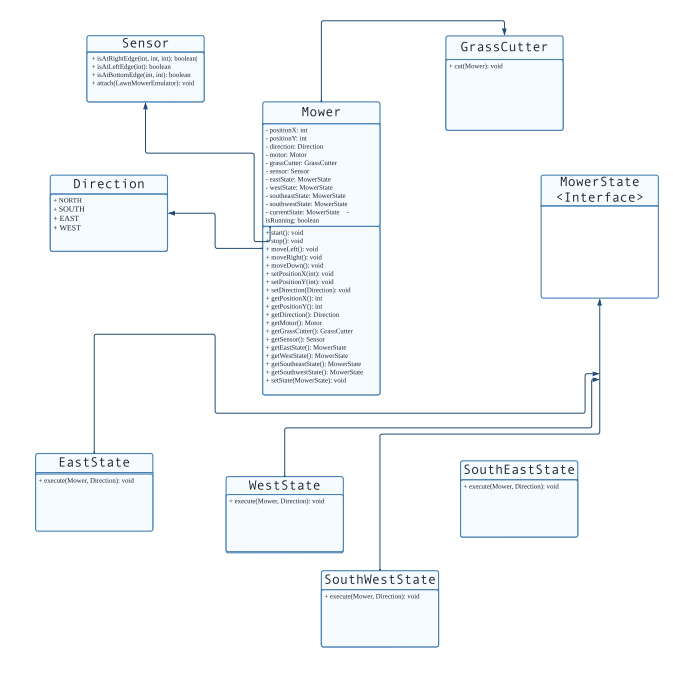
Report on Lawn Mower Emulator Program  
  
Introduction:  
The Lawn Mower Emulator is a Java program designed to simulate the movement of a lawn mower on a lawn using a graphical user interface. The program is designed using the Model-View-Controller (MVC) architecture, and implements both the state and observer patters as explained below.  
  
Classes:  
The Lawn Mower Emulator program has several classes that work together to simulate the movement of the lawn mower. The following are the classes in the program:  
  
Direction: This class is an enumeration that defines the different directions that the mower can move in, including east, west, southeast, and southwest.  
  
GrassCutter: This class is responsible for cutting the grass as the mower moves over it. It has a cut() method that is called by the Mower class.  
  
Lawn: This class defines the dimensions of the lawn as constants.  
  
LawnMowerEmulator: This class is the main controller class that extends the JFrame class and implements the MowerObserver interface. It creates the GUI and handles the starting and stopping of the mower.  
  
LawnPanel: This class is responsible for drawing the lawn and the mower on the GUI. It extends the JPanel class and has a paintComponent() method that draws the lawn and the mower.  
  
Motor: This class is responsible for starting and stopping the motor of the mower.  
  
Mower: This class represents the mower model and has instance variables for the position, direction, motor, grass cutter, sensor, and four MowerState objects. It also has various methods for moving the mower in different directions.  
  
MowerObserver: This interface has a single method onMove() which is called by the Sensor class when the mower moves.  
  
MowerState: This interface has a single method execute() which is implemented by four classes (EastState, WestState, SoutheastState, and SouthwestState) to execute the different states of the mower.  
  
Sensor: This class is responsible for detecting when the mower is at the edge of the lawn and notifying the observer object. It has instance variables for the observer and a getCurrentMower() method to get the current mower object.  
  
Design:  
The Lawn Mower Emulator program is designed using the Model-View-Controller (MVC) architecture. The Mower class represents the model and is responsible for the movement and state of the mower. The LawnPanel class represents the view and is responsible for drawing the lawn and the mower on the GUI. The LawnMowerEmulator class represents the controller and is responsible for creating the GUI and handling the starting and stopping of the mower.  
  
The Sensor class works as an observer to detect when the mower is at the edge of the lawn and notify the observer object (LawnMowerEmulator) of the mower's movements. The LawnMowerEmulator class, as the observer, implements the onMove() method to update the GUI and display the new position of the mower.  
  
The MowerState interface and its four implementing classes (EastState, WestState, SoutheastState, and SouthwestState) implement the state pattern. They represent the different states that the mower can be in and encapsulate the behavior of the mower when it is in a particular state.

State Diagram.

  
Conclusion:  
  
The Lawn Mower Emulator program is a well-designed Java program that simulates the movement of a lawn mower on a lawn using a graphical user interface. The program uses the Model-View-Controller (MVC) architecture and implements the state and observer design patterns to encapsulate the mower's behavior and notify the observer when the mower reaches the edge of the lawn. The program's classes work together seamlessly to create an engaging and interactive GUI that allows users to control the movement of the mower and see the effects of its movements on the lawn.