

05 NİSAN 2024

PROGRAMLAMA DİLLERİNİN PRENSİPLERİ

1. ÖDEV

SUDE ÖZKANOĞLU

G201210034

sude.ozkanoglu@ogr.sakarya.edu.tr

RAPOR

Ödev çalışmasında ilk olarak gönderilen ödev raporunu inceleyerek çalışmalara başladım. Eclipse ortamını iyi bir şekilde yaratabilmek için ödev dosyasını incelerken notlar çıkardım. Çıkardığım notlar için ilk olarak ödev isterlerini yazdım ardından kendimce önemli ve dikkat etmem gereken noktaları not aldım. Bu işlemleri gerçekleştirdikten sonra ilk düşündüğüm kısım özellikle farklı klasörler içerisinde gezinerek “.java” uzantılı dosyaları nasıl bulacağım oldu. GitHub Repo’larının “JGİT” adlı bir kütüphane ile kullanılabileceğini daha önceden biliyordum fakat, kullanılacak olan örnek ödev dosyasının içerisinde nasıl gezineceğimi öğrenmek en çok zamanımı alan noktalardan biriydi.

İlk olarak bu noktada neler yapabileceğimi öğrenmeye çalıştım, Java içerisinde dosya işlemleri ile alakalı bir kütüphane olduğunu bildiğim için, ilk görev olarak bu kütüphanenin dokümantasyonunu ve kodlarını inceleyerek başladım. Bu kısım için gerekli notlarımı aldıktan sonra Java projemi oluşturarak çalışmalara başladım. İlk olarak ders müfredatında da öğrendiğimiz Regex yapılarını başarılı bir şekilde kullanabilmek için kullanıcıdan girilecek olan “GitHub Repository URL” adresini kontrol etmek istedim.

Main dosyam içerisinde bir try-catch yapısı kurarak kullanıcıdan “GitHub Repository” adresini aldım. Ardından kendi yazdığım Regex ifadesi ile birlikte girilen adresi karşılaştırdım. Adres kontrolü başarılı bir şekilde sonuçlanırsa “JGİT” üzerinden kütüphane elemanlarını kullanabilmek adına farklı bir sınıf daha oluşturarak, kullanıcının girdiği link üzerinden gerekli parametre gönderme işlemlerini yaptım. Klonlanacak dizin ve parametre bilgileri ile birlikte “Fetch.java” dosyasında klonlama işlemlerine başladım. Oluşturduğum “cloneRepository()” methodu ile birlikte “JGİT” yapılarını kullanarak klonlama işlemini başarılı bir şekilde gerçekleştirdim.

Klonlama işlemi sonrası başarılı bir şekilde klonlanan projenin dizinini kullanabilmek ve sınıf işlemlerini gerçekleştirmek adına “ProcessClasses.java” isimli bir sınıf daha oluşturdum. Bu sınıf içerisinde belirli bir sınıf ile alakalı tüm kontrollerin yapıldığı yapıyı oluşturmayı hedefledim. Proje dahilinde ilk iki yapı olan “Main.java” ve “Fetch.java” kısmında neredeyse zorlanmadım diyebilirim, fakat en çok zorlandığım kısım kesinlikle tüm kontrollerin başarılı bir şekilde yapılmasının gerekli olduğu “ProcessClasses.java” yapısı oldu. İlk olarak derste öğrendiğim tekniklerle “kapsülleme” yapısını kurmaya çalıştım ve değişkenleri “private”

olarak tanımlayarak, dışarıdan müdahaleye izin vermedim. Ardından ilk olarak klonladığım dizin üzerinden dosyaları okuyabilmek adına, getAllJavaFiles(File path) metodunu tanımladım. Bu method ile birlikte, proje dizini içerisinde gezerek, tüm “.java” uzantılı dosyaları işleme soktum. Gerekli ödev kısıtlarına uyan tüm sınıfları bu method üzerinden, readClass(allFiles[i].getAbsolutePath(), baseSubstring) methodu ile yönlendirdim. Bu yönlendirme ile birlikte hem dosyanın dizinine, hem de baseSubstring değişkeni ile birlikte sınıfın adına sahip oldum. Burada en çok zorlandığım kısım dosyaların ödev isterleri ile uyuşması kontrolünü sağlamaktı. Gerekli if-else ve try-catch yapıları ile birlikte buradaki hataları da başarılı bir şekilde atlattım.

readClass() method içerisinde gerekli işlemleri yapabilmek adına methodRegex, yapıcıMethodRegex ve classRegex ifadeleri tanımladım. Bu tanımlamalar ile birlikte başarılı bir şekilde kontrollerimi gerçekleştirebilmek adına Regex ifadeler ile ilgili büyük bir yol kat ettiğimi düşünüyorum. Bu aşama için ödevin en zorlayıcı kısmı kesinlikle bu Regex ifadelerin yazılması kısmı oldu. Özellikle, metotlar için yazılacak olan Regex beni çok zorladı. Bu tanımlamaları başarılı bir şekilde yaptıktan sonra, elime ulaşan tüm sınıfların tüm satırlarını okuyarak checkLines(pattern, patternConstructor, line) metoduna yolladım. Bu metot içerisinde ödev isterleri içerisinde bulunan tüm yapıları if-else yapıları ile kontrol ettim. Ödev isterlerini eksiksiz bir şekilde yerine getirmek için en çok zorlandığım diğer bir kısım ise kesinlikle bu metot içerisinde yazmaktı. Bu metot içerisinde en çok zorlandığım ödev isteri ise, javaDoc ve çoklu yorum satırlarını hesaplamaktı. Ödev isterinde özellikle başlangıç ve bitiş ifadelerinin arasındaki yorumlar istendiği için buradaki kontrolü boolean değerler ve sayac değişkenleri ile kontrol etmekte epey zorlandım. Satırları doğru okuyabilmek adına .trim(), .startsWith() ifadelerini kullanmayı öğrenmek benim için çok büyük kazanımlardan biriydi. Özellikle sınıfların metotlarını bu şekilde kullanmayı öğrenebilmek benim için çok değerliydi.

Buradaki işlemleri de başarılı bir şekilde gerçekleştirdikten sonra ödev isteri içerisindeki gibi yorum yüzdesi hesabı yapan metodumu yazdım ve bu metodu printResult() içerisinde çağırdım. printResult() metodu benim ödev boyunca hesapladığım tüm değerleri ekrana yazdığım metot olarak tasarladım ve örnek ödev çıktısındaki ekran görüntüsünün aynısını yaptım. Ekrana yazdırma kısmında büyük bir zorlukla karşılaşsam da yüzde hesabı noktasında işlem önceliklerini dikkate almadığım için uzun bir süremi kod içerisinde hata aramakla geçirdim. Bu kısım da başarılı bir şekilde tamamladıktan sonra genel ödev dizinimde, kullanıcının girdisini kontrol eden ve gerekli yönlendirmeleri yapan “Main.java” sınıfı, klonlama işlemi için

“Fetch.java” sınıfı ve sınıfları gerekli isterler noktasında kontrol edebilmek için de “ProcessClasses.java” sınıfını yarattım. Gereksiz olabilecek, okunaklığı azaltabilecek tüm kod bloklarını temizleyip, en temiz kodu yazmaya çalıştım. Bu noktada birçok kod prensibi öğrendim ve Java üzerinde, metotlar, kütüphaneler ve hata kontrolleri noktasında birçok değerli bilgi edindim.