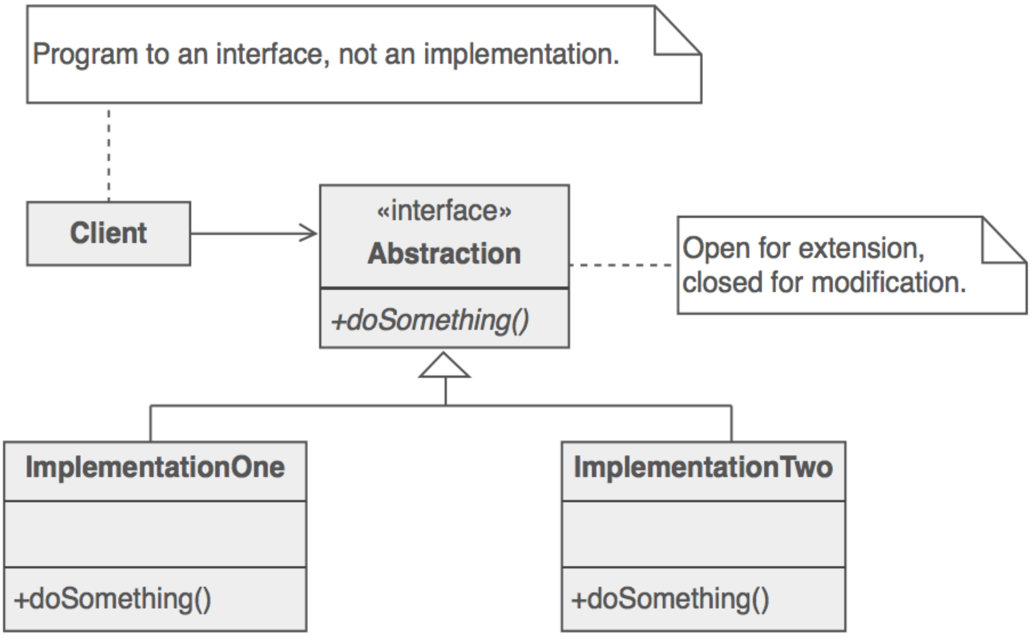
What it is:

- Encapsulates an algo. inside a subclass
- Define a family of algorithms
    - encapsulate each one
    - make them interchangeable

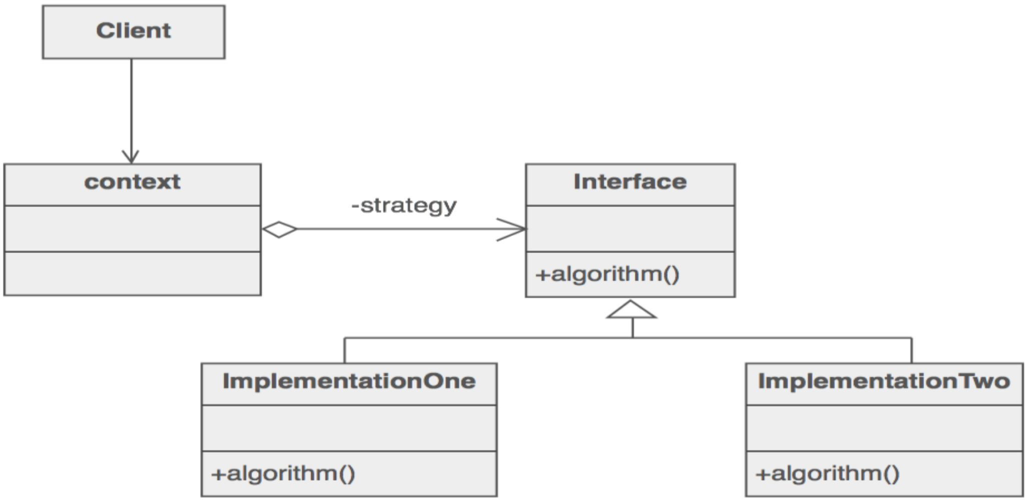
when to use it:

- have one behavior in a class that is similar to other behaviors in list
- need to use one of several behaviors dynamically
-

Problem:



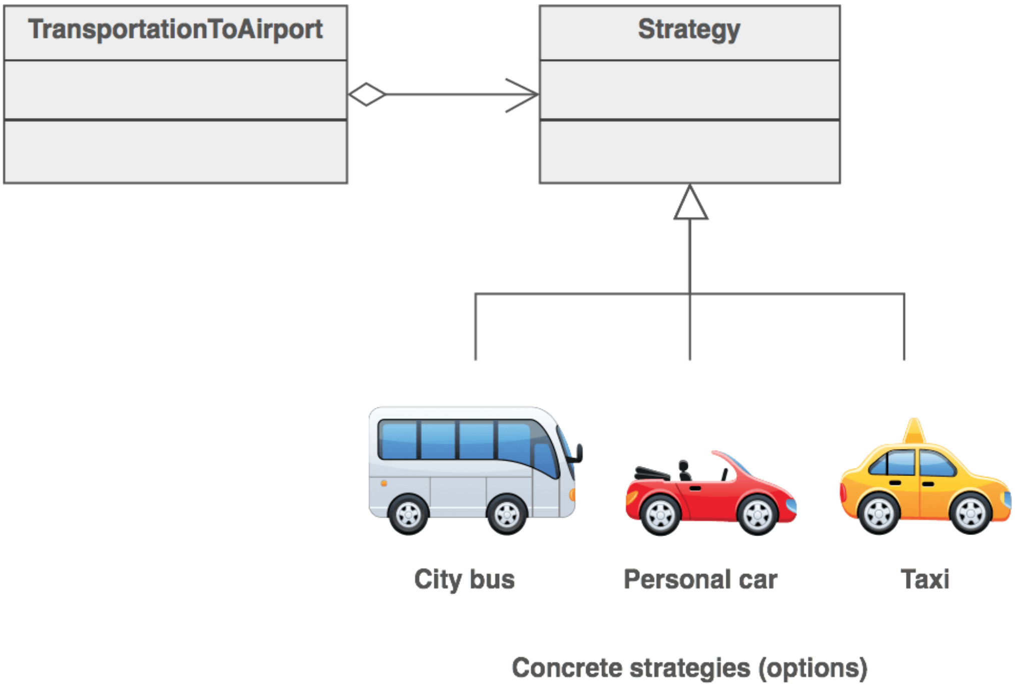
Structure:



2 types:

- Interface can be an abstract base class: Dynamic polymorphism
- Interface is interface: static polymorphism

Example:



Coding Check List:

1. Identify an algo (i.e a behavior) that the client would prefer to access through a "flex point"
2. Specify the signature for that algorithm in an interface.
3. Implementation in derive class
4. Clients only couple/know themselves to an interface

Rules Of Thumb: