Class Description

1. Class Driver (main method)

purpose: The driver for the whole gasPump application, responsible for the gasPump selection.

2. Class GasPump1

purpose: the concrete class for GasPump-1 application, Since we use the abstract factory design pattern, we just import the AbstractFactory.AbstractFactory package, we do not need to classify the specified gasPump factory.

3. Class GasPump2

purpose: the concrete class for GasPump-2 application, Since we use the abstract factory design pattern, we just import the AbstractFactory.AbstractFactory package, we do not need to classify the specified gasPump factory.

4. Class AbstractFactory (Abstract Class)

purpose: This is an abstract class that provides virtual methods via signatures, that are to be implemented by the concrete gasPump factory classes. Since we use the abstract factory design pattern, the client-aspects(gasPump applications) could just import the AbstractFactory.AbstractFactory package, we do not need to classify the specified gasPump factory!

5. Class GasPump1_Factory

purpose: factory class for gas pump 1. Create all the objects that the GasPump application needs. For each actions needed in the gasPump-1, set the OutputProcessor's action strategies. Since we use strategy design patterns, the specified algorithm can be independently from clients that use it.

6. Class GasPump2_Factory

purpose: factory class for gas pump 2. Create all the objects that the GasPump application needs. For each actions needed in the gasPump-2, set the OutputProcessor's action strategies. Since we use strategy design patterns, the specified algorithm can be independently from clients that use it.

7. Class Data(Abstract)

Purpose: The abstract class for all the Data stored that are used by the gasPump application

8. Class DataForGasPump1

Purpose: store all the concrete data used by gasType-1

9. Class DataForGasPump2

Purpose: store all the concrete data used by gasType-2;

10. Class InputProcessor (Abstract)

purpose: This is an abstract class that provides virtual methods, that are to be implemented by the Input Processors of the two gasPumps.

11. Class IPGasPump1

purpose: The input processor for GasPump1, responsible for build the cause and effect relationship between actions and events in the GasPump1.

12. Class IPGasPump2

purpose: The input processor for GasPump2, responsible for build the cause and effect relationship between actions and events in the GasPump2.

13. Class OutputProcessor

purpose: This class represents the Output Processor of the MDA and is the client of the various action strategies. Each action having more than one strategy has an abstract class which is associated to the OP, while actions having only one strategy are directly associated with the OP.

15. Class StateMachine

purpose: This class represents the common functionality or platform independent logic of all of its clients (GasPump-1 and 2). Manages the states by tracking a pointer to the current state class of the EFSM and forwarding calls onto the next state class.

16. Class State (Abstract)

purpose: This is the abstract superclass for all the concrete state classes to define the initial operation for each events(print error), so that it will display an error message if the concrete state class calls the event function, that not override.

17. Class InitState

purpose: This is the Start State, any execution path leaving this state are implemented here and only executed when the state machine is in this state.

18. Class S0

purpose: This is the S0 state after start , any execution path leaving this state are implemented here and only executed when the state machine is in this state.

19. Class S1

purpose: This is the S1 state, any execution path leaving this state are implemented here and only executed when the state machine is in this state.

20. Class S2

purpose: This is the S2 state, any execution path leaving this state are implemented here and only executed when the state machine is in this state.

21. Class S3

purpose: This is the S3 state, any execution path leaving this state are implemented here and only executed when the state machine is in this state.

22. Class S4

purpose: This is the S4 state, any execution path leaving this state are implemented here and only executed when the state machine is in this state.

23. Class S5

purpose: This is the S5 state, any execution path leaving this state are implemented here and only executed when the state machine is in this state.

24. Class S6

purpose: This is the S6 state, any execution path leaving this state are implemented here and only executed when the state machine is in this state.

25. Class AbstractCancelMsg (Abstract)

purpose: encapsulate a group of individual factories that have a common theme without specifying their concrete classes. Use strategy design patterns, so that the specified algorithm can be independently from clients that use it.

26. Class CancelMsg1

purpose: one of the concrete implementations(actions) of cancelMsg function, called by gasPump-1. The methods in this subclass are override to satisfy the specified usage for gasPump-1.

27. Class CancelMsg2

purpose: one of the concrete implementations(actions) of cancelMsg function, called by gasPump-2. The methods in this subclass are override to satisfy the specified usage for gasPump-2.

all other abstract and concrete action classes are similar to class #25, #26 and #27.

……