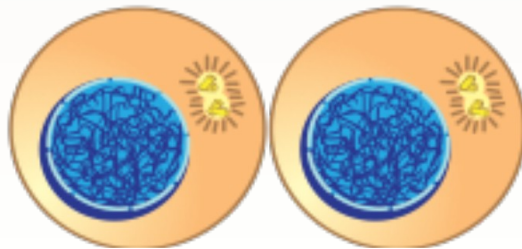
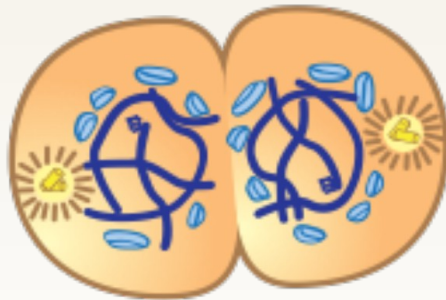
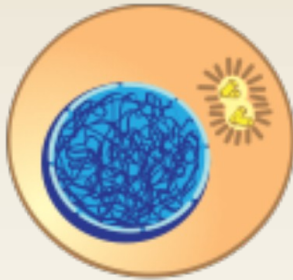


Prototype Örüntüsü

Prototype

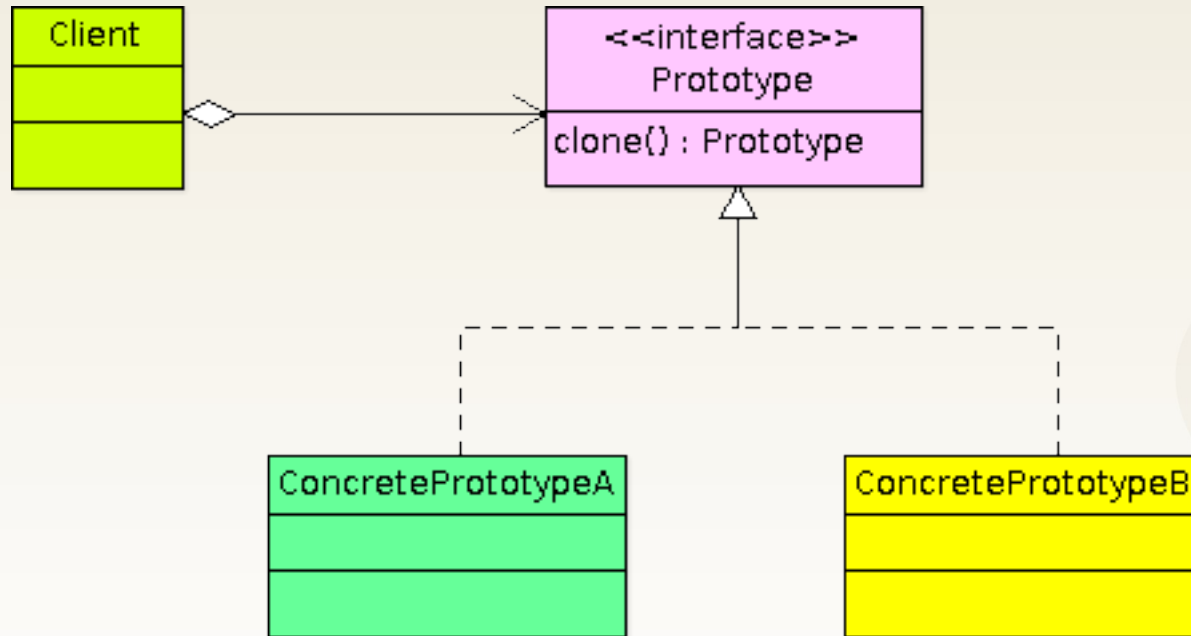


- Bazı durumlarda nesnelerin kullanılabilir duruma gelebilmeleri için **new operatörü** ile yaratılması ve **initialize edilmesi** birkaç adımdan oluşabilir ve uzun sürebilir
- Bu tür senaryolarda önceden bir nesne **initialize edilmiş biçimde** hazırlanarak tutulur
- Yeni nesneler bu prototip nesne **klonlanarak** elde edilir

Prototype

- Örneğin bir **grafik editör aracında** bir resmi edit ettiğimizi düşünelim. **Resim** üzerinde **border oluşturma, bevel ekleme, tonlama** gibi işlemler yaptığımızı farz edelim
- Bu aşamada resmin bu halini Ctrl+C, Ctrl+V ile **kopyala yapıştır** yapmaya çalışalım
- Uygulamanın kopyala yapıştır için izleyeceği yol hali hazırda belirli bir aşamaya gelmiş bu nesnenin on anda bire bir aynı kopyasını (**klon**) oluşturmaları olacaktır

Prototype Sınıf Diagramı



Java ve Prototype

- Herhangi bir nesnenin prototip olarak oluşturulması ve bu nesneden klonlama yolu ile diğer yeni nesnelerin oluşturulması için Java **clone** metodunu sunar
- Ancak clone metodunun kendisi **tasarımsal olarak problemlidir**
- Bu problemlili hali implementasyonda da geliştiricilerin **kolaylıkla bazı hatalara düşmelerine** neden olmaktadır

Clone Metodunun Problemleri

- Herhangi bir nesnenin klonlanabilir olması için öncelikle sınıfının **Cloneable** arayüzünü implement etmesi gerekir
- Ancak clone metodu bu arayüzde tanımlı değildir, aksine doğrudan **java.lang.Object** sınıfında tanımlıdır
- Fakat burada da **protected** olarak tanımlıdır
- Dolayısı ile clone metodunu **public modifer ile override etmeden çağırmak** mümkün değildir

