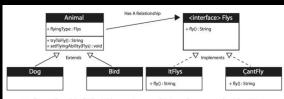
Strategy 算法具针版



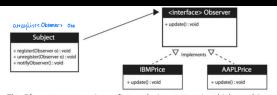
Define a family of algorithms, encapsulate each one, and make them interchangeable. **The Strategy pattern** lets the algorithm vary independently from clients that use it.

encopsulated interchangeable algorithms

When to Use the Strategy Pattern

- Often reduces long lists of conditonals
- Avoids duplicate code
- Keeps class changes from forcing other class changes
- Can hide complicated / secret code from the user
- Negative: Increased Number of Objects / Classes

Observer



The **Observer pattern** is a software design pattern in which an object, called the subject, maintains a list of its dependents, called observers, and notifies them automatically of any state changes, usually by calling one of their methods.

A use thread to update repeatedly

When to Use the Observer Pattern

- When you need many other objects to receive an update when another object changes
 - Stock market with thousands of stocks needs to send updates to objects representing individual stocks
 - The Subject (publisher) sends many stocks to the Observers
 - The Observers (subscribers) takes the ones they want and use them

When to Use the Observer Pattern

- Loose coupling is a benefit
- The Subject (publisher) doesn't need to know anything about the Observers (subscribers)
- Negatives: The Subject (publisher) may send updates that don't matter to the Observer (subscriber)