



1906003172019

Tasarım Desenleri

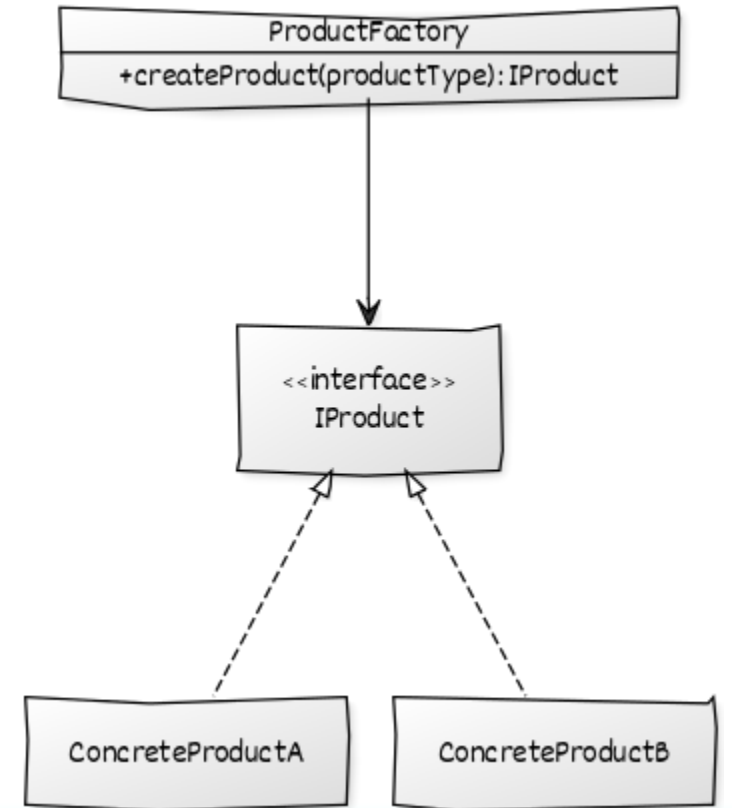
Dr. Öğr. Üy. Önder EYECİOĞLU
Bilgisayar Mühendisliği



Factory Patterns-Creational (Oluřturucu)

Konsept

Fabrika modeli , uygulama mimarisi tasarlanırken göz önünde bulundurulması ve uygulanması gereken en önemli ilke olan **sınıflar arasında gevşek bağlantı sağlar**. Gevşek bağlantı, somut uygulamalar yerine soyut varlıklara karşı programlama yaparak uygulama mimarisinde tanıtılabilir. Bu, mimarimizi yalnızca daha esnek kılmakla kalmaz, aynı zamanda daha az kırılğan hale getirir.



