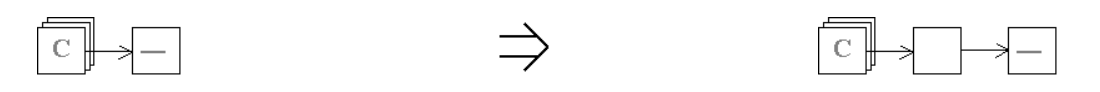
**Introduce Indirection**

*a feature heavily depended upon by many clients needs to be evolved*

*introduce an indirection level that decouples client code from a direct depencency upon the evolving feature*



**Context:**

Both [Hide Subsystem Complexity With Facade](https://web.archive.org/web/20090914101033/http:/www.refactoring.be/thumbnails/rtn-hscwf.html) and [Replace Singletons With Singleton](https://web.archive.org/web/20090914101033/http:/www.refactoring.be/thumbnails/rtn-rsws.html) are examples of [Introduce Indirection](https://web.archive.org/web/20090914101033/http:/www.refactoring.be/thumbnails/rtn-ii.html) where indirection level is used to encapsulate a design property that is difficult to control.

An indirection level that has outlived its purpose can be removed with [Inline Indirection](https://web.archive.org/web/20090914101033/http:/www.refactoring.be/thumbnails/rtn-ri.html). However, keeping the indirection level around manytimes proves helpfull during later refactorings.

[Introduce Indirection](https://web.archive.org/web/20090914101033/http:/www.refactoring.be/thumbnails/rtn-ii.html) is a key refactoring in [Break Module Depencencies With Adapter](https://web.archive.org/web/20090914101033/http:/www.refactoring.be/thumbnails/ec-adapter.html)