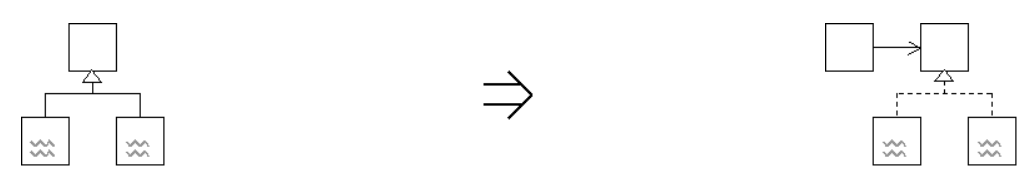
**Replace Template With Strategy**

*a varying concept is locked within a class hierarchy*



*capture the concept with an interface and extract the variation from the hierarchy*

**Context:**

[Eliminate Duplication By Inheritance](https://web.archive.org/web/20090914101225/http:/www.refactoring.be/thumbnails/rtn-edbi.html) essentialy leaves the code with a Template based structure that allows variation by subclassing thus coupling the varying concept with the class hierarchy.

[Replace Template With Strategy](https://web.archive.org/web/20090914101225/http:/www.refactoring.be/thumbnails/rtn-rtswss.html) decouples the variation from the class hierarchy by capturing the varying concept with an interface.

[Replace Template With Strategy](https://web.archive.org/web/20090914101225/http:/www.refactoring.be/thumbnails/rtn-rtswss.html) is a key refactoring in [Evolving to the Strategy Pattern](https://web.archive.org/web/20090914101225/http:/www.refactoring.be/thumbnails/ec-strategy.html)

See also : Access Subclass Features By Interface and Break Inheritance