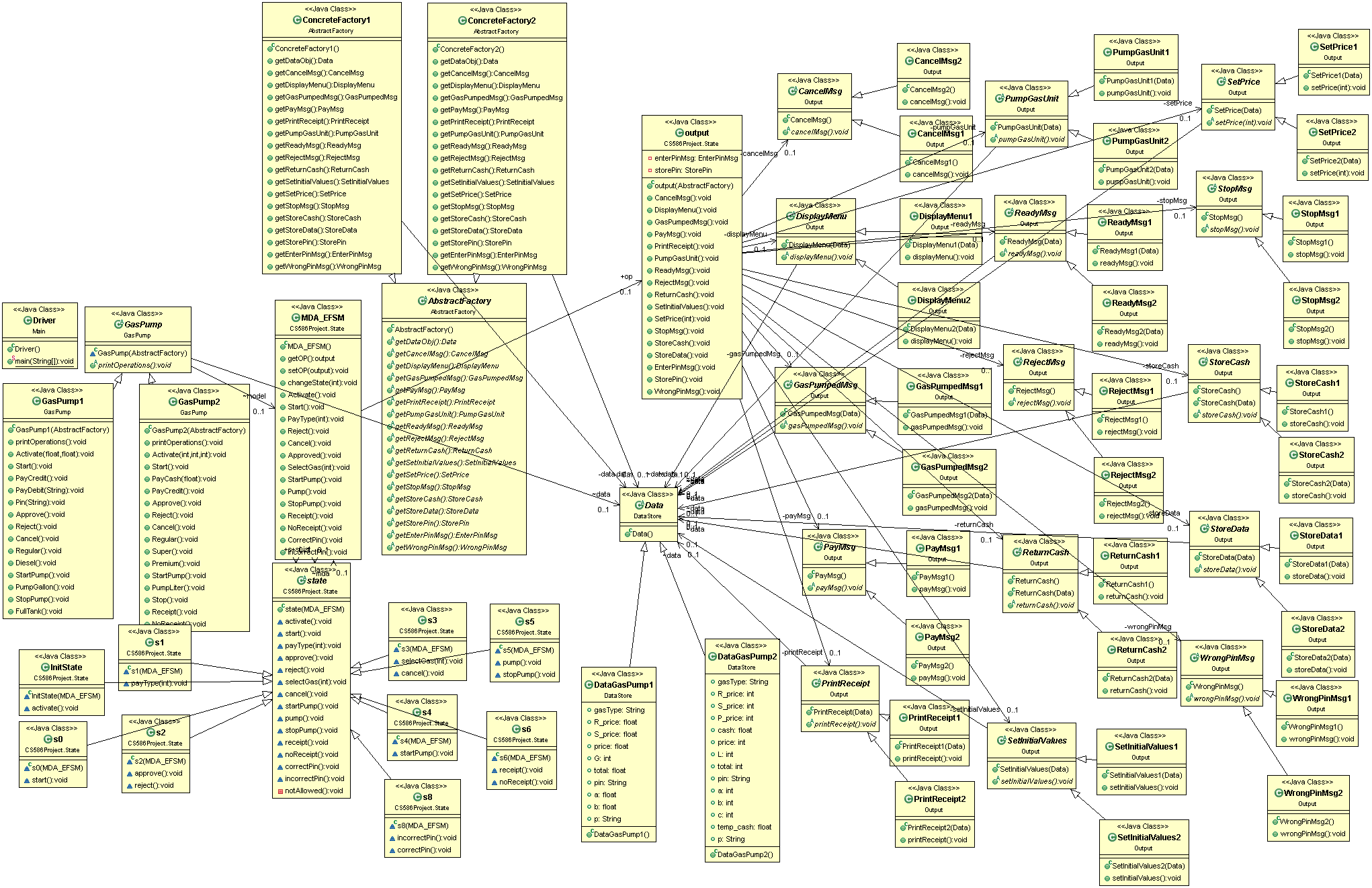
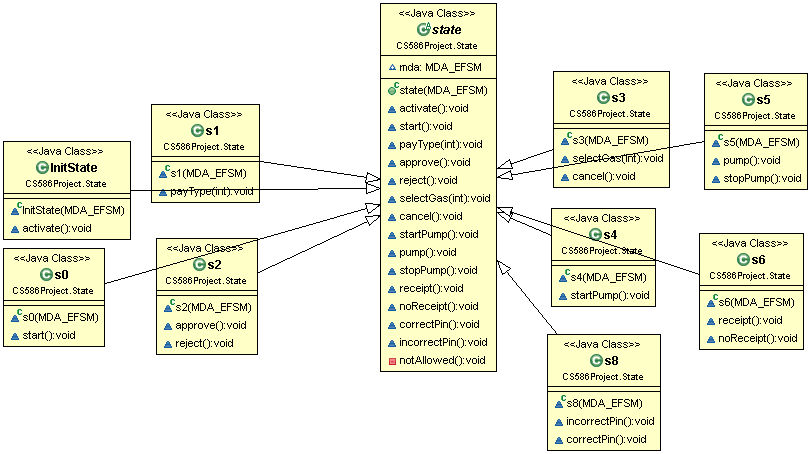
Class Diagram



