

100 design patterns

Reusable solution to common design problem



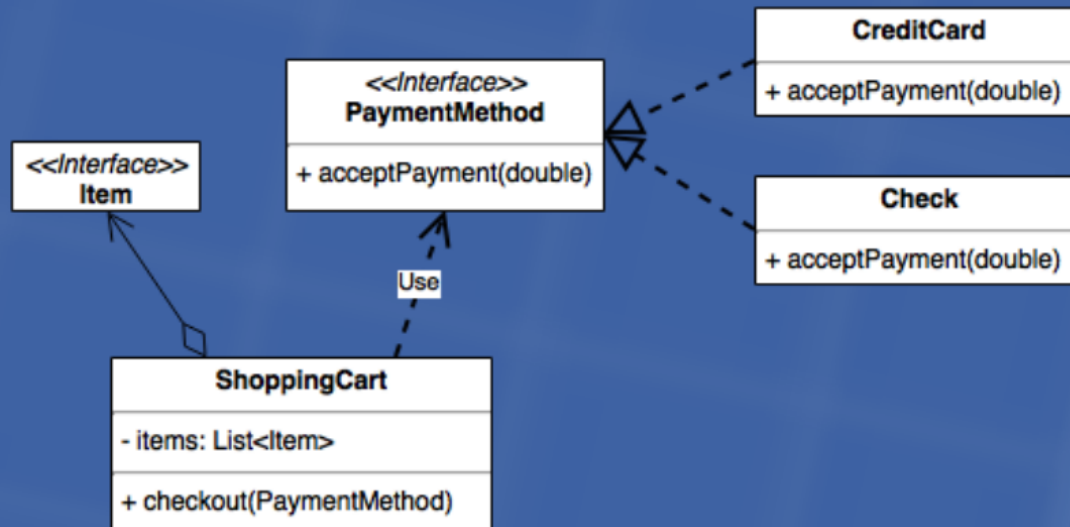
Tried & tested templates that incorporate good design *principles*

Show relationships/interactions between classes & objects

Allow programmers to *share* knowledge about design



Remember this example?



PaymentMethod here is a

Strategy

Defines a *family* of algorithms

Each algorithm captured by separate class

Makes algorithms interchangeable

Allows algorithm to vary *independently* of client using algorithm

Supports *open-closed* principle

A.k.a. *Policy* pattern

Design skeleton



Strategy

Defines a *family* of algorithms

Each algorithm captured by separate class

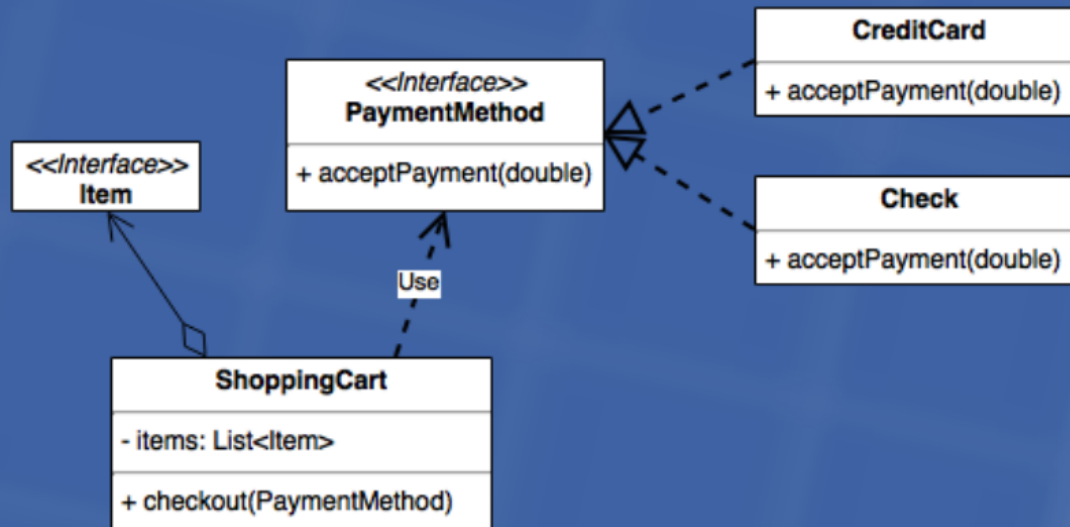
Makes algorithms interchangeable

Allows algorithm to vary *independently*
of client using algorithm

Supports *open-closed* principle

A.k.a. *Policy* pattern

Remember this example?



