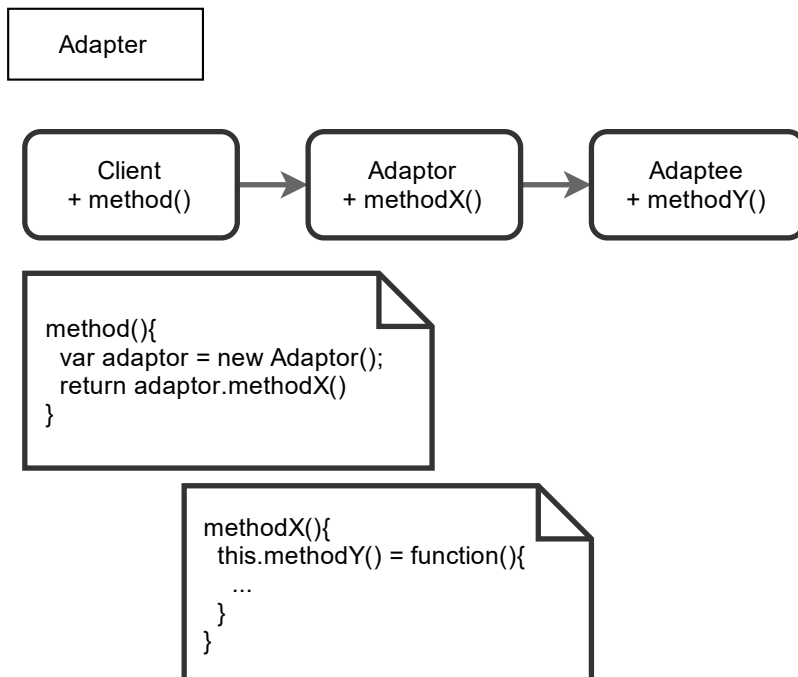


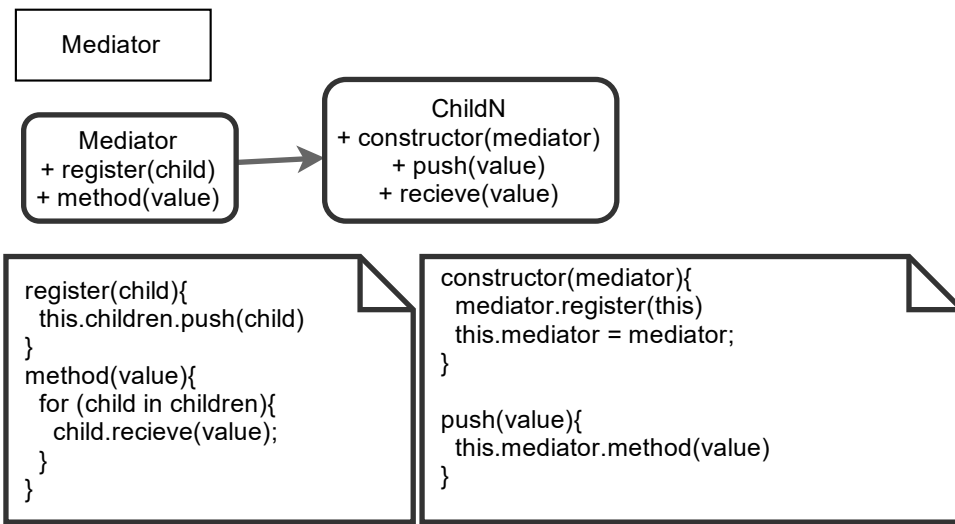
<http://www.se.rit.edu/~ehj2229/webengwork/designpatterns/facade.html>

- + hides complicated / complex code
- + simplifies old service solutions
- maintenance of Facade is still reliant on knowledge of services
- old implementations may rely on outdated / specific systems, increasing the requirements for the base application



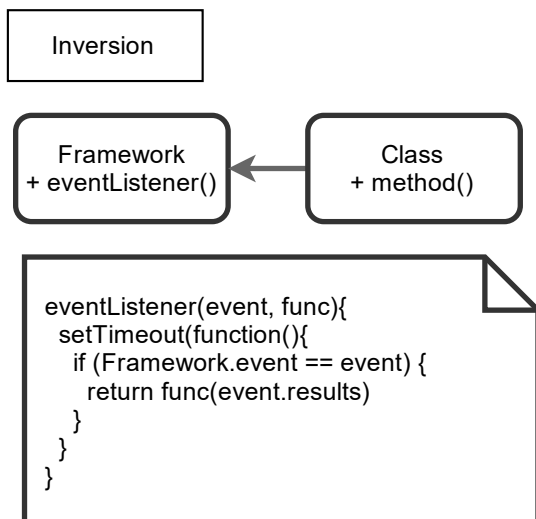
<http://www.se.rit.edu/~ehj2229/webengwork/designpatterns/adapter.html>

- + ability to work with many interfaces and services
- security concerns related to untested / unknown code



<http://www.se.rit.edu/~ehj2229/webengwork/designpatterns/mediator.html>

- + lower coupling between children
- + one object to handle logic between components
- slow down on heavy performance tasks



<http://www.se.rit.edu/~ehj2229/webengwork/designpatterns/inversion.html>

- + Utilize Framework for timing of class methods
- reliance of Framework implementation (and pitfalls of framework)