionNumber++;

this.manufacturedOn = new Date();

this.manufacturer = Item.MANUFACTURER;

}

@Override

public void setProductionNumber(int number) {

this.serialNumber = number;

}

@Override

public void setName(String name) {

this.name = name;

}

@Override

public String getName() {

return name;

}

@Override

public Date getManufactureDate() {

return manufacturedOn;

}

@Override

public int getSerialNumber() {

return serialNumber;

}

@Override

public String toString() {

return "Manufacturer : " + manufacturer + "\n" +

"Serial Number : " + serialNumber + "\n" +

"Date : " + manufacturedOn + "\n" +

"Name : " + name;

}

}

// MultimediaControl interface

interface MultimediaControl {

void play();

void stop();

void previous();

void next();

}

// AudioPlayer class extending Product and implementing MultimediaControl

static class AudioPlayer extends Product implements MultimediaControl {

private String audioSpecification;

private ItemType mediaType;

public AudioPlayer(String name, String audioSpecification) {

super(name);

this.audioSpecification = audioSpecification;

this.mediaType = ItemType.AUDIO; // Default type

}

@Override

public void play() {

System.out.println("Playing");

}

@Override

public void stop() {

System.out.println("Stopped");

}

@Override

public void previous() {

System.out.println("Previous track");

}

@Override

public void next() {

System.out.println("Next track");

}

@Override

public String toString() {

return super.toString() + "\n" +

"Audio Spec : " + audioSpecification + "\n" +

"Type : " + mediaType.getCode();

}

}

// Random class for demonstration of GUI

static class Random {

public static void main(String[] args) {

if (!java.awt.GraphicsEnvironment.isHeadless()) {

int rollOfDice = (int) (Math.random() \* 6) + 1;

String output = "You rolled a " + rollOfDice;

JOptionPane.showMessageDialog(null, output, "Random Number Demo",

JOptionPane.INFORMATION\_MESSAGE);

} else {

System.out.println("Running in a headless environment. Cannot display GUI.");

int rollOfDice = (int) (Math.random() \* 6) + 1;

System.out.println("You rolled a " + rollOfDice);

}

System.exit(0);

}

}

public static void main(String[] args) {

// Demonstrate usage of AudioPlayer

AudioPlayer player = new AudioPlayer("Cool Music Player", "High Bass");

System.out.println(player);

player.play();

player.stop();

player.previous();

player.next();

// Demonstrate Random class

Random.main(args);

}

}

Output:

