DDR Assignment 4

Salman Ahmed Khan

19K-1043 BS SE (A)

Q1.

State Pattern

State is a behavioral design pattern that lets an object alter its behavior when its internal state changes. It appears as if the object changed its class.

In State pattern a class behavior changes based on its state. This type of design pattern comes under behavior pattern.

In State pattern, we create objects which represent various states and a context object whose behavior varies as its state object changes.

public interface State {

public void doAction(Context context);

}

Step 2

Create concrete classes implementing the same interface.

StartState.java

public class StartState implements State {

public void doAction(Context context) {

System.out.println("Player is in start state");

context.setState(this);

}

public String toString(){

return "Start State";

}

}

StopState.java

public class StopState implements State {

public void doAction(Context context) {

System.out.println("Player is in stop state");

context.setState(this);

}

public String toString(){

return "Stop State";

}

}

Step 3

Create Context Class.

Context.java

public class Context {

private State state;

public Context(){

state = null;

}

public void setState(State state){

this.state = state;

}

public State getState(){

return state;

}

}

Step 4

Use the Context to see change in behaviour when State changes.

StatePatternDemo.java

public class StatePatternDemo {

public static void main(String[] args) {

Context context = new Context();

StartState startState = new StartState();

startState.doAction(context);

System.out.println(context.getState().toString());

StopState stopState = new StopState();

stopState.doAction(context);

System.out.println(context.getState().toString());

}

}

State can be considered as an extension of Strategy. Both patterns are based on composition: they change the behavior of the context by delegating some work to helper objects. Strategy makes these objects completely independent and unaware of each other. However, State doesn’t restrict dependencies between concrete states, letting them alter the state of the context at will.

Q2

KISS

It means Keep It Simple Stupid.

keep the code simple and clear, making it easy to understand. After all, programming languages are for humans to understand — computers can only understand 0 and 1 — so keep coding simple and straightforward. Keep your methods small. Each method should never be more than 40-50 lines.

Each method should only solve one small problem, not many use cases. If you have a lot of conditions in the method, break these out into smaller methods. It will not only be easier to read and maintain, but it can help find bugs a lot faster.