



T.C.
SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
PROGRAMLAMA DİLLERİNİN PRENSİPLERİ ÖDEV RAPORU

GIT DEPOSU ANALİZİ

B211210379-Yunus Emre Kara

SAKARYA

Nisan, 2024

Programlama Dillerinin Prensipleri Dersi

Özet

Bu proje, kullanıcıdan alınan bir GitHub depo linki üzerinden Java projesini klonlayan ve analiz eden bir Java programını içerir. Analiz sırasında her bir Java dosyasının içeriği taranır ve javadoc yorum satırı sayısı, diğer yorum satırı sayısı, kod satırı sayısı, LOC (Toplam Satır Sayısı), fonksiyon sayısı, yorum sapma yüzdesi ve sınıf adı gibi özellikler belirlenir. Bu özellikler, projenin sağlığını ve kod kalitesini değerlendirmek için kullanılır. Sonuçlar, kullanıcıya sunularak projenin genel performansı hakkında bilgi verilir.

© 2024 Sakarya Üniversitesi.

Bu rapor benim özgün çalışmamdır. Herhangi bir kopya işleminde sorumluluk bana aittir.

Anahtar Kelimeler: Modülerlik, Analiz, İlişkilendirme, Raporlama

1. Geliştirilen Yazılım

Başlangıçta problemi parçalara ayırdım: kullanıcıdan link alma, linkteki depoyu projeye dahil etme, deponun analizi (bu analiz aşaması kendi içerisinde de modülerite edilmiştir) ve ilgili şekilde ekrana yazdırma.

Kullanıcıdan link almak, linkteki depoyu klonlamak için gerekli sınıf ve metotları araştırma yaparak elde ettim. Deponun analiz aşamasında gereken metotlar ve kurguyu kendim tasarladım. Analiz aşamasında öncelikle ilgili özellikleri tutabilmek ve bu özellikleri tespit edebilmek için Özellikler sınıfını yazdım. Bu sınıf içerisindeki metotlar genellikle dışardan aldığı File türündeki dosyaları analiz eder ve hesaplayıp döndürdüğü değerleri sınıf içerisindeki uygun değişkenlere atar. Analiz sınıfı içerisindeki fonksiyonlarda taranan depodaki her bir class içeren java dosyası için Özellikler sınıfından elde edilen nesnelere gerekli işlemleri uyguladım ve her bir java dosyası için ilgili özelliklerin tutulduğu bir nesne elde edilmiş oldu. Taranan java dosyası kadar nesne üretildi ve bu nesneler bir Özellikler türünde listeye kaydedildi.

Son olarak bu Özellikler türündeki nesne listesi Yazdir sınıfındaki ListeYazdir metoduna parametre olarak verildi ve bu nesne listesinin her bir nesnesi için özellikler uygun şekilde ekrana basıldı.

2. Çıktılar

Arkadaşlarımdan paylaşmış olduğu depo örnekleriyle projemi denedim. Onların çıktılarıyla karşılaştırmalar yaptım. Benzer sonuç elde edilemediği durumlarda problemi çözmek için onlarla iletişim kurdum.

3. Sonuç

Bu proje, modüler bir yapıya sahiptir çünkü işlevler farklı sınıflara ayrılmıştır ve her bir sınıf belirli bir görevi yerine getirir. Örneğin, Analiz

sınıfı depo analizini gerçekleştirirken DepoGetirme sınıfı depo klonlamasını yönetir.

Ozellikler sınıfı, her bir Java dosyasının özelliklerini hesaplar ve saklar, Yazdır sınıfı ise analiz sonuçlarını ekrana yazdırır. Bu yapı, kodun okunabilirliğini ve bakımını kolaylaştırır, çünkü her bir sınıf belirli bir sorumluluğa sahiptir ve kodun parçaları birbirinden bağımsızdır. Bu, projenin genişlemesi veya değiştirilmesi gerektiğinde, sadece ilgili sınıfların güncellenmesini gerektirir, bu da kodun daha modüler ve esnek olmasını sağlar. Hayatta da problemlere bu bakış açısıyla yaklaşp modüllere ayırırsak ve büyük bir sorunla uğraşmak zorunda kalmak yerine daha küçük parçalarla ilgileniriz ve sorunun üstesinden daha rahat gelmiş oluruz.