Clone Metodunun Problemleri

- Normal şartlarda bir arayüz istemcilerine onu implement eden sınıfın **ne kabiliyete sahip olduğunu** içerdiği metot tanımları ile ilan eder
- **Cloneable** arayüzü ise Object sınıfındaki **clone metodunun davranışını** değiştirir
- Cloneable arayüzü implement edili değilken bir nesnenin clone metodu çağrılırsa **CloneNotSupportedException** fırlatılır
- Eğer arayüz implement edilirse metot bu durumda nesnenin **attribute'ları teker teker kopyalayarak** kopya nesneyi oluşturur

Clone Metodunun Problemleri

- Clone metodunun spesifikasyonuna göre **`x.clone().getClass() == x.getClass()`** olmalıdır
- Ayrıca nesnenin kopyası yaratılırken **herhangi bir constructor çağrılmayacağı** belirtilmiştir
- Ancak bunları clone metodunu **override eden alt sınıflarda** garanti edecek bir mekanizma mevcut değildir
- Aslında **eğer sınıf final ise alt sınıf olamayacağı için** constructor çağrısı ile nesne kopyası oluşturmak herhangi bir sorun da çıkarmayacaktır

Clone Metodunun Problemleri

- Ancak final olmayan sınıflarda clone metodu override edildiğinde nesne kopyası mutlaka **super.clone()** çağrısı ile oluşturulmalıdır
- Klonlanan nesnenin attribute'ları kopyalanırken **yüzeysel kopyalama** (swallow copy) yapılır
- Bu nedenle asıl nesne ve kopya **aynı nesnelere refer** ediyor olabilirler
- Attribute'lardaki **mutable nesneler** için bu ciddi sorun teşkil edebilir

Clone Metodunun Problemleri

- Böyle durumlarda **derin kopyalama** (deep copy) yapmak ve attribute'lara değerlerini bu derin kopyalama sırasında **ayrı ayrı oluşturmak ve atamak** gerekebilir
- Bunun için de attribute'ların **final tanımlanmamış olması** gerekir
- Constructor'larda olduğu gibi clone metodunda da **final olmayan metotların invoke edilmesi** risk teşkil edebilir
- Burada meydana gelebilecek bir hata nesnenin yaratımını başarısız kılacaktır

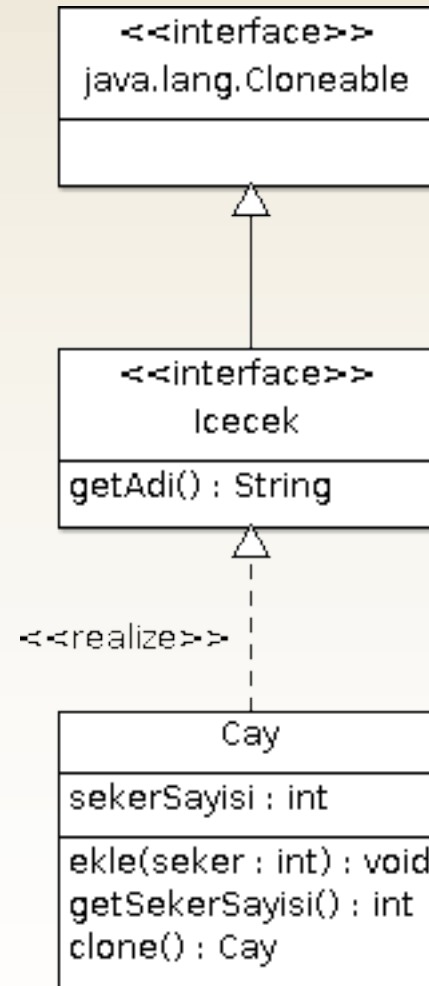
Clone Metoduna Alternatif: Copy Constructor

- Clone metodunu kullanmak yerine önerilen çözümlerden birisi “**copy constructor**” yaklaşımıdır
- Bir sınıf **constructor argümanı** olarak **kendi tipinden bir nesne** kabul eder
- Constructor içerisinde **input nesnenin state'i** yeni yaratılan nesneye kopyalanır

LAB ÇALIŞMASI: Prototype

- Kafedeki bazı yiyecek veya içecekler için günün belirli saatlerinde müşterilerden oldukça fazla talep gelmektedir. Örneğin tek şekerli çay, orta şekerli kahve gibi.
- Bu tür siparişlerin hızlı biçimde hazırlanarak karşılanmasını sağlamak gerekmektedir
- Bunun için, yoğun olarak siparişi verilen içecek veya yiyeceklerinin hızlı biçimde oluşturulmasını sağlayan bir yapı kurulmalıdır

LAB ÇALIŞMASI: Prototype



Prototype

Örüntüsünün Sonuçları

- Halihazırda yaratılmış, initialize edilmiş ve belli bir state'e gelmiş **nesnenin o anki gösteriminin elde edilmesi** çok daha kolay ve hızlı olur
- Yaratılacak nesnenin **sınıfının çalışma zamanında belirlenmesine** de imkan tanır
- Klon metodunun implemantasyonu nesne hiyerarşisinin **derin kopyasının** oluşturulmasında veya **döngüsel bağımlılıklarda** zor olabilir

İletişim



www.harezmi.com.tr

www.java-egitimleri.com



info@harezmi.com.tr

info@java-egitimleri.com



[@HarezmiBilisim](https://twitter.com/HarezmiBilisim)

[@JavaEgitimleri](https://twitter.com/JavaEgitimleri)