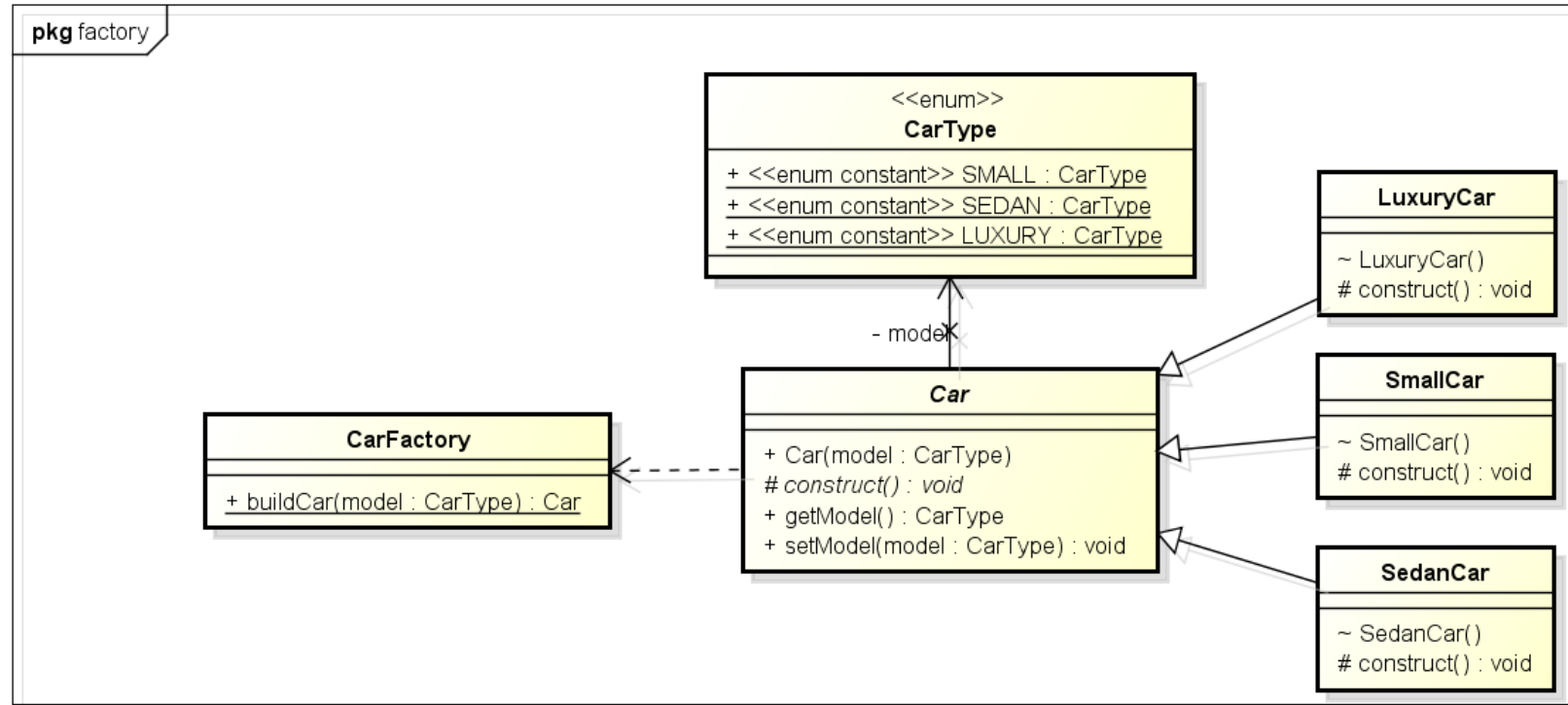
Factory Patterns-Creational (Oluřturucu)

Bilgisayar Dünyası Örneęi

- küçük, sedan ve lüks olmak üzere 3 tip araba üretebilen bir araba fabrikası örneęini kullanan ortak bir senaryoyu göstermektedir. Bir araba inşa etmek, aksesuarların tahsis edilmesinden son makyaja kadar birçok adım gerektirir. Bu adımlar, programlamada yöntemler olarak yazılabilir ve belirli bir araba tipinin bir örneęini oluştururken çağrılmalıdır.
- Eęer şanssızsak, SmallCaruygulama sınıflarımızda araba tiplerinin (örn.) örneklerini oluşturacağız ve böylece araba inşa mantığını dış dünyaya açacağız ve bu kesinlikle iyi deęil. Ayrıca kod merkezi olmadığı için araba yapım sürecinde deęişiklik yapmamızı da engelliyor ve tüm beste sınıflarında deęişiklik yapmak mümkün görünmüyor.

Factory Patterns-Creational (Oluşturucu)

İllüstrasyon



Abstract Factory Patterns-Creational (Oluşturucu)

GoF Tanımı: Abstract Factory tasarım deseni birbirleri ile ilişkili ürün ailesini oluşturmak için bir arayüz sağlar.