State Pattern



De-centralized state pattern

State is an abstract state superclass

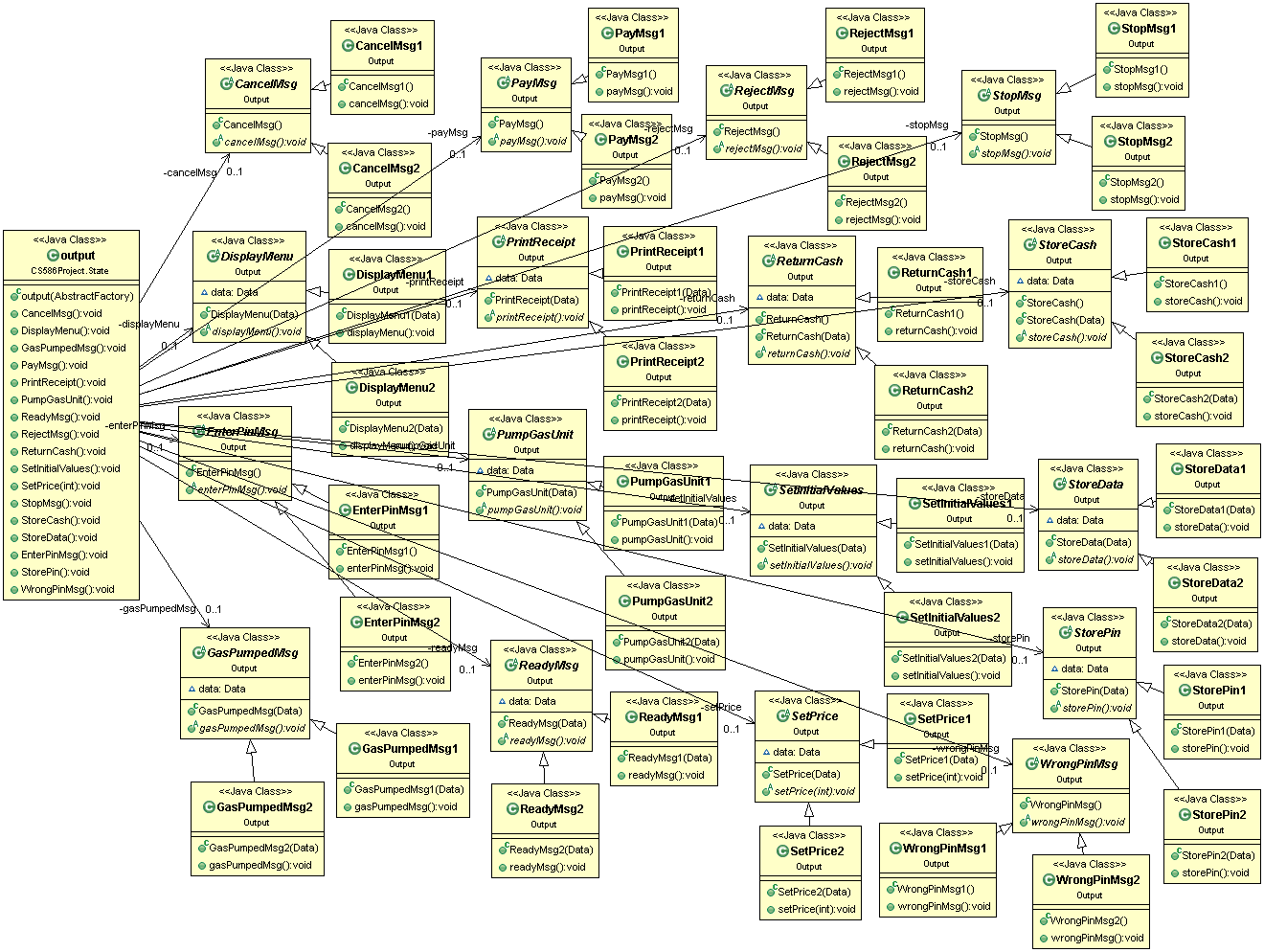
InitState, s0, s1, s2, s3, s4, s5, s6, s8 are the state subclasses

