

Design Pattern Definitions from the GoF Book

The Strategy Pattern

Defines a family of algorithms, encapsulates each one, and makes them interchangeable.

Creational Patterns

- The Factory Method Pattern
- The Abstract Factory Pattern
- The Singleton Pattern
- The Builder Pattern
- The Prototype Pattern

Structural Patterns

- The Decorator Pattern
- The Adapter Pattern
- The Facade Pattern
- The Composite Pattern
- The Proxy Pattern
- The Bridge Pattern
- The Flyweight Pattern

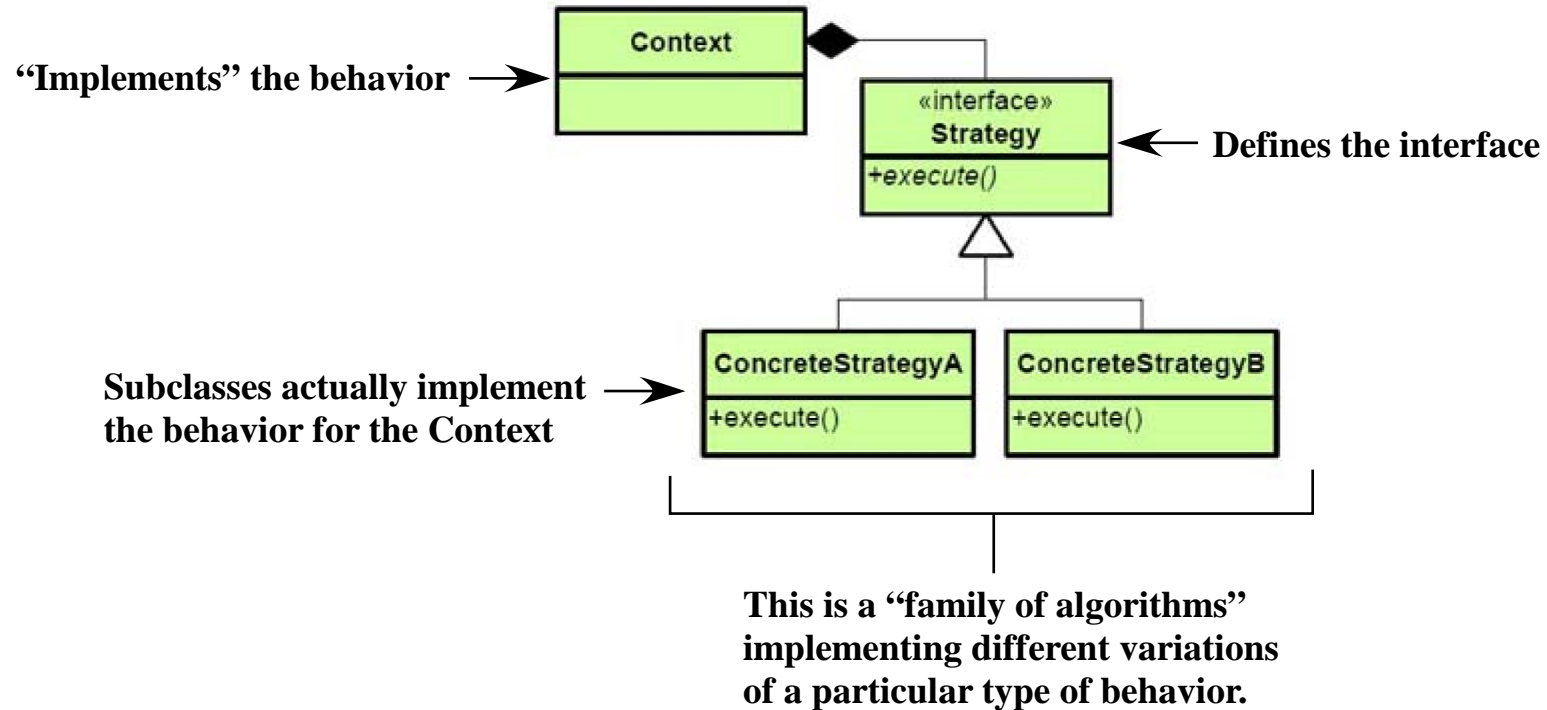
Behavioral Patterns

- The Strategy Pattern
- The Observer Pattern
- The Command Pattern
- The Template Method Pattern
- The Iterator Pattern
- The State Pattern
- The Chain of Responsibility Pattern
- The Interpreter Pattern
- The Mediator Pattern
- The Memento Pattern
- The Visitor Pattern

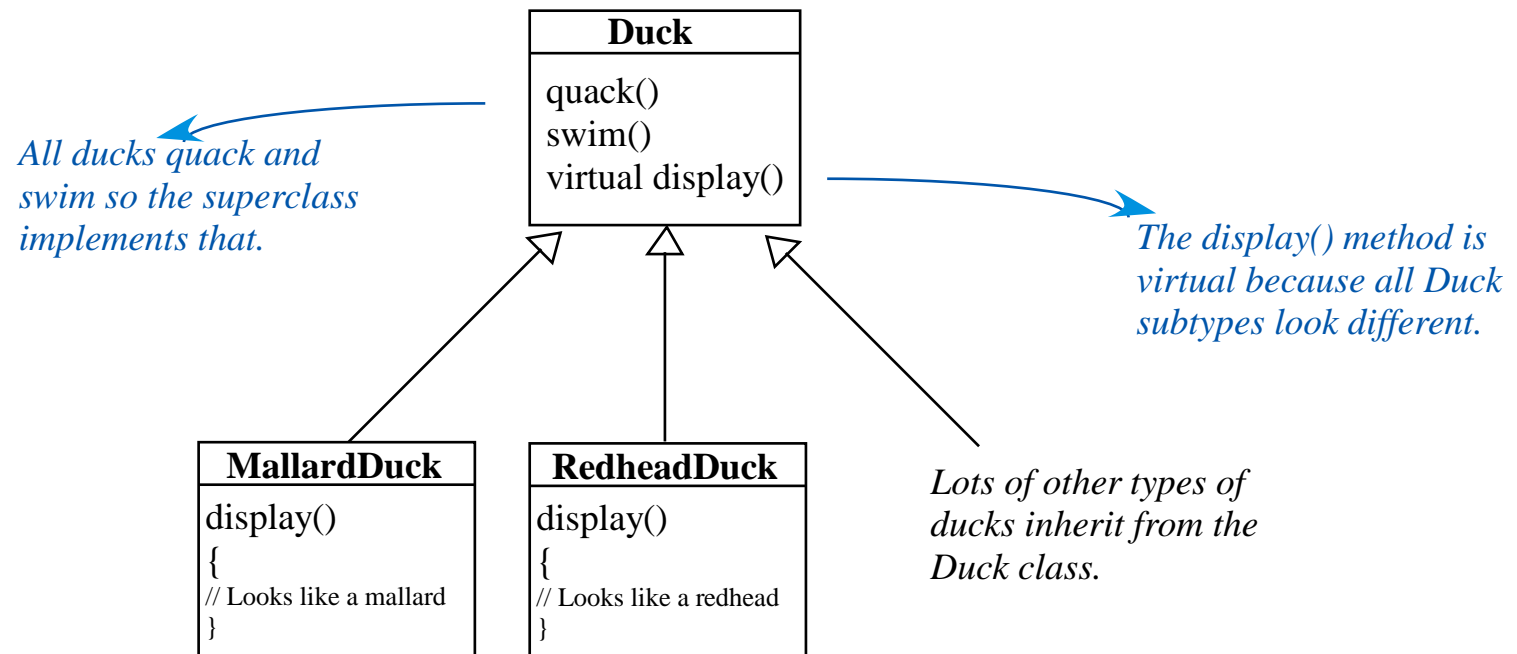
Design Patterns: Strategy

Quick Overview

Defines a family of algorithms, encapsulates each one, and makes them interchangeable.

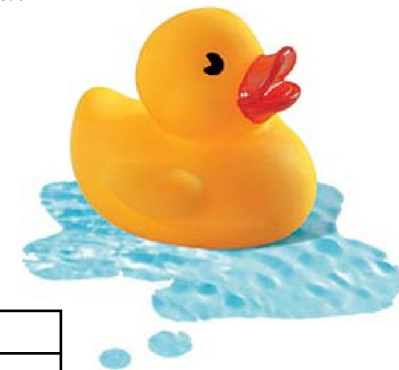
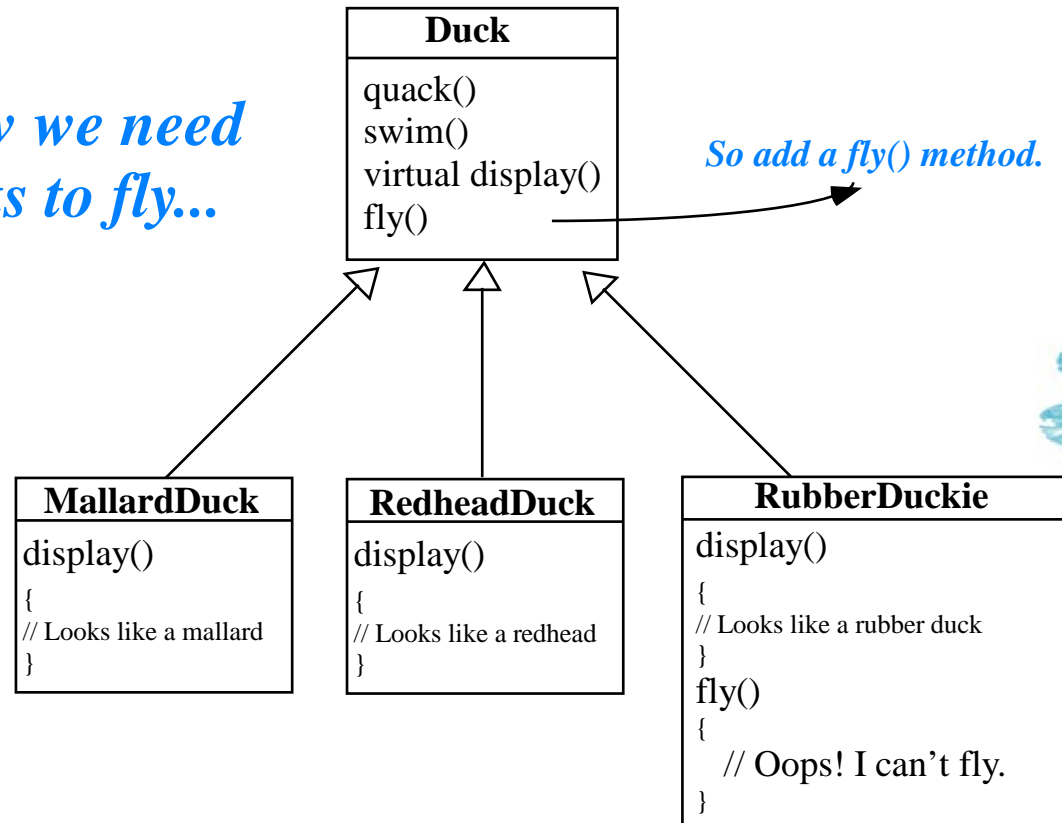


Design Patterns: Strategy



Design Patterns: Strategy

*But, now we need
the ducks to fly...*



Oops!

So what's a programmer to do?

Design Patterns: Strategy

The one constant in software development.

Change

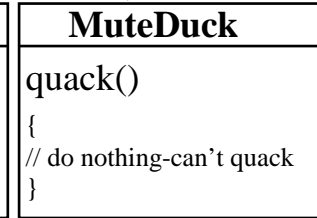
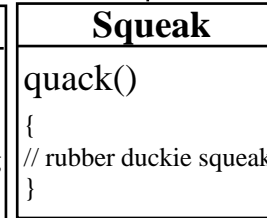
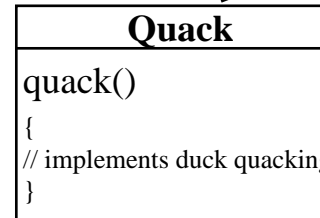
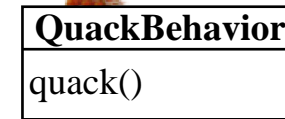
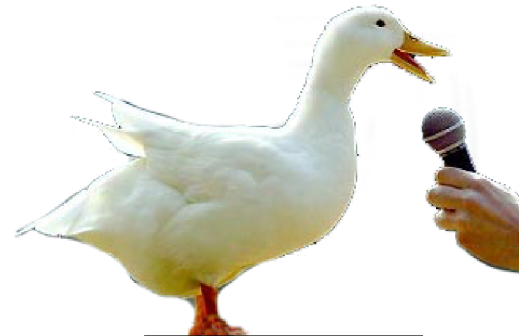
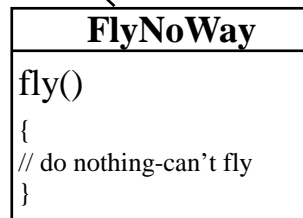
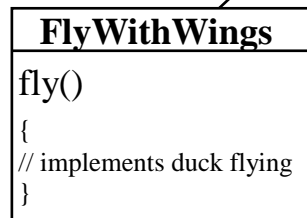
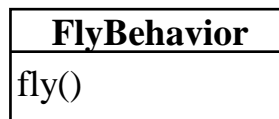
Design Principles

- *Identify the aspects of your application that vary and separate them from what stays the same.*

Remember me?

Design Patterns: Strategy

Encapsulate the part that changes



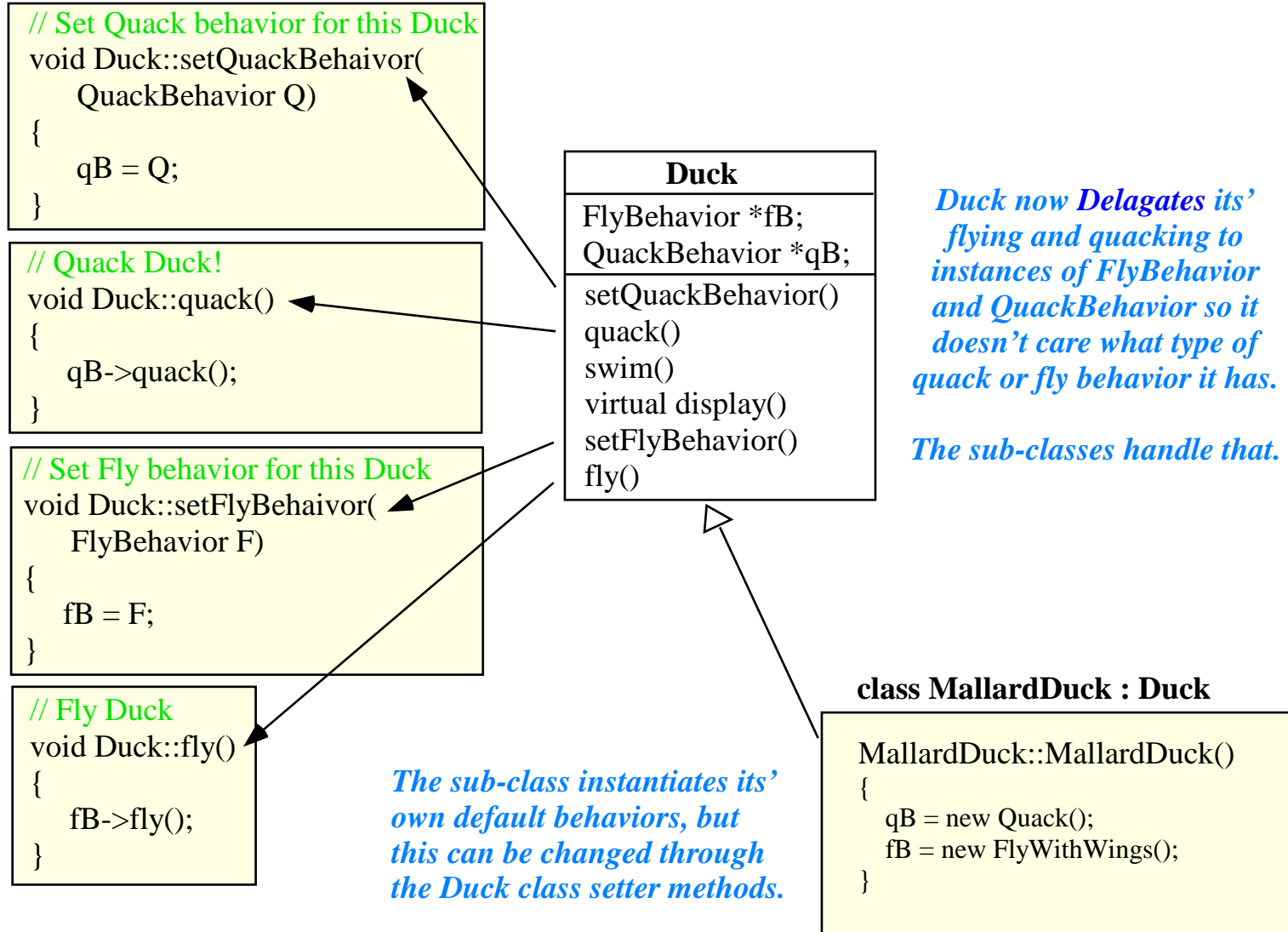
Design Principles

FlyBehavior and QuackBehavior are the interfaces

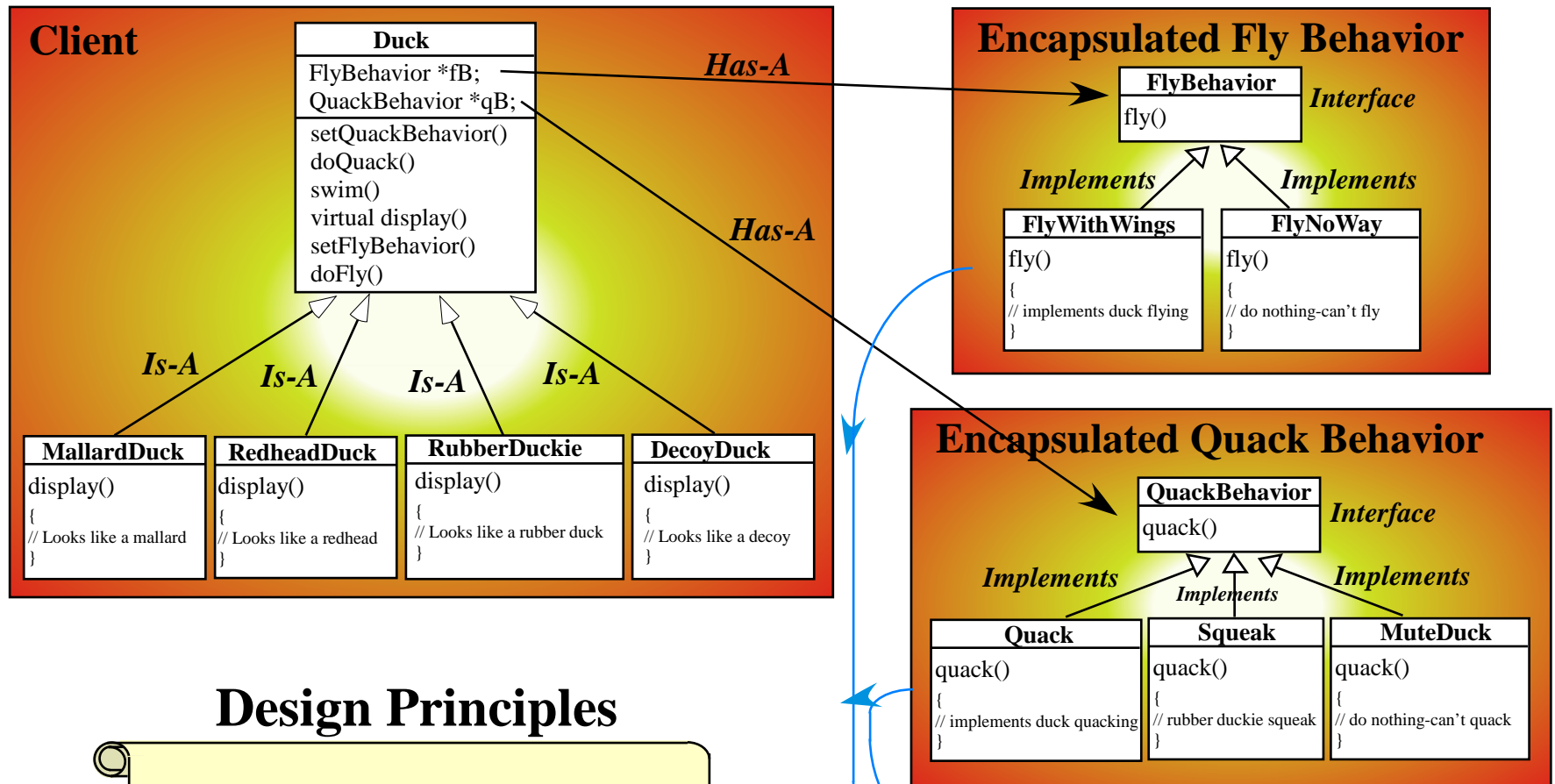
- Program to an interface, not an implementation.

