**State Design Pattern**

**Assignment - 1**

Name - Heet Dobariya Roll No. - 22BCP177 Group - G5

* **State Design Pattern :**

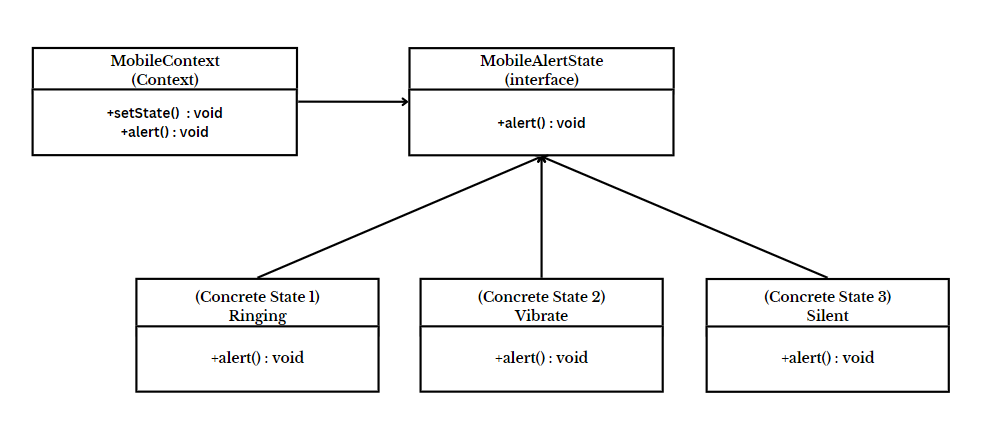
The State Design Pattern is a behavioral design pattern that allows an object to change its behavior when its internal state changes. It is useful when an object's behavior depends on its state and needs to change dynamically based on the state it's in.

The key idea behind the State pattern is to encapsulate each state of an object into a separate class, and the object itself holds a reference to the current state. This way, the object can delegate behavior to the current state class, and when its state changes, it can switch to a different state class seamlessly.

Here's a basic structure of the State Design Pattern:

1. Context: This is the class that maintains the current state and delegates the behavior to the current state object.
2. State: This is an interface or abstract class that defines a set of methods representing the behavior of the context object. Each concrete state subclass implements these methods.
3. Concrete States: These are the concrete implementations of the State interface, each representing a different state of the context object. They implement behavior specific to their state and may trigger transitions to other states.

* **Program :** Implement state design pattern for Mobile example.
* **UML Diagram :**



* **Code :**

interface MobileAlertState

{

public void alert();

}

class Ringing implements MobileAlertState

{

public void alert()

{

System.out.println("Phone is in Ringing Mode.");

}

}

class Silent implements MobileAlertState

{

public void alert()

{

System.out.println("Phone is in Silent Mode.");

}

}

class Vibrate implements MobileAlertState

{

public void alert()

{

System.out.println("Phone is in Vibrate Mode.");

}

}

class MobileContext

{

private MobileAlertState currentState;

public MobileContext()

{

currentState = new Ringing();

}

public void setState(MobileAlertState state)

{

currentState = state;

}

public void alert()

{

currentState.alert();

}

}

class MobileState

{

public static void main(String[] args)

{

MobileContext mc = new MobileContext();

mc.alert();

mc.setState(new Vibrate());

mc.alert();

mc.setState(new Silent());

mc.alert();

}

}

* **Output :**

