

# The Strategy Pattern

The Strategy pattern provides a way to define a family of algorithms, encapsulate each one as an object, and make them interchangeable. It makes good use of a key design principle:

*Design Principle:*

***Favor composition over inheritance***

For instance, a class that performs validation on incoming data may use a strategy pattern to select a validation algorithm based on the type of data, the source of the data, user choice, or other discriminating factors. These factors are not known for each case until run-time, and may require radically different validation to be performed. The validation strategies, encapsulated separately from the validating object, may be used by other validating objects in different areas of the system (or even different systems) without code duplication.

**Figure 1 - Using the strategy pattern for duck example**

