

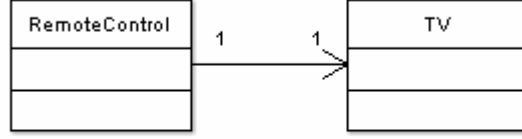
Dependency Inversion Principle (DIP)

Bağımlılıkların Tersine Çevrilmesi Prensibi

KurumsalJava.com

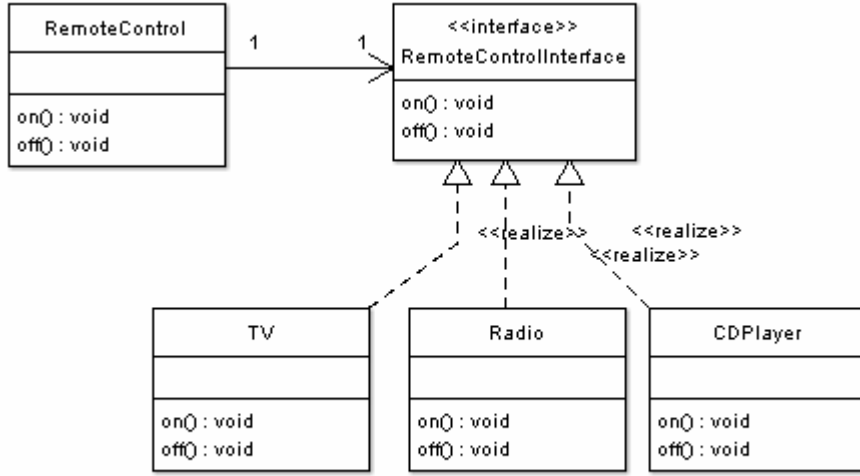
Özcan Acar
Bilgisayar Mühendisi
<http://www.ozcanacar.com>

Bu prensibe göre somut sınıflara olan bağımlılıklar soyut sınıflar ve interface sınıflar kullanılarak ortadan kaldırılmalıdır, çünkü somut sınıflar sık sık değişikliğe uğrarlar ve bu sınıflara bağımlı olan sınıflarında yapısal değişikliğe uğramalarına sebep olurlar.



Resim 1 RemoteControl ve TV sınıfları

Resim 1 de görülen yapı DIP prensibine ters düşmektedir, çünkü RemoteControl sınıfı somut bir sınıf olan TV sınıfına bağımlıdır. TV bünyesinde meydana gelen her değişiklik doğrudan RemoteControl sınıfını etkileyecektir. Ayrıca RemoteControl sınıfını TV sınıfı olmadan başka bir yerde kullanılması mümkün değildir.



Resim 2 Esnek bağ için gerekli sınıf yapısı

Resim 2 de yer alan tasarım DIP ile uyumludur, çünkü RemoteControl sınıfı somut olan TV yerine soyut olan bir interface sınıfına bağımlıdır. Bu tasarım RemoteControl sınıfı ile TV sınıfı arasındaki bağımlılığı yok eder. Ayrıca RemoteControl ve bağımlı olduğu RemoteControlInterface sınıfı beraber başka bir yerde tekrar kullanılabilir.

Bu prensibin uygulanması, somut sınıfların kullanımından doğan değişikliklerin azaltılmasını sağlar.

EOF (End Of Fun)

Özcan Acar