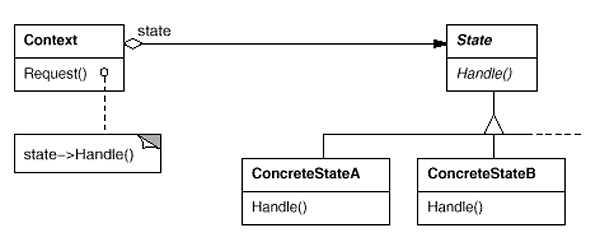
State Design Pattern

GOF : **Allow an object to alter its behavior when its internal state changes. The object will appear to change its class.**

**Structure Diagram**



Java Code : Example is mobile state silent, vibration

**public** **interface** MobileState {

**void** display(Context context);

}

**public** **class** Context {

**private** MobileState state;

**public** MobileState getState() {

**return** state;

}

**public** **void** setState(MobileState state) {

**this**.state = state;

}

**public** **void** request() {

state.display(**this**);

}

}

Class Diagram

**public** **class** Test {

**public** **static** **void** main(String[] args) {

Context context = **new** Context();

context.setState( **new** SilentState());

context.request();

context.setState( **new** VibrationState());

context.request();

}

}

**public** **class** SilentState **implements** MobileState {

@Override

**public** **void** display(Context context) {

System.***out***.println("Silent...");

}

}

**public** **class** VibrationState **implements** MobileState {

@Override

**public** **void** display(Context context) {

System.***out***.println("vibration...");

}

}

