

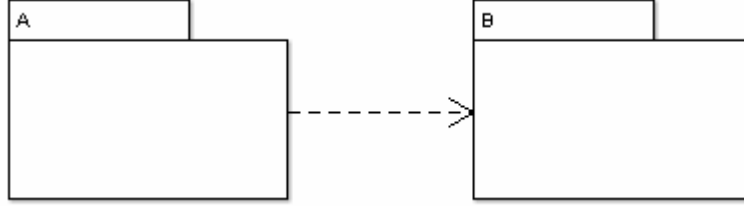
Reuse-Release Equivalence Principle (REP)

Tekrar Kullanım ve Sürüm Eşitliđi

KurumsalJava.com

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

Program modülleri paketler (packages) kullanılarak organize edilir. Paketler arasında sınıfların birbirlerini kullanmalarıyla bağımlılıklar oluşur. Bunun bir örneği resim 1 de yer almaktadır. Eğer paket B bünyesindeki bir sınıf paket A bünyesinde bulunan bir sınıf tarafından kullanılıyorsa, bu paket A'nın paket B'ye bağımlılığı olduğu anlamına gelir. Bu tür bağımlılıkların oluşması doğaldır. Amaç bu bağımlılıkları ortadan kaldırmak değil, kontrol edilebilir hale getirmek olmalıdır. Bu amaçla paket bazında uygulanabilecek tasarım prensipleri oluşturulmuştur. Bunlardan birisi Reuse-Release Equivalence (tekrar kullanım ve sürüm eşitliği) prensibidir.



Resim 1 Paket A (package) paket B'ye bağımlıdır, çünkü paket A içindeki sınıf ya da sınıflar paket B'den bir veya birden fazla sınıfı kullanıyorlar

Bir proje bünyesinde değişik modüllerden oluşan bir yazılım sistemini implemente etmek için çalıştığımızı düşünelim. Her modülden ayrı bir ekip sorumlu olsun. Üzerinde çalıştığımız modülün implementasyonunu yapabilmek için büyük bir ihtimalle diğer modülleri kullanmamız gerekecektir. Örneğin bilgibankası üzerinde işlem yapmamızı sağlayacak bir modülü başka bir ekip implemente etmiş olabilir. O ekip bilgibankası üzerindeki işlemler için gerekli tüm sınıfları *dao* isimli bir paket içine yerleştirmiş olabilir. Bizim bu paket içindeki sınıfları tekrar kullanabilmemiz (reuse) için belirli şartların yerine gelmesi gerekmektedir. Bunlar:

- Bilgibankası ekibi, bilgibankası üzerinde işlem yapmak için kullanılan tüm sınıfları, bu sınıflar birbirleriyle ilişkili olduğu için aynı paket içine koymalıdır. Bu tekrar kullanımı kolaylaştırır.
- Tekrar kullanımı desteklemek için oluşturulan paketin bir versiyon numarasıyla yeni sürümünün oluşturulması gerekmektedir.
- Paket kullanıcıları paket üzerinde yapılan değişikliklerden haberdar edilmelidir. Onlar için kullanabilecekleri yeni bir paket sürümünün oluşturulması yanı sıra, mevcut kodun kırılmasını önlemek için paketin eski versiyonlarının da paralel kullanıma açık tutulması gerekir. Sadece bu durumda paket kullanıcıları paket üzerinde yapılan değişikliklerinden etkilenmeden eski versiyonlarla çalışmaya devam edebilirler. Zaman içinde yeni paket versiyonuna geçerek, son değişiklikleri entegre ederler.

Tekrar kullanımı kolaylaştırmak için paket sürümlerinin oluşturulması şarttır. REP'e göre tekrar kullanılabilirlik (reuse) sürüm (release) ile direk orantılıdır. Sürüm ne ihtiva ediyorsa, o tekrar kullanılabilir.