State Design Pattern Example

**public** **class** Driver {

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

CarContext car = **new** CarContext();

AccelerationState normal = **new** NormalAccelerationState();

AccelerationState turbo = **new** TurboAccelerationState();

car.setAccelerationState(normal);

car.accelerate();

System.***out***.println("Current speed: " + car.getSpeed());

car.setAccelerationState(turbo);

car.accelerate();

System.***out***.println("Current speed: " + car.getSpeed());

}

}

//Context

**public** **class** CarContext {

**private** AccelerationState accelerationState;

**private** **int** speed = 0;

**public** **void** setAccelerationState(AccelerationState accelerationState) {

**this**.accelerationState = accelerationState;

}

**public** **void** accelerate() {

speed += accelerationState.accelerate();

}

**public** **int** getSpeed() {

**return** speed;

}

}

//Concrete State

**public** **class** TurboAccelerationState **implements** AccelerationState {

@Override

**public** **int** accelerate() {

**return** 10;

}

}

//Concrete State

**public** **class** NormalAccelerationState **implements** AccelerationState {

@Override

**public** **int** accelerate() {

**return** 5;

}

}

//State

**public** **interface** AccelerationState {

**int** accelerate();

}