PaymentMethod here is a

Strategy

Defines a *family* of algorithms

Each algorithm captured by separate class

Makes algorithms interchangeable

Allows algorithm to vary *independently* of client using algorithm

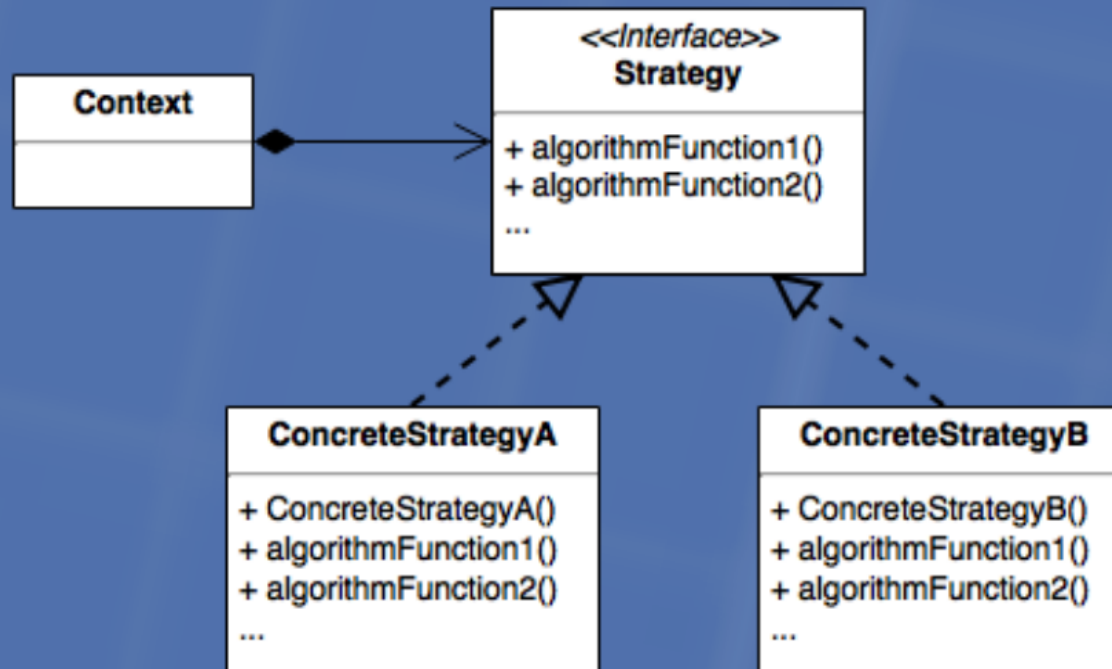
Supports *open-closed* principle

A.k.a. *Policy* pattern

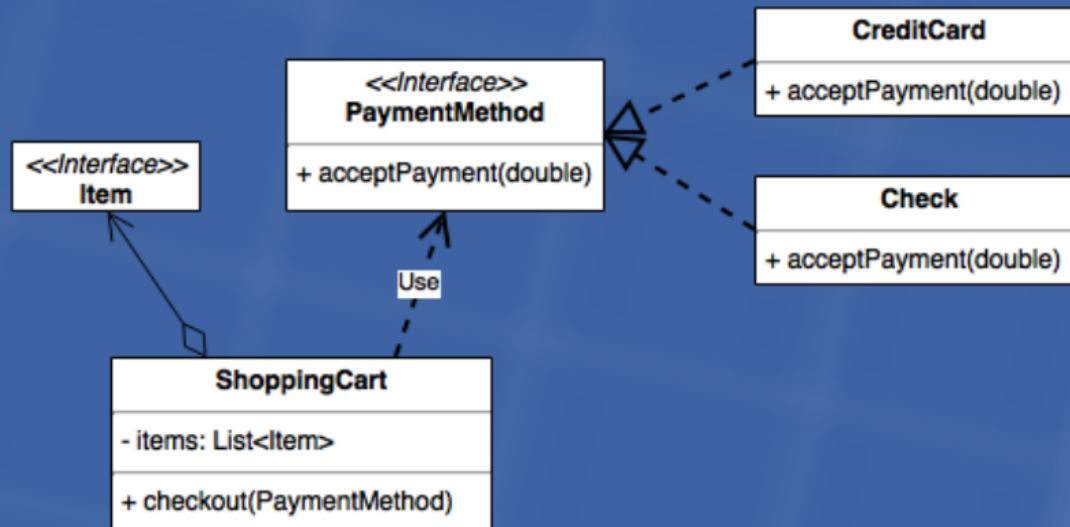
Design skeleton



Design skeleton



Remember this example?



PaymentMethod here is a

Strategy

Defines a *family* of algorithms

Each algorithm captured by separate class

Makes algorithms interchangeable

Allows algorithm to vary *independently* of client using algorithm

Supports *open-closed* principle

A.k.a. *Policy* pattern

Design skeleton

