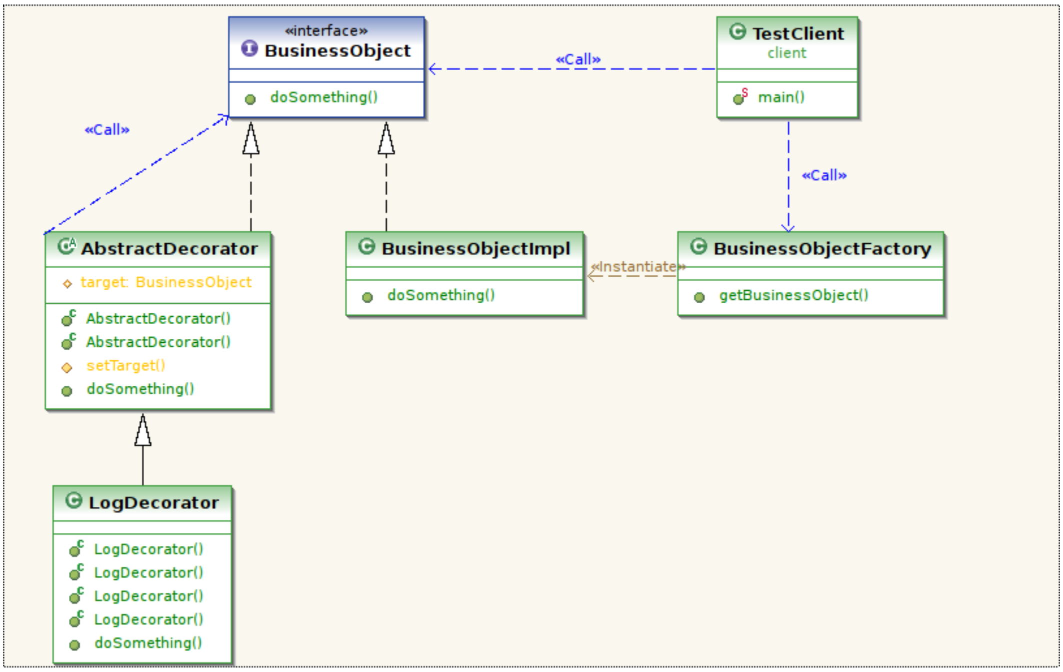
Lab Description

Decorator

Overview	
Chain the things using the Decorator	
Objectives	Learn how to implement an Decorators
Deployments / Builds	---
What's new	Decorator, maybe
Aproximate time	20 min.

Tasklist

Well, we had covered Proxies but we didn't put them into a sequence. Time to change this now, we want to make use of the Decorator pattern to come closer to something people call: Aspect-Oriented Programming (huh)



✗ **Implement interface**

So the magic of this lies within the clever combination of a „joint" interface. Check this interface first and make the BusinessObjectImpl implement it.

✗ **Implement the magic super-class: AbstractDecorator**

As the „endpoint" and this abstract class share the same interface, we should simply implement the method provided by the interface to invoke a target which will be set at runtime during instantiation.

Weaving part one: LogDecorator

The first in line: Logging! So extend the magic superclass an we are done.

Lab Description

- ✗ **Weaving part two: SecurityDecorator**

  Well, second in line: Security!! As before …

- ✗ **Make it happen: BusinessObjectFactory**

  Our access point for the clients. Provide a method to get access to our BusinessObject and some switch to decide whether we will use our chain.

  If so, we have to instantiate the last in line and due to chained constructor calls we will set all targets …

- ✗ **Test it**

  By using the client provided