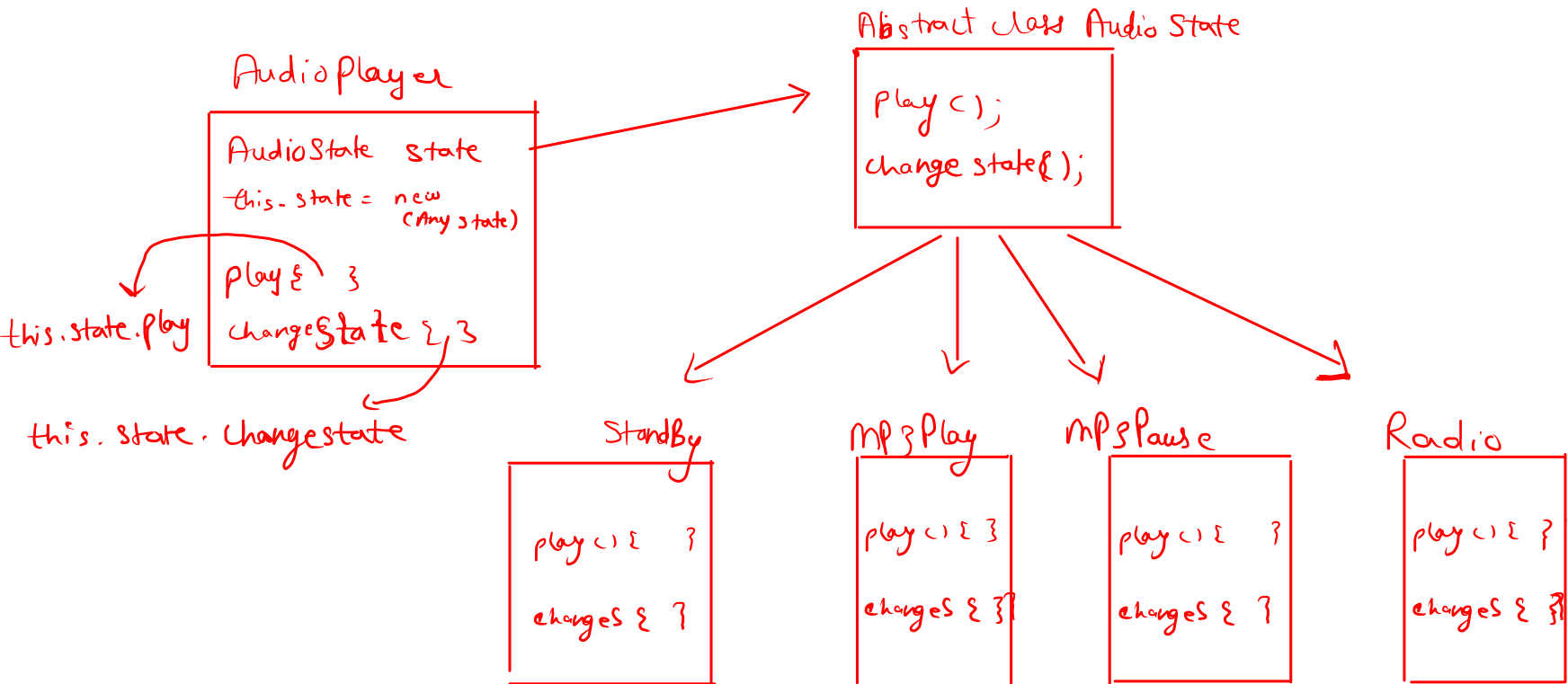


State :- Behavioural Design Pattern.



```

class Test {
    Run | Debug
    public static void main(String[] args) {
        AudioPlayer player = new AudioPlayer();

        player.pressPlay();
        player.pressChangeSource();
        player.pressPlay();
        player.pressChangeSource();
        player.pressPlay();
        player.pressPlay();
    }
}

```

```

public class AudioPlayer {
    AudioPlayerState state;

    AudioPlayer() {
        this.state = new MP3StandByState();
    }

    public void pressPlay() {
        this.state.pressPlay(this); //calling AudioPlayer's object's AudioPlayerState object reference's
    }                                     // pressPlay fuction and passing this AudioPlayer object to it.

    public void pressChangeSource() {
        this.state.pressChangeSource(this);
    }
}

```

composition

default state

```

public abstract class AudioPlayerState {
    public abstract void pressPlay(AudioPlayer p);
    public abstract void pressChangeSource(AudioPlayer p);
}

```

Single Responsibility classes of States.

extends Inheritance

```

public class MP3PlayingState extends AudioPlayerState {
    @Override
    public void pressPlay(AudioPlayer p) {
        System.out.println("Paused Mp3");
        p.state = new MP3PausedState();
    }

    @Override
    public void pressChangeSource(AudioPlayer p) {
        System.out.println("Setting source to radio");
        p.state = new RadioState();
    }
}

```

```

public class MP3PausedState extends AudioPlayerState {
    @Override
    public void pressPlay(AudioPlayer p) {
        System.out.println("Playing Mp3");
        p.state = new MP3PlayingState();
    }

    @Override
    public void pressChangeSource(AudioPlayer p) {
        System.out.println("Setting Source to Radio");
        p.state = new RadioState();
    }
}

```

```

public class RadioState extends AudioPlayerState {
    @Override
    public void pressPlay(AudioPlayer p) {
        System.out.println("Changing the Station");
    }

    @Override
    public void pressChangeSource(AudioPlayer p) {
        System.out.println("Going to MP3");
        p.state = new MP3PausedState();
    }
}

```

```

public class MP3StandByState extends AudioPlayerState {
    @Override
    public void pressPlay(AudioPlayer p) {
        System.out.println("Invalid Operation");
    }

    @Override
    public void pressChangeSource(AudioPlayer p) {
        System.out.println("Setting Sorce to Radio");
        p.state = new RadioState();
    }
}

```