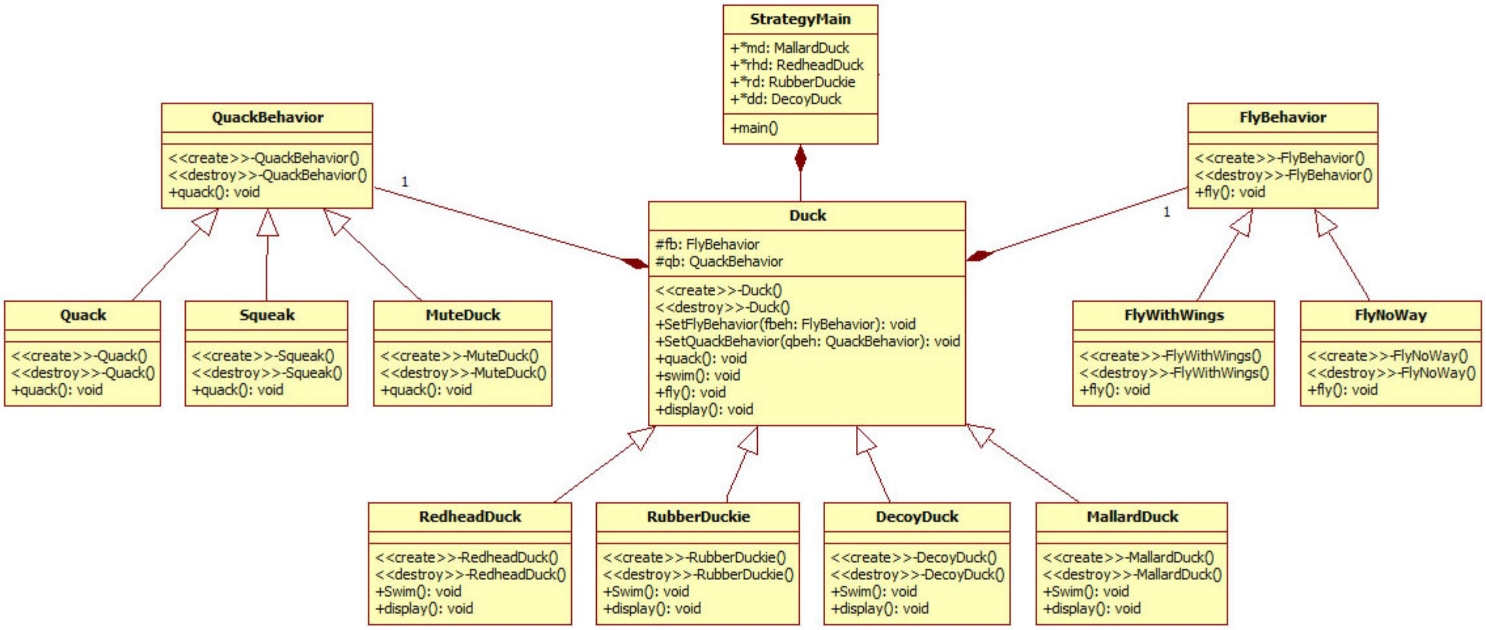
Strategy Design Pattern

GoF Statement: **Defines a family of algorithms, encapsulates each one, and makes them interchangeable.**

Category: **Behavioral**

UML Diagram:



Description of the Demonstration:

A class called Duck is created along with two "families" of algorithms: QuackBehavior and FlyBehavior. The QuackBehavior parent class has three subclasses: Quack, Squeak, and MuteDuck. The FlyBehavior parent class has two subclasses: FlyWithWings, and FlyNoWay. When an instance of one of the subclasses of Duck (see UML diagram) is created it is given an appropriate instance of one subclass of QuackBehavior and one subclass of FlyBehavior. Because all subclasses implement the common interfaces an instance of a subclass of Duck can be composed with its appropriate quack and fly behavior at run time. These behaviors can also be easily changed while the application is running.

When the demonstration program is run an instance of each of the subclasses of Duck is created then the quack() and fly() functions are called on each of the "ducks". Note in the UML diagram that a swim() method has also been implemented in Duck. This behavior could also be extracted into a third family of algorithms and swimming handled the same way as Quack and Fly.