T.C.

SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

PROGRAMLAMA DİLLERİNİN PRENSİPLERİ ÖDEV RAPORU

ÖDEV 1

B201210024 – Ahmet Furkan SÖĞÜTCÜ

SAKARYA

Nisan, 2024

Programlama Dillerinin Prensipleri Dersi

PDP 1.ÖDEV

Ahmet Furkan SÖĞÜTCÜ

B201210024 – 1/C

Özet

Bu dersin ödevinde bizden yapılması istenilen işler; program çalıştığı zaman konsolda girilecek olan github repository linki ile repository dosya klasör dizinine klonlanacak ve klonlanan repository içinde bulunan dosyalar arasından da .java uzantılı dosyalar seçilecekti. Ardından bu seçilen dosyalar arasından da sadece sınıf olanları bir şekilde ayıklanacak ve ayıklanan dosyalar da çeşitli analizlere tabi tutulacaktı. Bu analizler sırası ile kod içerisindeki javadoc yorum satırlarının sayısını bulma, geri kalan yorum satırlarının sayısını bulma, kod satır sayısını bulma, dosyadaki tüm satır sayısını bulma, fonksiyon sayılarını bulma ve son olarak da yorum sapma yüzdesini hesaplama gibi işlemlerdi. Bu işlemler sonucunda da tüm değerlerin sırası ile ekrana yazdırılması gerekiyordu.

Ben bu ödevi yaparken ilk olarak github üzerinden repository klonlama işini gerçekleştirmeye çalıştım. Bunu yaparken ilk başta jGit kütüphanesi ile yapabileceğimi düşünüyordum lakin hem ekstra kütüphane indirmek olsun hem de çeşitli hatalarla karşılaşmam olsun bu gibi sıkıntılar yüzünden farklı arayışlara girdim ve processBuilder ile karşılaştım[1]. Bu yöntem hem daha basitti hem de kullanışlıydı. Git komutlarını processBuilder sayesinde verdikten sonra konsol üzerinden alacağım repository linkini aynı dosya dizinine kopyalama işlemlerimi yaptım. Bunları da file.java dosyamda yaparak kodda sadeliği sağlamaya çalıştım. Bundan sonra da yalnız sınıf dosyalarını seçip sırası ile dosya okuma yapıp regex işlemlerimi yapmak kalıyordu geriye. Ben de yeni bir regex.java adında dosya oluşturup orada da sırayla okunacak dosyalar üzerinde bizden istenilen değerlerin regex ile tespit edilip ekrana yazdırılması vardı.

Anahtar Kelimeler: Regex, Java, ProcessBuilder

1. GELİŞTİRİLEN YAZILIM

1.1. Dosya İşlemleri

Ödevi yapmaya önce repository klonlama ile başladım. Tabi öncelikle klonlama işleminin yapılacağı dosya dizinini belirtmem lazımdı ve bunun için de kendim java kodu ile clone isimli bir klasör oluşturdum. Böylelikle klonlanacak dosya oraya aktarılacaktı. Birden fazla kez programın çalıştırılmasına karşın da işlemler bittikten sonra dosyayı silerek klasörün temizlenmesini sağlıyorum. Klonlama işlemi için de bir süre araştırdıktan sonra processBuilder sınıfını buldum ve git komutlarını kullanabilmem için çok yararlı bir ortam sağladı bana. Bu işlemi dosya.java dosyasında cloneRepository fonksiyonu altında processBuilder'ın command metodunda parametre olarak git komutlarını sırası ile girilmesini sağlayarak klonlama işlemini yaptım. Ardından klonlanan dosyada gezerek .java uzantılı ve sadece sınıf olanların alındığı bir fonksiyon yazdım. Alınan dosyaları bir listeye koydum ki tekrardan regex ile analiz yaparken kolaylıkla ulaşılabilir olsunlar.

Ödev Sorumlusu: Ahmet Furkan SÖĞÜTCÜ, B201210024
Mail Adresi: ahmet.sogutcu@ogr.sakarya.edu.tr

1.2. Regex

Dosya işlemleri bittikten sonra bulunacak değerler için sırası ile regex'lerini tasarlamam gerekiyordu. Onun için de önce javadoc satır sayılarını hesaplamaktan başlayarak istenilen analizleri yaptım. Yaparken çeşitli regex sitelerinden ve videolardan da yardım aldım[2]. Regex.java dosya düzenine gelecek olursam da öncelikle dosya listesinden teker teker dosyalar okunduktan sonra her dosya için dosya analizlerinin yapıldığı fonksiyon çağırılıyor. Çağırılan fonksiyonda da bufferedReader ile dosya okunduktan sonra satır satır gezilerek tasarlanan regex kodları ile bulunması gerekilen satır sayıları hesaplanıyor. Son olarak da ekrana hesaplanan değerler yazdırılıyor.

Bulunacaklar	Regex Kodları
Javadoc	\/**[^*]**+(?:[^/*][^*]**+)*\/
Yorum satırı	(\/\/[^\n\r]+?(?:*\)|[\n\r]))|(\/*(?!*)(?:[^*]|*(?!\/))**\/)
Kod satırı	^(?!\s*$)(?!\s*\/\/)(?!\s*\/*)(?!\s**\/)(?!\s**).+
Fonksiyon	\b(public|protected|private)?\s+(\w+\s+)?\w+\s*\([^)]*\)\s*(throws\s+[\w\,]+\s*)?\{+\{?

Tablo 1. Tasarlanan Regex Kodları

2. ÇIKTILAR

Hocamızın örnek olarak bize verdiği github dosyasını deneyerek hocamızın alığı çıktıların aynısını almayı başarabildim. Farklı olarak da github üzerinden birkaç dosyayı daha deneme fırsatım oldu ve onları da deneyerek ödevin doğruluğunu biraz daha arttırmaya çalıştım. Çıktılar yine doğruydu bu şekilde de dosya işlemlerimin regex kodlarımın doğru şekilde çalıştığını tekrar tekrar görmüş oldum. Tabi bu noktaya gelmek için çok fazla regex kodları deneyerek deneme yanılma yolları ile test ederek en optimum regex kodunu oluşturdum. Javadoc ile uzun yorum satırını ayırmak örneğin biraz uğraştırdı.

3. SONUÇ

Bu ödev çalışmasının sonucunda istenilen ifadeler başarılı şekilde ve bizden talep edilen formda çıktı olarak belirtildi. Bu sayede bu program sayesinde çeşitli yorum satırları ve kod dosyası hakkındaki bazı bilgiler çıktı olarak doğru şekilde ekrana verilmektedir

Referanslar

[1] https://docs.oracle.com/javase/8/docs/api/java/lang/ProcessBuilder.html

[2] https://regexr.com