PROJECT DESIGN and REPORT

DESIGN and IMPLEMENTATION

The goal of the second part of the project is to design two *GasPump* components using the Model-Driven Architecture (MDA) and then implement these *GasPump* components based on this design using the OO programming language. This OO-oriented design should be based on the MDA-EFSM for both *GasPump* components that was identified in the first part of the project. You may use your own MDA-EFSM (assuming that it was correct) or you can use the posted sample MDA-EFSM. In your design, you **MUST** use the following OO design patterns:

- state pattern
- strategy pattern
- abstract factory pattern

In the design, you need to provide the class diagram, in which the coupling between components should be minimized and the cohesion of components should be maximized (components with high cohesion and low coupling between components). In addition, two sequence diagrams should be provided as described on the next page (Chapter 4 of the report).

After the design is completed, you need to implement the *GasPump* components based on your design using the OO programming language. In addition, the driver for the project to execute and test the correctness of the design and its implementation for the *GasPump* components must be implemented.

Outline of the Report & Deliverables

1. MDA-EFSM model for the *GasPump* components
   i. A list of meta events for the MDA-EFSM
   ii. A list of meta actions for the MDA-EFSM with their descriptions
   iii. A state diagram of the MDA-EFSM
   iv. Pseudo-code of all operations of Input Processors of *GasPump-1* and *GasPump-2*

2. Class diagram(s) of the MDA of the *GasPump* components. In your design, you **MUST** use the following OO design patterns:
   i. State pattern
   ii. Strategy pattern
   iii. Abstract factory pattern

3. For each class in the class diagram(s) you should:
   a. Describe the purpose of the class, i.e., responsibilities.
   b. Describe the responsibility of each operation supported by each class.

4. Dynamics. Provide two sequence diagrams for two Scenarios:
   a. Scenario-I should show how one gallon of Regular gas is disposed in *GasPump-1*, i.e., the following sequence of operations is issued: *Activate(3.1, 4.3), Start(), PayCredit(), Approved(), Regular(), StartPump(), PumpGallon(), StopPump()*

   b. Scenario-II should show how one liter of Premium gas is disposed in *GasPump-2*, i.e., the following sequence of operations is issued: *Activate(3, 4, 5), Start(), PayCash(6), Premium(), StartPump(), PumpLiter(), PumpLiter(), NoReceipt()*

Note

Well documented (commented) source code. **Printed hardcopy of the source code is required and should be a part of the report.** Otherwise, **15 POINTS** will be automatically deducted from the project.

IMPORTANT: The project executable(s) of the *GasPump* components with detailed instructions explaining the execution of the program must be prepared by students and made available for grading.