

# Test Driven Development (TDD) Eğitimi

Süre: 3 Gün

## Açıklama

TDD Eğitimi, birim testleri, mock ve refactoring konularını ele alan 3 günlük bir eğitim programıdır.

Test Driven Development programında kullanıcı hikayeleri, bağımlılık yönetimi, test kodu kalitesi, BDD ve test araçları uygulamalı çalışmalarla ele alınır.

Eğitmen eşliğinde gerçekleştirilen bu üç günlük eğitim, etkili bir uygulama için gerekli olan tüm adımları içeren TDD hakkında ayrıntılı bilgileri sağlar. Katılımcılar, Birim Testleri (Unit Tests), Kullanıcı Hikayeleri (User Stories), Tasarım, Kod İyileştirme, Çerçeveler hakkında bilgi alırlar ve bunları mevcut çözümlere nasıl uygulayabileceklerini öğrenirler. Ayrıca; kod bağımlılıkları, akış gereksinimleri ve sorunların erken tespiti gibi konuların etkilerini de ele alır. Bu, uygulamalı laboratuvar çalışmalarını içeren etkileşimli bir eğitimidir. Bu eğitimden en iyi şekilde yararlanılabilmesi için katılımcıların bu etkinliklere tam katılım göstermeleri önerilir. Bu eğitim, geliştiricilerin ve ekiplerin kodların tüm yaşam döngüsü boyunca kaliteli uygulama sürdürülebilirliği oluşturabilmeleri için ihtiyaç duydukları becerileri gösterir.

## Bu eğitimde neler öğreneceksiniz?

- Birim Testi ilkeleri ve uygulamaları
- Yazılım geliştirme ve testi süreçlerinde Birim Testlerinin önemi
- Etkili Birim Testleri yazma
- Etkili birim testleri özellikleri
- “Test edilen sistemi” izole etmek için sahte nesneleri kullanma
- Kodların etkili bir şekilde iyileştirilmesi
- İlk önce test et ve TDD yaklaşımlarının faydaları
- TDD yaklaşımının başarılı bir şekilde kabul edilmesine yardımcı olacak teknikler ve uygulamalar
- TDD'yi daha geliştirmek için Kabul Testi ve Davranışa Dayalı Geliştirmeyi (Acceptance Testing and Behavior Driven Development) kullanma

## Eğitim İçeriği

### TDD Temelleri

- TDD Nedir?
- Agile Manifesto
- TDD'nin Temel Felsefesi
- TDD Prensipleri
- Unit Test'in Amacı
- Test Yöntem ve Tipleri

## **Nesne Yönelimli Programlama Prensipleri (Özet)**

- Unit Test ve Araçlar
- Hello Unit Testing
- Hello Test First Development

## **Dependency Injection**

- IoC Nedir? Ne İşe Yarar
- Temel DI Prensipleri
- Test Kodunda DI'nın Önemi

## **Mock Kavramı**

- Fake Object ile Test Kodu Yazmak
- State ve Interaction Temelli Verifikasyon
- Stub ve Mock Kavramları
- Dinamik Mock Kütüphaneleri ile Çalışmak
- Harici kütüphanelerin Test Edilmesi ( Db, FileSystem vs.)

## **TDD ve Refactoring**

- Temel Refactoring Teknikleri
- Mevcut Bir Kodun Test Edilebilir Hale Gelmesi
- Temel OOP Prensipleri ve Test Koduna Etkisi
- Test Edilebilir Yazılım Tasarımı

## **Test Kodunun Kalitesi**

- İyi bir Unit Testin Yapısı
- Test Fixture'da Olması Gerekenler
- ObjectBuilder ve ObjectMother Kalıpları
- Test Smells
- Unit Test Patterns/Antipatterns

## **Davranış Odaklı Geliştirme (BDD)**

- TDD ile Farkları

- Acceptance Test Driven Development
- Specification by Example

### **Test Araçlarının Kullanımı**

- Code Coverage Kavramı
- Başarılı ve Başarısız Testlerin Ele Alınması
- Test Kodunun Organizasyonu
- Test Kodunun Tekrar Kullanımı

## **Katılım Koşulları**

---

Katılım için gerekli hazırlık ve teknik koşulları bu sayfadaki İletişim Formu üzerinden teyit edebilirsiniz.