**Design Pattern’ler**

**Patternler 3 sınıfa ayrılır:**

**-**Creational Patterns (Yaratıcı Desenler): Bu patternler nesnelerin oluşma şeklini yönetir. Amacı nesne oluşumunu daha esnek hale getirmektir.

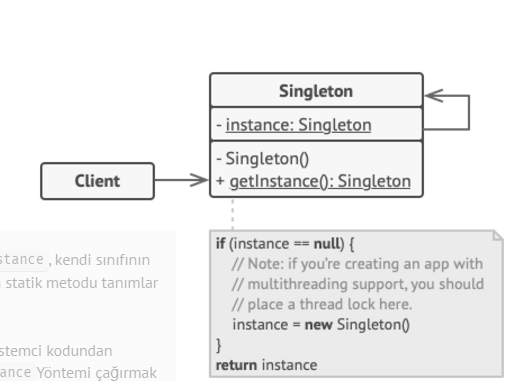
- Structural Patterns (Yapısal Desenler): Nesnelerin ve yapıların bir araya gelerek büyük yapılar oluşturmasına yardımcı olur.

-Behavioral Patterns (Davranışsal Desenler): Nesneler arası iletişimi yönetir.

Patternler’in ana başlıkları bu şekildedir.

**Creational Patterns**

**Singleton:** Singleton bir sınıfın yalnızca bir örneğinin olup o sınıfı public olarak tüm sınıflarda kullanmamızdır.



**Factory Method:** Bu pattern üst sınıfta nesne oluşturmak için arayüz sağlayan ancak nesne oluşturma işini alt sınıflara bırakan patterndir.

metin, taslak, diyagram, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Abstract Factory:** Bu pattern birbiriyle ilişkili veya bağımlı nesneleri somut sınıflarını belirtmeden oluşturmak içi kullanılır. Factory methoda benziyor ancak farkı birden çok ürün ailesiyle bağlanabilmesi.

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

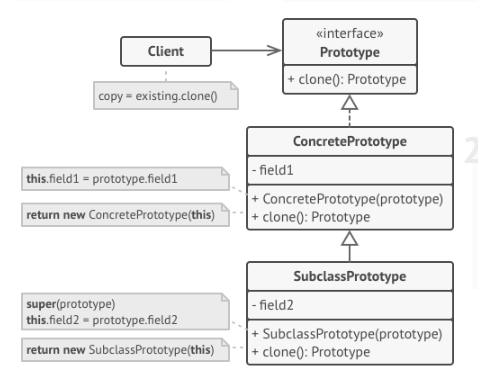
Açıklama otomatik olarak oluşturuldu

**Builder:** Bu pattern karmaşık olan nesneleri adım adım oluşturmamızı sağlıyor. Nesne oluşturulurken her bir parçası birer adımda belirlenir ve en sonunda final bir nesne oluşur.

metin, taslak, ekran görüntüsü, dikdörtgen içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Prototype:** Bu pattern nesneleri kopyalamamıza yarar. Bu pattern sayesinde büyük ve karmaşık nesneleri tekrar tekrar oluşturmak yerine klonunu oluşturabiliyoruz.



**Structural Patterns**

**Adapter:** Bu pattern birbirinden uyumsuz 2 arayüzün birlikte çalışmasını sağlar.

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

Açıklama otomatik olarak oluşturuldu