State classes are responsible for performing both actions and state transitions

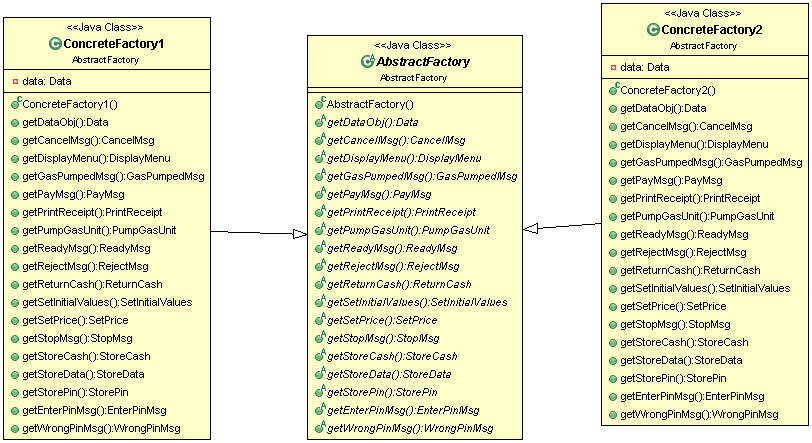
Strategy Pattern

Output is the client class that needs to be initialized with proper action strategies

One class is the abstract strategy that groups different implementations of a specific strategy



Abstract Factory Pattern



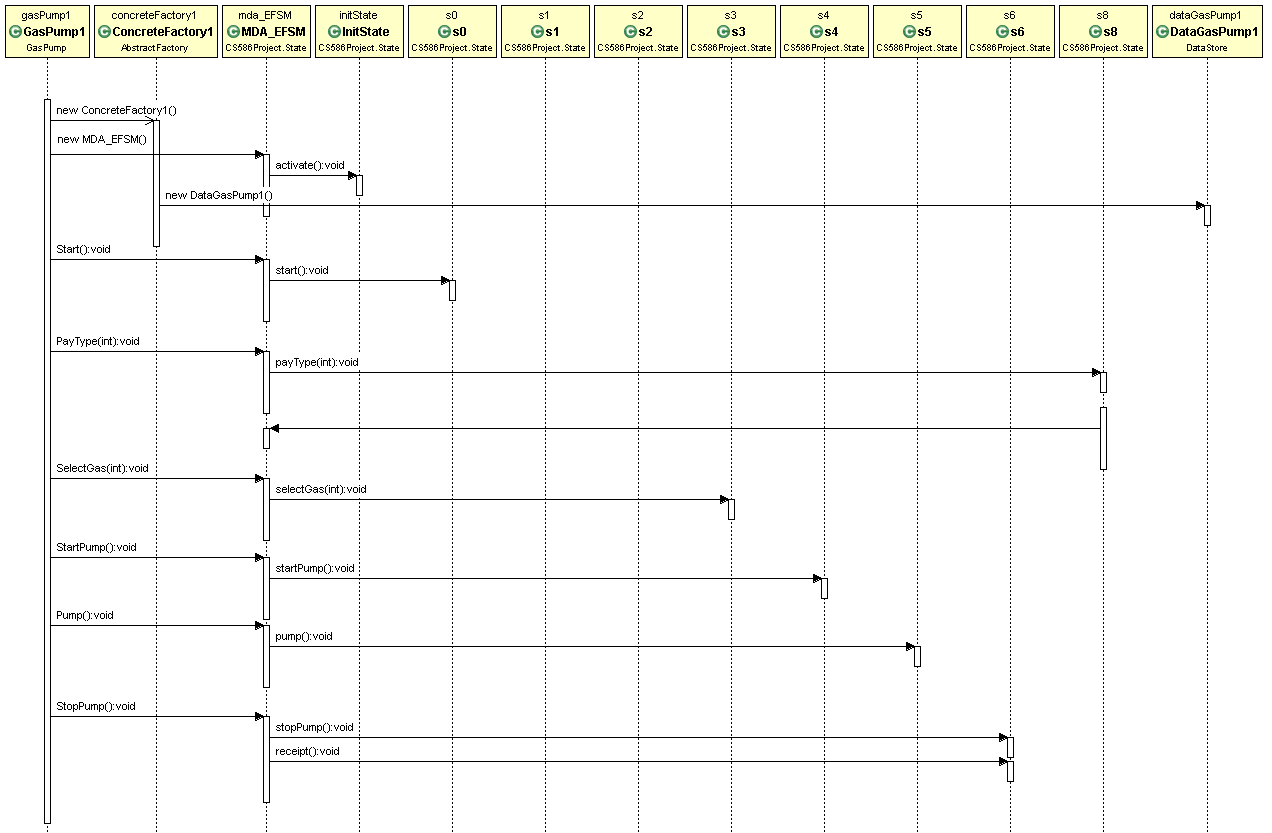
AbstractFactory is the abstract superclass

ConcreteFactory1 is the factory that returns the necessary driver objects for GasPump1

ConcreteFactory2 is the factory that returns the necessary driver objects for GasPump2

Sequence diagrams

Scenario 1



Scenario 2

