**Solid Principles**

Solid prensibinde amaç, yazılımda sürdürülebilirliği sağlamaktır.

1. Single Responsibility Principle (SRP)
2. Open - Close Principle (OCP)
3. Liskov Substition Principle (LSP)
4. Interface Segregation Principle (ISP)
5. Dependency Inversion Principle (DIP)

**Single Responsibility Principle (SRP)**



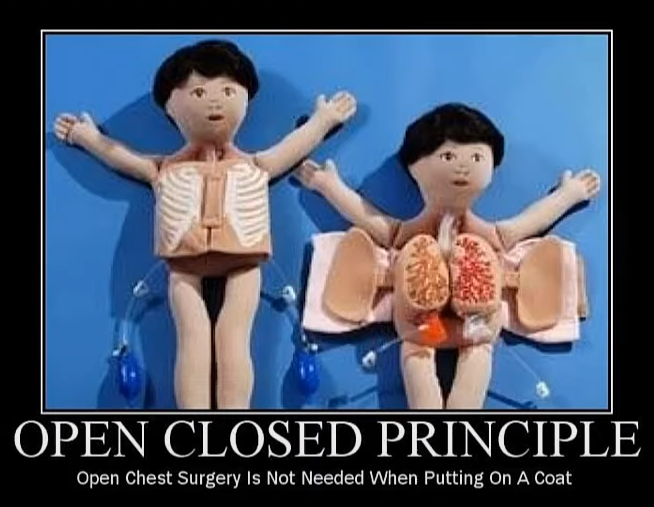
metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturulduYanlış.

Tek fonksiyonda iki ayrı işlem yapmamalıyız.

Ayrıca iki araba classı olduğunu düşünelim. Ortak parametreleri, fonksiyonları ortak base classa taşımak gerekir.

**Open - Close Principle (OCP)**



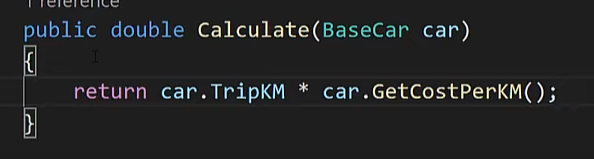
Geliştirmeye açık, değiştirmeye kapalı olma durumu.

metin, ekran görüntüsü, yazılım, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Buraya yeni bir araba eklersem burdaki kodu güncellemem gerekecek bu durumda bu istemeyeceğimiz bir olaydır.

Bunun yerine BaseCar classını abstract tanımlarız ve bir abstract metotla kalıtım alan classlara override etmesini sağlarız.



Kısacası dinamiklik diye özetleyebiliriz. Var olan classı değiştirmeyiz, yeni ekleneni güncelleriz.