Konsept

Soyut fabrika deseni , başka bir yaratıcı tasarım desenidir ve **fabrika deseni** üzerinde başka bir soyutlama katmanı olarak kabul edilir . Soyut fabrika örüntüsü, bir istemcinin, somut sınıfları belirtmeden nesneleri oluşturmasını sağlar. Bunun için bir fabrika nesnesi kullanır. Fabrika nesneleri istenen türde nesne oluşturmakla görevlidir. Aşağıda soyut fabrika örüntüsünün sınıf diyagramı görülmektedir. Burada istemci SoyutFabrika arayüzünü gerçekleştiren SomutFabrika nesnelerinden birini üretir. Bu somut fabrika da SoyutÜrün tiplerinden birinden istenen türde nesneleri üretmekle görevlidir. Sonrasında istemci hangi somut fabrika ve hangi somut ürün olduğu ile ilgilenmeden soyut fabrika arayüzünü kullanarak istediği tipte ürünü üretmek için metodu çağırarak nesne üretir. Böylece istemci fabrika ve ürün detaylarından tamamen soyutlanmış olur.

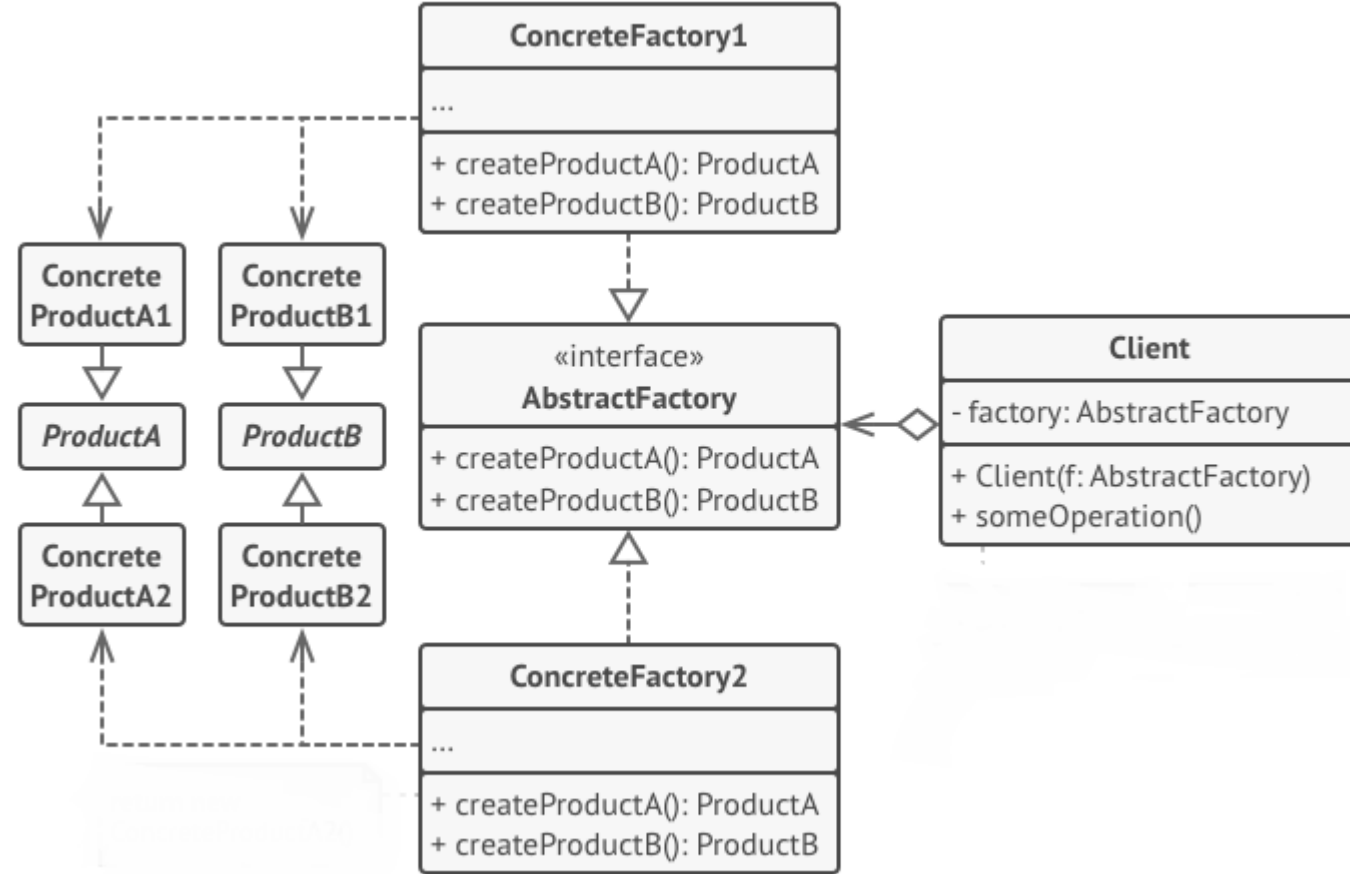


Abstract Factory Patterns-Creational (Oluřturucu)

Konsept

Factory tasarım deseninde bir ürünün oluşturulması soyutlanmış iken Abstract Factory deseninde birbirleri ile ilişkili ürün ailelerinin oluşturulması soyutlanmıştır. **Factory üreten Factory deseni olarak da düşünölebilir.**

Anlayacağımız; birden fazla ürün ailesi ile çalışmak zorunda kaldığımız durumlarda, istemciyi bu yapılardan soyutlamak için Abstract Factory doğru bir yaklaşım olacaktır.



Abstract Factory Patterns-Creational (Oluřturucu)

