

Nesne Temelli Tasarım ve Tasarım Desenleri

Giriş

“Karmaşık, uzun ömürlü, geliştirme ve bakım destek maliyetleri düşük, kaliteli yazılımlar nasıl geliştirilir?” hepimizin cevap aradığı önemli bir sorudur. Bu soru, yazılımın geliştirme sürecinin her kısmını ilgilendirir. Bu eğitim kapsamında; yazılım geliştirme sürecinin detaylı nesne tasarımı bölümü üzerinde durulacaktır. Bu alanda dünya çapında yaşanan iyi tecrübeler, ortaya koyulan yaklaşımlar, prensipler ve farklı desenler incelenecektir. Bu tecrübelerin, yazılım geliştirme kalitesinin yükseltilmesine katkıları ortaya çıkarılacaktır.

Ön koşul

C#,Java,C++ gibi bir nesne temelli programlama dilini temel seviyede bilmek.

Hedef Kitle

İyi bir nesne temelli tasarım ve geliştirme alt yapısına sahip olmak isteyen uygulama yazılımı ve alt yapı bileşenleri (Framework) geliştiricileri,

Nesne Temelli programlama, Tasarım Desenleri konularını öğrenmek veya eksikliklerini gidermek isteyen “Bilgisayar Mühendisliği, Yazılım Mühendisliği, Bilgisayar Programcılığı” öğrencileri

Uygulama

Eğitimin uygulama kısmında; her bir desenin kullanımını gösteren kod örnekleri verilecektir. Pratik uygulamalar için .NET platformu ve C# dili kullanılacaktır. Geliştirme ve sunum ortamı ise, Visual Studio olacaktır.

Eğitim içeriği

Nesne Temelli yaklaşım ve Kavramlar

- Yapısal programlamadan, Nesne Temelli yaklaşıma geçiş
- Soyutlama ve genelleştirme
- Sınıf ve Nesne kavramları
- Kapsülleştirme (Encapsulation), Kalıtım (Inheritance), Çok şekillilik (Polymorphism)
- Soyut sınıflar ve Arayüzler (Interfaces)

Nesne Temelli Tasarım

- Nesnelere arası bağılıklar (Sıkı, Gevşek, Bağılılık olmama)(Tight Coupling, Loose Coupling, Decoupling)
- Yüksek Yakınlık (High Cohesion)
- İlginin ayrıştırılması (Separation Of Concerns)
- Delegasyon
- Delegasyon & Kalıtım yaklaşımları inceleme

Sınıf tasarım prensipleri

- Tek sorumluluk (Single Responsibility) Prensibi
- Açık kapalı (Open Closed) Prensibi
- Liskov'un yerine geçme (Liskov Substitution) Prensibi
- Arayüz ayrım (Interface Segregation) Prensibi
- Bağımlılığın ters çevrilmesi (Dependency inversion) Prensibi

Paket tasarım prensipleri

- Genel Tekrar kullanım Prensibi (Common Reuse Principle - CRP)
- Genel Kapama Prensibi (Common Closure Principle - CCP)

- Tekrar kullanılabilirlik/Sürüm denklik prensibi (Reuse/Release Equivalence Principle - REP)
- Dönerli Bağımlılık Prensibi (Acyclic Dependencies Principle - ADP)
- Durağan Bağımlılık prensibi (Stable Dependencies Principle - SDP)
- Durağan Soyutlar prensibi (Stable Abstractions Principle - SAP)

Tasarım desenlerine giriş

- Tasarım desenlerinin önemi
- Nesne temelli yazılım geliştirmeye katkıları
- Tasarım desenleri ve Yazılım alt yapı bileşenleri (Frameworks)

GOF Tasarım Desenleri - I (Detaylı İnceleme)

- 1) Cephe (Facade)
- 2) Tek Nesne (Singleton)
- 3) Fabrika Metodu (Factory Method)
- 4) Strateji (Strategy)
- 5) Adaptör (Adapter)
- 6) Şablon Metodu (Template Method)
- 7) Gözlemci (Observer)
- 8) Aracı (Mediator)
- 9) Vekil (Proxy)
- 10) Durum (State)
- 11) Komut (Command)
- 12) Bileşik (Composite)
- 13) İnşacı (Builder)
- 14) Sorumluluk Zinciri (Chain of Responsibility)
- 15) Köprü (Bridge)
- 16) Soyut Fabrika (Abstract Factory)
- 17) Dekorator (Decorator)

GOF Tasarım Desenleri - II (Sadece kısa bilgilendirme)

- 18) Tekrarlayıcı (Iterator)
- 19) Prototip (Prototype)
- 20) Küçük Nesnelere (Flyweight)
- 21) Arşivci (Memento)
- 22) Yorumlayıcı (Interpreter)
- 23) Ziyaretçi (Visitor)

Eğitim Süresi

Eğitim Adı	Tarih	Saati
Nesne Temelli Tasarım ve Tasarım Desenleri Eğitimi	24-25 Ekim (Cumartesi-Pazar) ve 31 Ekim - 1 Kasım (Cumartesi-Pazar)	4 Gün - 28 Saat