

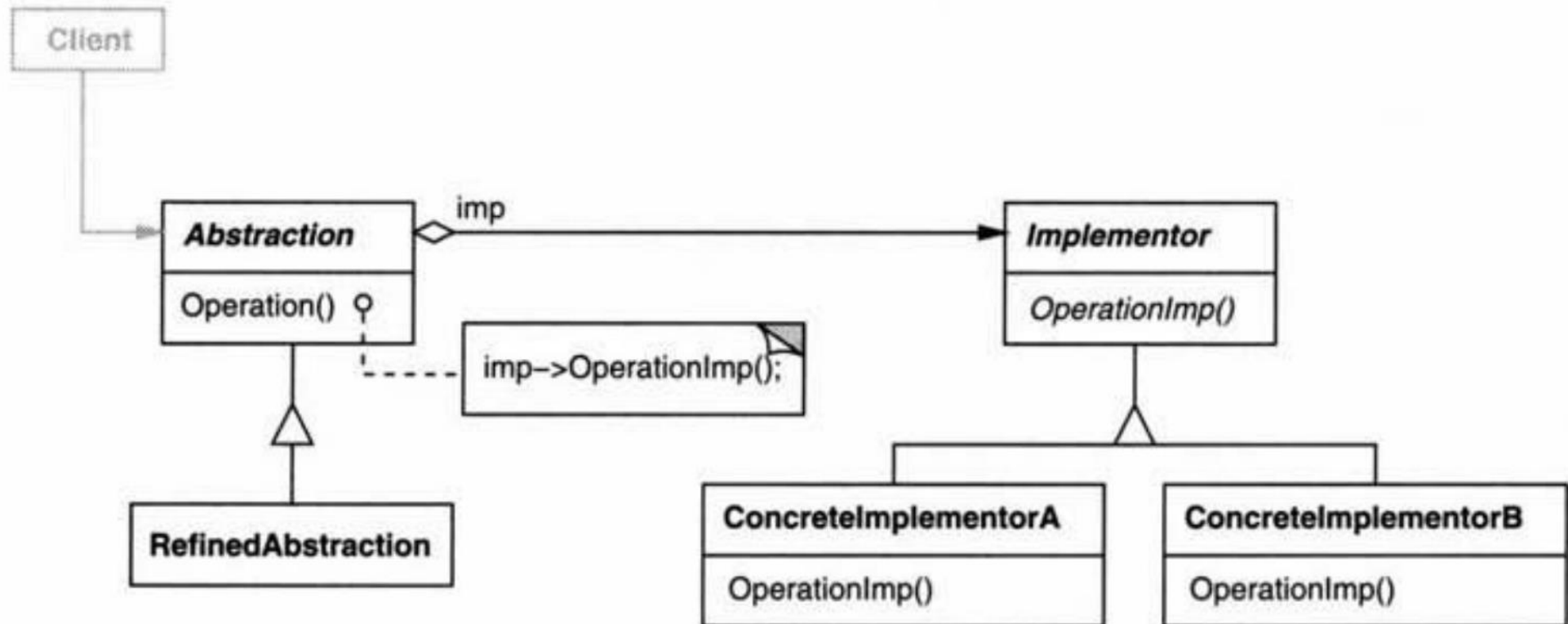
# Bridge Pattern

Bridge Pattern bridges abstraction and its implementation, letting them vary independently.

# Use cases

- separate API from the implementation - hiding implementation to clients,
- decoupling interface and implementation,
- open for changes,

# Bridge pattern structure (Design Patterns: Elements of reusable software)



# Elements

Abstraction

Implementator

Concrete Implementator

# Elements

Refined Abstraction: implements abstraction. Add specific implementation one level higher.

- Can implements low level functions from abstractions and run inside some functions specified by the implementator.
- Can implements high level functions from abstraction and add high level functionality which is different than in implementator.