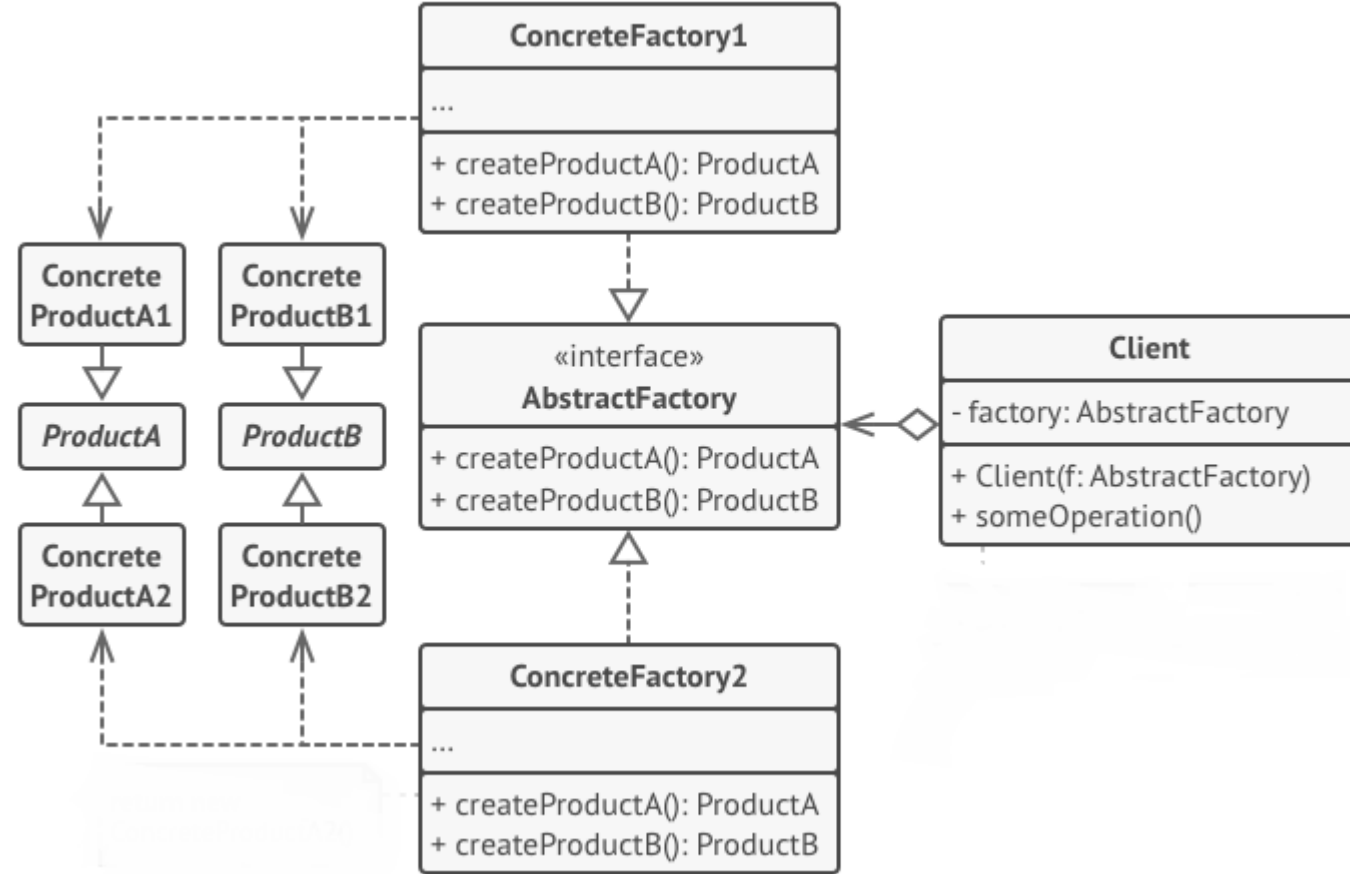
Konsept

ProductA, ProductB: Temel sınıflarımız, soyuttur ve oluşturulmasını istediğimiz sınıflar bunlardan türer.

ConcreteProduct: Üretmek istediğimiz sınıflardır.

AbstractFactory: Herbir sınıfın oluşturulması için metotların tanımlandığı arayüzdür.

ConcreteFactory: AbstractFactory arayüzünü uygulayarak gerekli sınıfların oluşturulmasını sağlar.



Abstract Factory Patterns-Creational (Oluřturucu)

Bilgisayar Dünyası Örneęi

- Soyut fabrika modelini kullanarak küresel otomobil fabrikasını tasarlayalım.
- Küresel operasyonları desteklemek için , farklı ülkeler için farklı araba yapım tarzlarını desteklemek için sistemi geliřtirmemiz gerekecek. Örneęin bazı ülkelerde direksiyonu sol tarafta, bazı ülkelerde ise saę tarafta görüyoruz. Arabaların farklı kısımlarında ve yapım süreçlerinde bunun gibi daha birçok farklılık olabilir.
- Soyut fabrika modelini tarif etmek için 3 tür markayı ele alacaęız – ABD , Asya ve dięer tüm ülkeler için varsayılan . Birden fazla konumun desteklenmesi, kritik tasarım deęiřikliklerine ihtiyaç duyacaktır.

Abstract Factory Patterns-Creational (Oluřturucu)

Bilgisayar Dünyası Örneęi

- Her řeyden önce, sorun bildiriminde belirtilen her lokasyonda araba fabrikalarına ihtiyacımız var. yani USACarFactory , AsiaCarFactory ve DefaultCarFactory . řimdi, uygulamamız kullanıldığı yeri tespit edecek kadar akıllı olmalı, bu yüzden dahili olarak hangi araba fabrikası uygulamasının kullanılacağını bile bilmeden uygun araba fabrikasını kullanabilmeliyiz. Bu aynı zamanda bizi belirli bir konum için yanlış fabrikayı arayan birinden de kurtarır.
- Temel olarak, konumu belirleyecek ve kullanıcıya tek bir ipucu bile vermeden dahili olarak doğru araba fabrikası uygulamasını kullanacak başka bir soyutlama katmanına ihtiyacımız var. Soyut fabrika deseninin çözmek için kullanıldığı sorun tam olarak budur.

Abstract Factory Patterns-Creational (Oluşturucu)

İllüstrasyon

