JAVA Eğitimleri

Tasarım Örüntüleri
Sorular & Cevaplar

Örüntülerin Temel Prensipleri

- GOF tasarım örüntülerinin altında yatan temel prensipler nelerdir?

Örüntülerin Temel Prensipleri

JAVA Eğitimleri

- Sistemde değişen veya değişiklik gösteren kısımları tespit edip, bunları arayüz, soyut sınıf gibi yapıların ardında gizlemek (**encapsule** etmek)
- Inheritance yerine mümkün olduğunca **composition**'ı tercih etmek
- Her zaman için soyut tiplere (**abstract data types**) depend etmek

Strategy ve State

- Strategy ve State örüntüleri birbirlerine hangi noktalardan benzerler ve farklılık gösterirler?

Strategy ve State

- Her iki örüntüde de davranışın dinamik olarak değişmesi söz konusudur
- Strategy algoritmayı encapsule eder
- State ise state bilgisini encapsule eder, state bilgisini davranış olarak ifade etmeyi sağlar

Strategy ve Template Method

JAVA Eğitimleri

- Strategy ve Template Method örüntüleri birbirlerine hangi noktalardan benzerler ve farklılık gösterirler?

Strategy ve Template Method

- Her iki örüntü de belirli bir algoritmayı, yani davranışı encapsule eder
- Ancak Strategy bunu yapmak için composition'ı, Template Method ise inheritance'ı tercih eder

Adapter, Proxy ve Decorator

- Adapter, Proxy ve Decorator örüntüleri birbirlerine hangi noktalardan benzerler ve farklılık gösterirler?

Adapter, Proxy ve Decorator

- Adapter örüntüsü subject'den farklı yeni bir tip ortaya çıkarır ancak davranış olarak ilave yeni bir şey eklemez, asıl iş yine subject tarafından gerçekleştirilir
- Proxy ve Decorator örüntüleri ise aynı arayüz üzerinden çalışırlar
- İkisi de ilave iş yaparlar
- Proxy subject'e işi delege edebilir veya etmeyebilir, Decorator'de ise asıl iş subject tarafından mutlaka yapılır

Adapter ve Bridge

- Adapter ile Bridge örüntüleri birbirlerine hangi noktalardan benzerler ve farklılık gösterirler?

Adapter ve Bridge

- Her iki örüntü de benzer bir yapıya sahiptirler
- Adapter farklı bir arayüzü sisteme uyarlar
- Bridge ise arayüz ve bunun implemantasyonunu birbirlerinden ayırır
- Adapter halihazırda geliştirilmiş iki sistemi veya mevcut bir sistemi başka bir sistem içerisinde kullanılabilir hale getirir
- Bridge ise yeni geliştirilmeye başlanan bir sistemi farklı implementasyon terichlerine açık biçimde geliştirmeyi sağlar
- Arayüz ve implemantasyonun birbirlerinden bağımsız ilerlemelerini hedefler

Adapter ve Façade

- Adapter ile Façade örüntüleri birbirlerine hangi noktalardan benzerler ve farklılık gösterirler?

Adapter ve Façade

- Adapter mevcut iki farklı arayüzü birbirleri ile konuşturur
- Façade ise yeni bir arayüz tanımlar ve sistem bu arayüz üzerinden çalışır
- Her ikisi de asıl nesneleri wrap ederler ve asıl işi arka taraftaki bu nesnelere delege ederler

Singleton ve Flyweight

- Singleton ile Flyweight örüntüleri birbirlerine hangi noktalardan benzerler ve farklılık gösterirler?

Singleton ve Flyweight

- Her ikisi de sistemdeki nesne sayılarını kontrol altında tutmayı hedefler
- Singleton belirli bir tipten sadece tek bir nesne olmasını sağlar
- Flyweight'de ise belirli tipten nesne sayısı genelde tek olsa bile, duruma göre birden fazla da olabilir
- Nesnelerin state bilgisi singleton'da tamamen kendi içlerinde yönetilir, ancak Flyweight'de ise state bilgisi dahili ve harici olarak ikiye ayrılır
- Harici state bilgisi nesne dışında yönetilir veya hesaplanır

Composite, Visitor ve Iterator

- Composite, Visitor ve Iterator örüntüleri arasında nasıl bir ilişki vardır?

Composite, Visitor ve Iterator

JAVA Eğitimleri

- Nesne hiyerarşileri veya nesne grupları üzerinde işlem yapmayı sağlayan Visitor ve Iterator örüntüleri sıklıkla Composite örüntüsü ile birlikte kullanılır
- Iterator, Composite nesne hiyerarşisini dolaşmak için kullanılır
- Visitor ise bu hiyerarşi üzerinde uygulanacak işlemlerin tek bir yerde toplanmasını sağlar

İletişim

www.harezmi.com.tr

www.java-egitimleri.com

info@harezmi.com.tr

info@java-egitimleri.com

@HarezmiBilisim

@JavaEgitimleri