

App.java

```
package com.github.andredias20.observer;

import com.github.andredias20.observer.fusca.motors.Motor1500;
import com.github.andredias20.observer.listenerOldStyle.MotorEvent;
import com.github.andredias20.observer.listenerOldStyle.MotorListener;
import com.github.andredias20.observer.listenerOldStyle.MotorListenerAdapter;
import com.github.andredias20.observer.listenerOldStyle.ObservableMotor;
import com.github.andredias20.observer.motor.Motor;
import com.github.andredias20.observer.motor.MotorState;

public class App {
    static MotorListener motorListener = new MotorListener() {
        @Override
        public void beforeStart(MotorEvent me) {
            System.out.println("beforeStart " + MotorState.stringify(me.state));
        }

        @Override
        public void afterStart(MotorEvent me) {
            System.out.println("afterStart " + MotorState.stringify(me.state));
        }

        @Override
        public void beforeStop(MotorEvent me) {
            System.out.println("beforeStop " + MotorState.stringify(me.state));
        }

        @Override
        public void afterStop(MotorEvent me) {
            System.out.println("afterStart " + MotorState.stringify(me.state));
        }

        @Override
        public void beforeSlowDown(MotorEvent me) { /*Nothing*/ }

        @Override
        public void afterSlowDown(MotorEvent me) { /*Nothing*/ }

        @Override
        public void beforeSpeedUp(MotorEvent me) { /*Nothing*/ }

        @Override
        public void afterSpeedUp(MotorEvent me) { /*Nothing*/ }
    };
    static MotorListener speedMotorListener = new MotorListenerAdapter(){
        @Override
        public void beforeSpeedUp(MotorEvent me) {
            System.out.println("beforeSpeedUp "+ MotorState.stringify(me.state));
        }
        @Override
        public void afterSpeedUp(MotorEvent me) {
            System.out.println("afterSpeedUp "+ MotorState.stringify(me.state));
        }
    };

    public static void main(String[] args) {
        Motor motor1500 = new Motor1500();
        ObservableMotor motor = new ObservableMotor(motor1500);

        motor.addMotorListener(motorListener);
        motor.addMotorListener(speedMotorListener);

        motor.start();
        motor.speedUp(0.5f);
        motor.stop();
    }
}
```

```
}
```