

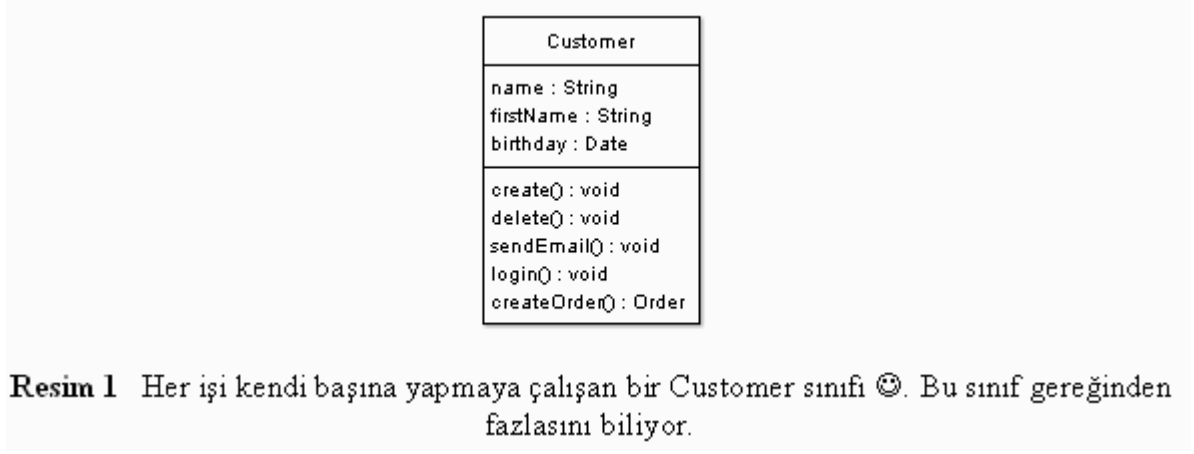
Single Responsibility Principle (SRP)

Tek Sorumluk Tasarım Prensibi

KurumsalJava.com

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

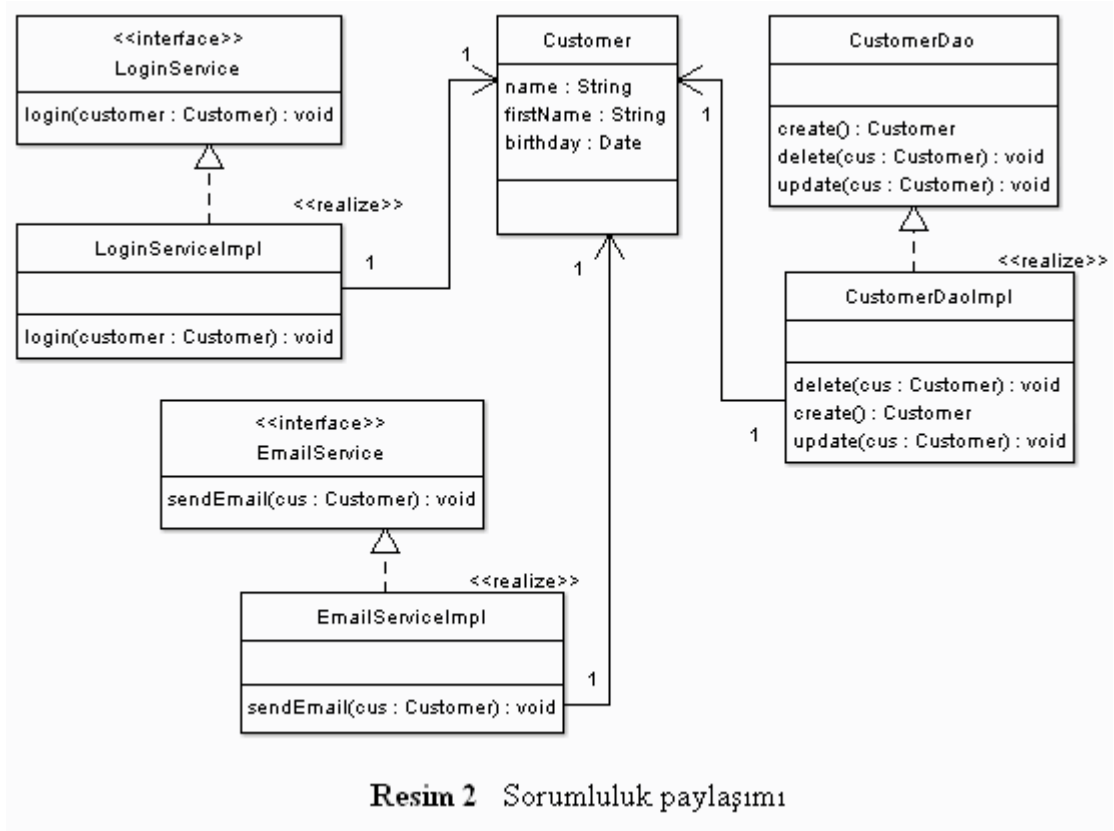
Resim 1 deki gibi bir sınıfa daha önce bir yerlerde rastlamışsınızdır. Bu sınıf kendisini bilgi bankasına ekleme, silme, müşteriye email gönderme, login yapma (shop sistemi olabilir) ve sipariş oluşturma işlemlerini yapabilmektedir.



Resim 1 Her işi kendi başına yapmaya çalışan bir Customer sınıfı 😊. Bu sınıf gereğinden fazlasını biliyor.

Büyük bir ihtimalle bu sınıfı programlayan programcı kendisi ile gurur duyuyor olmalıdır. Aslında böyle bir sınıfın ve programcının küçümsenecek hiçbir tarafı yok. Bu sınıf büyük emek harcanarak oluşturulmuş olabilir, çünkü ihtiva ettiği metotlar basit değildir. Lakin bu sınıf saatli bir bombadır, çünkü içinde bulunduğu ortamdaki her değişiklik, sınıfın yapısal değişikliğe uğramasına sebep olabilir. Bunun yanı sıra Customer sınıfında implemente edilen kod tekrar kullanılmak istendiğinde, Customer sınıfının bağımlı olduğu sınıf ve API'lerin bu sınıfla beraber kullanılma zorunluluğu vardır, yani Customer sınıfı gittiği yere beraberinde kullandığı diğer sınıflar ve API'leri götürür. Bu kodun tekrar kullanımını sağlamak için istenmeyen bir durumdur.

Customer sınıfını test edilebilir ve bakımı kolay bir hale getirmek için, sahip olduğu sorumlulukların başka sınıflara yüklenmesi gerekmektedir. Bunun bir örneğini resim 2 de görmekteyiz.



Resim 2 de yer alan Customer sınıfı sahip olduğu sorumlulukların başka sınıflara yüklenmesiyle hafiflemiş ve değişikliklere karşı daha dayanıklı bir hale gelmiştir. Bunun yanı sıra bu sınıfın test edilebilirliği yükselmiştir. Bu ve diğer sınıfların değişmek için artık sadece bir sebebi vardır. Bunun sebebi sadece bir sorumluluk sahibi olmalarında yatmaktadır.

Bir sınıfın sorumluluğunu sadece bir sebepten dolayı değişebilir olması olarak tanımlayabiliriz. Eğer bir sınıfın değişikliğe uğraması için birden fazla sebep varsa, bu sınıf birden fazla sorumluluk taşıyor demektir. Bu durumda sınıfta sadece bir sorumluluk kalacak şekilde, sorumlulukların diğer sınıflarla paylaşılması yoluna gidilmelidir.

EOF (End Of Fun)

Özcan Acar