FlyWithWings, FlyNoWay, Quack, Squeak, MuteDuck are the implementations.

Design Patterns: Strategy



Design Patterns: Strategy



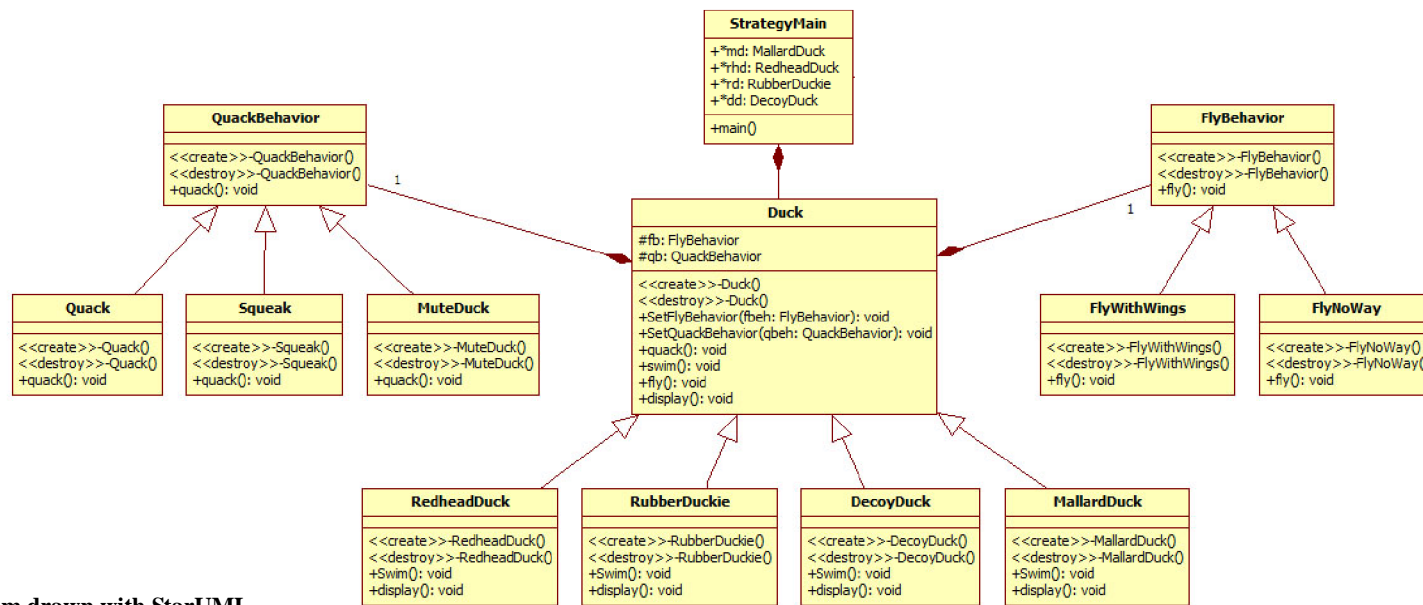
Design Principles

- *Favor composition over inheritance.*

Think of each set of behaviors as a family of algorithms each of which is interchangeable with any of the others. Now you can compose the behavior.

Design Patterns: Strategy

Code Sample



StrategyMain

Instantiates instances of:

RedheadDuck

Instantiates instances of Quack and FlyWithWings

RubberDuckie

Instantiates instances of Squeak and FlyNoWay

DecoyDuck

Instantiates instances of MuteDuck and FlyNoWay

MallardDuck

Instantiates instances of Quack and FlyWithWings

Let's look at the code and run the demonstration.