Clean Code and Code Refactoring Eğitimi

Süre: **3 Gün**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Overview

İyi bir yazılım geliştiricinin sahip olması gereken iki önemli beceri vardır:**temiz ve anlaşılır kod yazmak** ve **mevcut kodun dışarıdan gözlemlenen davranışını değiştirmeden iç yapısını iyileştirmek**.

Clean Code, yazılım kodunun insanlar tarafından kolayca okunabilmesini, anlaşılabilmesini ve gerektiğinde değiştirilebilmesini sağlayan prensipler bütünüdür. Temiz kod yazımında amaç, yalnızca çalışan bir yazılım oluşturmak değil, aynı zamanda kodun başka geliştiriciler tarafından da rahatlıkla anlaşılmasını sağlamaktır.

Bu yaklaşımda kodun mantıksal olarak düzenlenmesi, farklı bileşenler arasındaki ilişkilerin açık biçimde tanımlanması ve değişken, metot ve sınıf isimlerinin anlamlı olması önem taşır. Böylece geliştiriciler, kodun hangi görevi yerine getirdiğini daha kolay anlayabilir ve yazılım üzerinde güvenle değişiklik yapabilir.

Refactoring ise yazılımın mevcut işlevselliğini ve dış davranışını koruyarak iç yapısının yeniden düzenlenmesi sürecidir. Bu süreç, kodun daha sade, anlaşılır, sürdürülebilir ve değişikliklere açık hale getirilmesini hedefler.

Clean Code ve Kod Refactoring Eğitimi, katılımcılara bu iki yaklaşımın temel prensiplerini ve uygulama tekniklerini öğretmeyi amaçlamaktadır. Eğitim boyunca okunabilir kod geliştirme, doğru isimlendirme, fonksiyon ve metot tasarımı, kod inceleme, Code Smell tespiti ve farklı refactoring yöntemleri ele alınacaktır.

Katılımcılar, öğrendikleri teknikleri kullanarak mevcut kodların kalitesini artırma, gereksiz karmaşıklığı azaltma ve yazılım bakım süreçlerini kolaylaştırma konusunda bilgi ve beceri kazanacaklardır.

What You Will Learn

Bu eğitim sonunda katılımcılar;

- Clean Code prensiplerini öğrenerek okunabilir ve sürdürülebilir kod yazabilecek,
- Kod biçimlendirme ve anlamlı isimlendirme standartlarını uygulayabilecek,
- Fonksiyon ve metot yapılarını daha anlaşılır ve düzenli hale getirebilecek,
- Kod inceleme ve dokümantasyon süreçlerini geliştirebilecek,
- Code Smell (kod kokuları) problemlerini tespit edebilecek,
- Refactoring teknikleriyle kodun davranışını değiştirmeden iç yapısını iyileştirebilecek,
- Metot, sınıf ve nesne yapılarını yeniden düzenleyebilecek,
- Performans ölçümleme ve tasarım kalıplarına dayalı refactoring yaklaşımlarını tanıyabileceklerdir.

Clean Code and Code Refactoring Eğitimi

Süre: **3 Gün**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Outline

1. Clean Code'a Giriş

- Clean Code nedir?
- Clean Code neden önemlidir?
- Clean Code prensipleri nelerdir?

2. Kod Biçimlendirme ve İsimlendirme

- Kod biçimlendirme ve girintileme kuralları
- Değişken, metot ve sınıflar için isimlendirme standartları
- Anlamlı isimlendirme ile kod okunabilirliğinin artırılması

3. Fonksiyonlar ve Metotlar

- Fonksiyon parametreleri ve dönüş değerleri
- Fonksiyonların isimlendirilmesi ve düzenlenmesi

4. Kod İnceleme ve Dokümantasyon

- Kod inceleme süreçlerinin ne zaman kullanılması gerektiği
- Açık, anlaşılır ve yoruma yer bırakmayan kod açıklamalarının yazılması
- Kodun sürdürülebilirliğini destekleyen inceleme ve doğrulama yaklaşımları

5. Refactoring'e Giriş

- Refactoring nedir?
- Refactoring neden önemlidir?
- Refactoring uygulamalarının sağladığı faydalar

6. Refactoring Teknikleri ve Uygulamaları

- Code Smell problemlerinin tespit edilmesi ve giderilmesi
- Kod kalitesini iyileştirmeye yönelik teknikler
- Uygulamalı refactoring çalışmaları
- Performans iyileştirme
- Performans karşılaştırma ve ölçümleme (Performance Benchmarking)

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Metotların yeniden düzenlenmesi
- Sınıf ve nesne yapılarının refactoring işlemleri
- Sınıf hiyerarşilerinin yeniden yapılandırılması
- Tasarım kalıplarına dayalı refactoring teknikleri

7. Code Smell (Kod Kokuları)

- Code Smell kavramı ve kod kokularının belirlenmesi
- C# kodlamasında sık karşılaşılan Code Smell örnekleri
- Refactoring kullanarak kod kalitesinin artırılması