



Bölüm 11: Test ve Hata Ayıklama

JAVA ile Nesne Yönelimli Programlama



Test ve Hata Ayıklama

- *Testing is not about finding bugs, it's about making sure that the system works as expected. **Michael Bolton***
- *The best way to predict the future is to create it. **Abraham Lincoln***
- *A bug is a feature that hasn't been discovered yet. **Unknown***
- *Quality is not an act, it is a habit. **Aristotle***
- *If you don't test, it's like playing Russian roulette with software. **Unknown***



Yazılım Geliřtirmede Test Türleri

- Birim Testi (Unit Testing)
- Modül Testi (Module Testing)
- Alt Sistem Testi (Sub-System Testing)
- Sistem Testi (System Testing)
- Kabul Testi (Acceptance Testing – Alfa/Beta Testi)



Yazılım Geliştirmede Test Türleri

- **Birim Testi (Unit Testing)**
 - Bireysel, bağımsız bileşenlerin test edilmesine odaklanır.
 - Her bileşenin tek başına doğru çalıştığından emin olunur.
- Modül Testi (Module Testing)
- Alt Sistem Testi (Sub-System Testing)
- Sistem Testi (System Testing)
- Kabul Testi (Acceptance Testing – Alfa/Beta Testi)



Yazılım Geliştirmede Test Türleri

- Birim Testi (Unit Testing)
- **Modül Testi (Module Testing)**
 - İlgili bileşenlerden oluşan bir modülün test edilmesidir.
 - Modül içindeki bileşenlerin entegrasyonu ve işlevselliği doğrulanır.
- Alt Sistem Testi (Sub-System Testing)
- Sistem Testi (System Testing)
- Kabul Testi (Acceptance Testing – Alfa/Beta Testi)



Yazılım Geliştirmede Test Türleri

- Birim Testi (Unit Testing)
- Modül Testi (Module Testing)
- **Alt Sistem Testi (Sub-System Testing)**
 - Alt sistemlerin arayüz uyumsuzluklarını test eder.
 - Bileşenler arasındaki iletişim ve veri akışının doğruluğunu kontrol eder.
- Sistem Testi (System Testing)
- Kabul Testi (Acceptance Testing – Alfa/Beta Testi)



Yazılım Geliştirmede Test Türleri

- Birim Testi (Unit Testing)
- Modül Testi (Module Testing)
- Alt Sistem Testi (Sub-System Testing)
- **Sistem Testi (System Testing)**
 - Alt sistemler arasındaki etkileşimlerin test edilmesi.
 - İşlevsel olan ve olmayan gereksinimlerin karşılandığının doğrulanması.
- Kabul Testi (Acceptance Testing – Alfa/Beta Testi)



Yazılım Geliştirmede Test Türleri

- Birim Testi (Unit Testing)
- Modül Testi (Module Testing)
- Alt Sistem Testi (Sub-System Testing)
- Sistem Testi (System Testing)
- **Kabul Testi (Acceptance Testing – Alfa/Beta Testi)**
 - Simüle veriler yerine gerçek verilerle sistemin test edilmesi.
 - Sistemin gerçek dünyadaki kullanılabilirliğini ve performansını doğrular.
 - Kullanıcılar ile beraber yapılır.



JUnit

- Bir *birim*, bir *sınıf* anlamına gelir.
- JUnit, temel olarak birim testi için tasarlanmış olsa da oldukça esnek.
- JUnit ayrıca şu tür testlerde de kullanılabilir:
 - Modül Testi: Birden fazla sınıfı içeren senaryoların test edilmesi.
 - Sistem Testi: Sistem seviyesindeki etkileşimlerin test edilmesi.



Regresyon Testi

- Yapılan deęişikliklerden sonra önceden çalışan her şeyin hala çalıştığını doğrulamak anlamına gelir.
- Regresyon Testinin Özellikleri
 - Belirleyici (*Deterministic*) Olmalıdır:
 - Testler, aynı koşullarda her seferinde aynı sonucu vermelidir.
 - Tekrarlanabilir (*Repeatable*) Olmalıdır:
 - Testler, farklı zamanlarda ve durumlarda tekrar edilebilir şekilde tasarlanmalıdır.



Otomatik Testler

- Kodu manuel olarak test etmek yerine, otomatik testler yazılmalı.
- Harcanan çaba aynı olur, ancak sonunda otomatik bir test elde edilir.
- Bu test, istenildiği zaman tekrar çalıştırılabilir.
- Yazılım bir kez doğru çalıştığında, bir daha test etmenin gereksiz olduğunu düşünmek yanlıştır.



Hata Ayıklayıcı (Debugger)

- Çalışan bir programın durumunu incelemeye olanak sağlar.
- Hata Ayıklayıcı ile Yapılabilecekler
 - Programı komut komut çalıştırma
 - Çalışan programın kaynak kodunu inceleme
 - Değişkenlerin değerlerini görüntüleme ve gerektiğinde değiştirebilme
 - Programa kesme noktaları (*breakpoint*) ekleyebilme
 - Programı belirtilen bir kesme noktasına kadar çalıştırma
 - Durdurulmuş programın durumunu (bir *core file* üzerinden) inceleme.



Profil Oluřturucu (Profil)

- Sonlandırılmıř bir programın zamanını nerede harcadıđını gsterir.
- Profil oluřturucu, ađrı grafiđini eřitli formatlarda gsterebilir.
- Teknik detaylar, kullanılan derleyiciye gre farklılık gsterebilir.
- `java -Xprof`



Kod Kapsamı Aracı (Coverage Tool)

- Test çalıştırması sırasında kodun hangi bölümlerinin çalıştırıldığını gösterir.
- Çalıştırılmayan veya kullanılmayan kod parçaları belirlenebilir.
- Test edilmemiş kod bölgelerini bulmaya yardımcı olur.
- Kod kapsamı, yazılımın güvenilirliğini artırmak için kritik bir araçtır.



FindBugs

- Java kodunun statik analizi için kullanılan bir araçtır.
- Kod hatalarını hızlı bir şekilde bulmaya yardımcı olur.



QuickCheck

- Otomatik olarak testler üretir.
- Java dahil birçok dil için sürümleri mevcuttur.



Selenium

- Web uygulamalarını test etmek için bir çerçevedir (framework).
- Tarayıcı tabanlı testleri kolaylaştırır.



Eclipse Memory Analyzer

- Java yığını (heap) analiz ederek bellek sızıntılarını tespit eder.
- Bellek optimizasyonu için kullanılır.



VisualVM

- Bellek tüketimi, iş parçacığı kullanımı ve daha fazlasını görselleştirir.
- Java uygulamalarının performansını izlemek ve analiz etmek için kullanılır.



SON