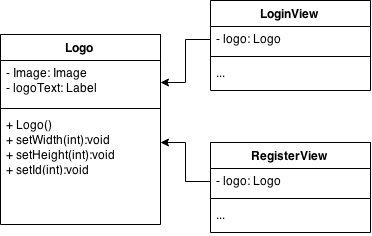
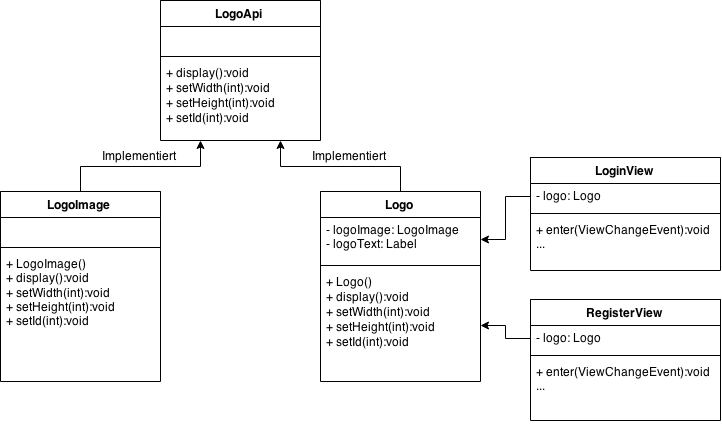
# Design Pattern im Projekt SocialFunnel

Das Projekt verwendet nun das [Proxy-Pattern](http://www.tutorialspoint.com/design_pattern/proxy_pattern.htm), um Ressourcen effizienter zu laden. Das Ziel des Proxy-Pattern ist es, Ressourcen nur wenn sie benötigt werden auch tatsächlich zu laden und ein mehrfaches Laden zu verhindern. Dieses Vorgehen soll den Speicherverbrauch reduzieren und Ladezeiten durch Wiederverwendung bereits geladener Ressourcen kürzen.

**UML - vor Implementierung**

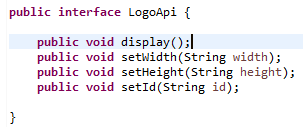


**UML - nach Implementierung**



**Code**

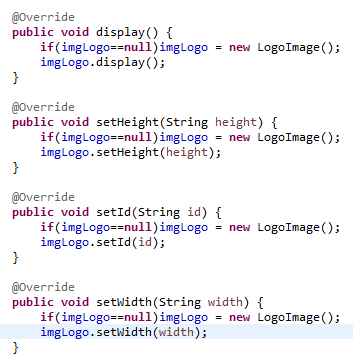
**LogoApi.java:**



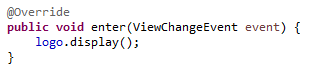
**LogoImage.java:**



**Logo.java:**



**Aufruf in Login- und RegisterView.java:**



**Testen des implementierten Patterns**

Unit Tests machen Für dieses Pattern keinen Sinn, da es ja beabsichtigt ist das Verhalten der Klasse vollständig zu Kapseln und den Ladevorgang für Ressourcen zu optimieren.

Dieses Verhalten lässt sich lediglich durch zusätzliche Ausgaben bei Laufzeit aufzeigen.

Das Szenario in diesem Fall ist das Starten der Anwendung und mehrfaches hin und herspringen zwischen dem Login- und Registerview.

Dieses Vorgehen produziert folgende Ausgabe:

