**Паттерн Facade.**

**Помещаем последовательность вызовов большой количества методов разных классов в один метод класса.**

**Запуск автомобиля может быть довольно слоен и требовать некоторых усилий**. Например реализации следующих методов объектов разных классов:

airFlowController.takeAir()  
fuelInjector.on()  
fuelInjector.inject()  
starter.start()  
coolingController.setTemperatureUpperLimit(DEFAULT\_COOLING\_TEMP)  
coolingController.run()  
catalyticConverter.on()

Точно так же остановка двигателя также требует нескольких шагов:

fuelInjector.off()  
catalyticConverter.off()  
coolingController.cool(MAX\_ALLOWED\_TEMP)  
coolingController.stop()  
airFlowController.off()

Фасад – это то, что нам здесь нужно. **Всю сложность мы спрячем в двух методах: startEngine() и stopEngine()** .

Давайте посмотрим, как мы можем это реализовать:

public class CarEngineFacade {  
 private static int DEFAULT\_COOLING\_TEMP = 90;  
 private static int MAX\_ALLOWED\_TEMP = 50;  
 private FuelInjector fuelInjector = new FuelInjector();  
 private AirFlowController airFlowController = new AirFlowController();  
 private Starter starter = new Starter();  
 private CoolingController coolingController = new CoolingController();  
 private CatalyticConverter catalyticConverter = new CatalyticConverter();  
  
 public void startEngine() {  
 fuelInjector.on();  
 airFlowController.takeAir();  
 fuelInjector.on();  
 fuelInjector.inject();  
 starter.start();  
 coolingController.setTemperatureUpperLimit(DEFAULT\_COOLING\_TEMP);  
 coolingController.run();  
 catalyticConverter.on();  
 }  
  
 public void stopEngine() {  
 fuelInjector.off();  
 catalyticConverter.off();  
 coolingController.cool(MAX\_ALLOWED\_TEMP);  
 coolingController.stop();  
 airFlowController.off();  
 }

Теперь, **чтобы завести и остановить машину, нам нужно всего 2 строчки кода, вместо 13:**

facade.startEngine();  
*// ...*  
facade.stopEngine();