



**T.C.**  
**SAKARYA ÜNİVERSİTESİ**

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
**PROGRAMLAMA DİLLERİNİN PRENSİPLERİ PROJE RAPORU**

**GİT KLONUNDAKİ JAVA DOSYALARININ ANALİZİ**

**Sinem Şaziye Karaca**  
**G211210025**  
**1 - A**

**SAKARYA**  
**Nisan, 2024**  
Programa Dillerinin Prensipleri

## GİT KLONUNDAKİ JAVA DOSYALARININ ANALİZİ

Java programlama dili kullanılarak hazırlanan ilk ödevde, kullanıcıdan alınan git repo linkinin kullanıcı bilgisayarına klonlanması ve bu klon klasör üzerinde isterler dikkate alınarak analiz yapılması hedeflenmiştir. Klon klasörün içinde bulunan klasör ve dosyalar incelenerek içerisinde yalnızca class yapısı bulunan, enum veya interface gibi yapıları içermeyen Java dosyaları alınarak bu dosyaların içerisindeki Javadoc yorum sayısı, yorum satırı sayısı, boşluk ve yorum satırları haricindeki kod satır sayısı, toplam kod sayısı ve fonksiyon sayısı tespit edilmektedir. Ardından bu değerler kullanılarak yorum sapma yüzdesi hesaplanmakta ve tüm bu değerler kullanıcı terminaline çıktı olarak sunulmaktadır.

Proje boyunca en çok kullanılan Java sınıfı File sınıfıdır. Bu sınıf özellikle git repo URL'sinin klonlanmasında ve klonlana klasörden isterlere uygun dosyaların seçilmesi ve analiz edilmesi konusunda kullanılmıştır. Bu sınıf ayrıca dosya uzantılarının Java uzantısı olup olmadığını kontrol etmede ve analiz aşamasında dosya içeriklerinin incelenmesinde kullanılmıştır.

Projenin ilk aşaması kullanıcıdan alınan git repo URL'sinin klonlanması işlemi gerçekleştirilmekte. Bunun için GitSlemleri isimli, bir adet propertie ve cleanDirectory ile clonedRepository isimli iki adet fonksiyondan oluşan sınıf oluşturuldu. Fonksiyonlardan ilki dizinin dolu olması halinde dizinin içindeki tüm klasör ve dosyaları temizlerken diğer fonksiyon boş dizine git reposunu klonlamakla görevli. Test sınıfından GitSlemleri sınıfının fonksiyonlarının çağrılmasının ardından alınan dosya yolu JavaDosyalarıBul sınıfının javaFiles isimli fonksiyonuna parametre olarak gönderilir ve bu fonksiyonun içinde tanımlanan string ile dizindeki dosyaların isimlerinin uzantı kısmı tek tek karşılaştırılır. Herhangi bir eşleşme olması halinde dosyanın Java dosyası olduğu kanıtlanmış olur. Dizinde kontrol edilen dosya normal dosya değil yine bir çeşit dosya olan klasörse recursive bir şekilde javaFiles fonksiyonu bu klasör için tekrar çağrılır ve bu klasör içindeki dosyalar da Java dosyası olup olmamasına göre kontrol edilir. Dizinin kontrol işlemi dosya dosya gerçekleşeceğinden bir dosyanın Java dosyası olması halinde içinde sadece Class barındırıp barındırmadığının kontrolü için Java uzantılı dosyaların dosya yolu SadeceClassCerenler sınıfının dosyalCeriği fonksiyonuna parametre olarak gönderilir. Bahsedilen kontrolü yapmak için bu sınıfta dosyanın içeriğinde enum veya interface yapılarının olup olmadığı kontrol edilir. Bu yapıları içeren dosyalar Java dosyası olmalarına rağmen isterleri karşılamadığından bir sonraki aşama olan analiz aşamasına geçemezler. İçinde yalnızca Class bulunan dosyalar analiz edilmek üzere DosyaAnaliz sınıfına gönderilirler. Burada regex kullanılarak dosya içerisindeki fonksiyon sayısı; dosya işlemleri ve if yapıları kullanılarak dosyadaki Javadoc, yorum, kod ve toplam kod satır sayıları tespit edilir. Elde edilen bu değerler çeşitli işlemlere tabi tutularak yorum sapma yüzdesinin hesaplanmasında kullanılır. Tüm bu işlemlerin ardından değerler tüm Java dosyaları için ayrı olacak şekilde dosya isimleriyle beraber ekrana basılır.

Projenin geliştirme aşamasında en çok zorlandığım kısım kullanıcıdan git repo URL'sini alarak bu repoyu klonlamak oldu. Daha önce geliştirdiğim projelerde gerçekleştirmediğim, benim için yeni bir işlemdi. Fakat yaptığım araştırmalar hem git teknolojisinin ne olduğunu ve ne işe yaradığını gördüm hem de bu yapının kullanıcı bilgisayarında nasıl kullanılabileceğini ve klonlama aşamasında hangi süreçlerin gerçekleştiğini öğrendim. Bu aşamada karşılaştığım en büyük problem kod gerçekleştirmesinin ardından yürüttüğüm test aşamasında reponun klonlandığı dizinin hiçbir zaman temizlenmemesi ve her yeni reponun, dizine eklenen ilk repodan dolayı çalışmaması ve çıktıda sürekli ilk reponun değerlerini almam oldu. Kısa bir araştırma sonucu bu sorunun dizinin hiçbir aşamada temizlenmemesinden kaynaklandığını, bir if else yapısıyla bu sorunu kolayca çözebileceğimi gördüm. Gerçekten de sorunu gidermek için izlediğim adımların ardından projeyi tekrar çalıştırdığımda öncelikle mevcut dizin doluydu temizlendi, ardından yeni repo dizine sorunsuz bir şekilde klonlanarak gerekli işlemlere tabi tutuldu. Zorlandığım aşamalardan bir diğeri regular expression yapısını kullanmak oldu.