

.NET Tasarım Kalıpları (Design Patterns) Eğitimi

Açıklama

Yazılımcılar, uygulamaları geliştirirken veya yazılım uygulaması Yaşam Döngüsünde yaygın ve tekrarlanan birçok sorunla karşılaşmaktadır. Tasarım Kalıpları bir yazılımın yarattığı problemlerde en esnek çözümleri sunmaktadır. Yeniden kullanılabilir çözümler sayesinde birçok yazılımcının deneyimlerini aktarmakta fayda sağlar.

Nesne yönelimli dünyadaki Tasarım Modelleri, gerçek dünyadaki uygulama geliştirmede tekrar tekrar ortaya çıkan yaygın yazılım tasarım sorunlarına yeniden kullanılabilir bir çözümdür. Birçok durumda kullanılacak problemlerin nasıl çözüleceğine dair bir şablon veya açıklamadır.

Eğitim İçeriği

Tasarım Kalıplarına Giriş

- Tasarım Kalıpları Nedir?
- Tasarım Kalıplarının Önemi
- Tasarım Kalıpları Kategorileri

SOLID Prensipleri

- Single responsibility(SRP)
- Open/Close (OCP)

- Liskov Substitution(LSP)
- Interface Segregation(ISP)
- Dependency Inversion(DIP)

Yaratıcı Tasarım Kalıpları

- Singleton Design Pattern (SRP)
- Factory Design Pattern (OCP)
- Abstract Factory Design Pattern
- Builder Design Pattern
- Prototype Design Pattern

Yapısal Tasarım Kalıpları

- Adapter Design Pattern
- Bridge Design Pattern:
- Composite Design Pattern
- Decorator Design Pattern
- Facade Design Pattern.
- Flyweight Design Pattern
- Proxy Design Pattern

Davranışsal Tasarım Kalıpları

- Chain Of Responsibility Design Pattern
- Command Design Pattern
- Interpreter Design Pattern.
- Iterator Design Pattern
- Mediator Design Pattern
- Memento Design Pattern
- Observer Design Pattern
- State Design Pattern
- Strategy Design Pattern.
- Template Design Pattern.
- Visitor Design Pattern

Ön Koşullar

Herhangi bir ön koşul bulunmamaktadır.