



**T.C.**  
**SAKARYA ÜNİVERSİTESİ**

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
**YAZILIM TESTİ ÖDEV RAPORU**

**B201210084 - Abdulkadir BAŞTUĞ**

**SAKARYA**

Yazılım Testi Dersi

Dr.Öğr.Üyesi MUHAMMED FATİH ADAK

## Özet

Ödevde github reposunu klonlayıp bu repodaki class içeren .java uzantılı dosyalarda her class için Javadoc yorum satır sayısı, Diğer yorumların satır sayısı, Kod satır sayısı, toplam satır sayısı, fonksiyon sayısı ve yorum sapma yüzdesi hesaplanıp web sitesinde gösterilecek. Bu gösterilen sonuçlar aynı zamanda postgresql veritabanına kaydedilecektir. Ayrıca daha önceden analiz edilip sonuçları veritabanına kaydedilen analiz sonuçları tekrardan web sitesinde gösterilecektir ve bu proje için birim ve entegrasyon testleri yazılacaktır.

## Programda Yapılanlar

Öncelikle Github reposu nu lokal dizine kopyalanmalıydı. Bunu gerçekleştirmek için form yardımıyla kullanıcıdan alınan url `getRuntime.exec` ile terminalde `git` komutu yazdırıldı ve verilen lokal dizine repo kopyalama işlemi gerçekleştirildi. Bu işlemi ilk defa gerçekleştirirken bir probleme rastlanılmadı fakat daha sonra başka bir repo kopyalamak istediğinizde üstüne yazmıyor herhangi bir işlem gerçekleştirmiyordu. Bu yüzden kopyalanacak lokal dizindeki dosyaları silmek için özyinelemeli bir metod yazıldı ve performans düşüklüğüne sebebiyet verdi. Verilen dizindeki dosyaları siliyor ama dizine rastlarsa tekrar kendini çağırıyor ve tüm dosyalar silinince en son klasörü de siliyor. Silme işleminde tekrardan başka bir probleme rastlanıldı. Repo içinde `.git` isminde gizli bir klasör bulunmakta ve sadece read only olduğundan silme işleminde problem çıkarıyordu. Bu yüzden `onerror` fonksiyonu oluşturuldu ve bu hatayla karşılaşınca erişim izinleri değiştirilip silme işlemine sorunsuzca devam edildi. Bu kısım çok zamanımı aldı basit gözükmesine rağmen.

Analiz edilmesi için .java uzantılı dosyaları bulmak için `get_java_files` kullanıldı. Her bulunan dosya tek tek analiz edilmek üzere `analyzeJavaFile` metoduna gönderildi.

Analizde regex kullanmak string sınıfının metotlarını kullanmaktan çok daha kolay olacağından önce regex kullanılarak denendi. Fonksiyon sayısı, regex ifadesi `countRegexMatches` metoduna gönderilerek eşleşme sayısı kadar lokal değişken arttırılıp bulundu. Toplam satır sayısı `\n` karakteriyle string ayrılıp uzunluğundan bulunmuştur. Kod satır sayısı da aynı şekilde ayrılıp her ayrılan string diziye atandı ve her eleman için `trim` metoduyla boşluklar silindi ve `isEmpty` ile boş satırlar anlaşılacak bulundu.

Yorum satır sayısı ve javadoc yorum satır sayısı başta regex ile denendi. Doğru Regex ifadesi yazıldı fakat çoklu satır içinde bir pattern olmadığından sadece regex ifadesinin

eşleşme sayısından çoklu yorum satır sayısı elde edilemiyordu(Tekli yorum satırları hariç). Bu regex ifade başka bir metota gönderilip burada \n ile ayrılıp toplam satır sayısındaki gibi satır sayısı bulunup döndürülebilirdi 2 işi farklı yerde yapmak yerine hem yorum satırlarını bulmayı ve sayısını döndürmek için ayrı bir fonksiyon yazıldı. Stringin endsWith, contains ve startsWith gibi metotların yardımıyla döngü ve koşullar yazılarak toplam yorum satır sayısı ve javadoc yorum satır sayısı bulundu. Yorum sapma yüzdesini hesaplamak için de ayrı bir metot yazılarak dökümanda belirtilen formülden yararlanıldı.

Bunları yaparken oluşturulan fonksiyonlar için birim testleri yazıldı. Birim testlerinde unitest, sahte veri üretmek için faker, parametrelili testler için parameterized ve başka nesnelerin yerine kullanmak için mock kütüphaneleri kullanıldı. Ayrıca karmaşıklığı azaltmak adına print fonksiyonlarının sonuçlarının ekrana basılmaması için StringIO kütüphanesi kullanıldı.

### Eksiklikler

Kod satırını bulurken herhangi bir yazı olması durumunda satır sayısı artıyordu bu da haliyle yorum satırını da kod satırına dahil ediyor. Bunun önüne geçebilmek için toplam kod satırı toplam yorum satırından çıkarıldı ama eğer aynı satır hem kod satırıysa hem de yorum satırıysa kod satır sayısını eksik bulacaktır.

### !Önemli Notlar!

Testlerde veritabanı testleri olduğundan testler çalıştırılırken dikkatli olunmalı çünkü test çalıştırıldığı anda veritabanındaki tüm veriler silinecektir. Çünkü testlerde faker ile rastgele veriler eklenirken entegrasyon testlerinde de veritabanı ile ilgili işlemler gerçekleştirilir ve bu işlemleri temizlemek adına veritabanı sıfırlanır.