

The Strategy Pattern

Design Patterns



Overview

Strategy Pattern

- **Intent**
- **Also Known As**
- **Motivating Example**
- **Applicability**
- **Implementation Example**
- **Consequences**
- **Known Uses**
- **Related Patterns**

Motivating Example

Strategy
Pattern

- **An order shipping calculator**
- **The calculator must determine shipping costs**
 - For FedEx
 - For UPS
 - For the US Postal Service (USPS)

Intent

Strategy
Pattern

- **Encapsulate a family of related algorithms**
- **Let the algorithm vary and evolve separate from the class using it**
- **Allow a class to maintain a single purpose**
- **Separate the calculation from the delivery of its results**

Applicability

Strategy
Pattern

- **Switch statements are a red flag**
- **Adding a new calculation will cause a class file to be modified**

Implementation Example

Strategy
Pattern

- **Create classes for each calculation (Strategies)**
- **Use a common interface for each Strategy**

Consequence

Strategy Pattern

- **Strategies may not use members of the containing class**
- **Tests may now be written for individual concrete strategies**
- **Strategies may be mocked when testing the Context class**
- **Adding a new Strategy does not modify the Context class**

Known Uses

Strategy
Pattern

- **Delegates in C# 3.5 and greater**
- **Service classes passed to constructors for**
 - ASP.NET MVC Controllers

Variations

- **Property Injection**

Related Patterns

Strategy
Pattern

References

Strategy
Pattern