



T.C.
SAKARYA ÜNİVERSİTESİ

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

GİT REPOSU ANALİZİ

SAKARYA

Nisan, 2024

Programlama Dillerinin Prensipleri Dersi

GİT REPOSU ANALİZİ

FATİH UÇAR

^a G211210038 -- 2B

GELİŞTİRİLEN YAZILIM

Bu ödevde ilk olarak git reposunun klonlaması için gerekli jar dosyalarını araştırdım. Gerekli jar dosyalarını bulduktan nasıl kullanılması gerektiğini öğrendim. Gerek jar dosyasının eklenmesi gerek dosyanın nasıl indireleceği hakkında bilgi sahibi oldum sonrasında nasıl bir sınıf oluşturmam gerek onu düşündüm. Sınıfın yöntemleri olsun sınıftan oluşturulan nesnelerin nasıl çağırılması gerektiği gibi şeyleri düşündükten sonra kodu yazmaya başladım.

İlk olarak bir github_repo_analyzer sınıfı yaptım bu sınıfın constructorına klonlanacak dosyanın repo linkini ve reponun nereye klonlaması gerek onlarla ilgili parametreleri yazdım. Reponun klonlaması sırasında sadece .java uzantılı dosyaları indirdim ve analiz için sadece class içeren java dosyalarını ayıkladım.

Sonrasında javadoc yorum satırlarının nasıl bulunması gerektiği ile ilgili bir private yöntem yazdım. Bu yöntemde javadoc yorum satırlarının başlangıcını (/**) ve bitişini (*/) dahil etmedim. Başlangıç ve bitiş arasındaki satır sayısını aldım ve bazı ek kontroller yaptım örneğin javadoc yorum bloğunun içinde satırın başında yıldız olmazsa onu da javadoc yorum satırı sayılsın ayrıca javadoc'un başlangıç ve bitiş bir string içinde kullanıldıysa onun javadoc yorum satırı sayılmaması için gerekli kontrolleri yaptım. Sonrasında yorum satırı sayısına geçtim. İlk olarak düz yorum satırı (//) sayısını bulan bir yöntem yaptım ve string içinde mi değil mi yine gerekli kontrolleri yaptım. Bir diğer yorum türü olan yorum bloklarına geçtim (/* */) ve javadoc'ta olduğu gibi başlangıç ve bitiş almadım ve string kontrolü yaptım ayrıca farklı formlar içinde eklemeler yaptım örneğin yorum satırı (/* ornek yorum satırı */) şeklinde ise 1 satır yorum satırı sayılması gerektiğini yöntemime ekledim. Bu formu javadoc'a da ekledim sonrasında. Bundan sonra Kod satır sayısını bulan yöntemime geçtim. Bu yöntemde boş satırları, javadocları ve yorum satırlarını dahil etmedim. Sıra LOC yöntemine geldi ve .java uzantılı dosyayı tarattım ve satır sayısını döndürdüm. Sonrasında Fonksiyon bulma yöntemine geçtim. Fonksiyon bulma yönteminde ilk olarak parantezlerin yerine ve şekline dikkat etmedim abstract methodlar vs.. onları da bu yöntemin içine ekledim. Yorum sapma yüzdesi fonksiyonunda ise bu int değerleri döndürdüm sonrasında veri kaybı olmaması için double veri türüne casting yaptım. Sonrasında yaptığım bütün yöntemleri analyze yöntemi içinde topladım. Bu yöntemi nesneden erişebilmek için public yaptım. Sonrasında bir example main açtım. Klonlanacak repoyu istedim ve gerekli analizleri yaptırdım.

Bu ödevde en çok zorlandığım yer javadoc yorum satırlarının başlangıç ve bitişini almamak ve farklı formlarını yazmakta zorlandım.

Ödev bana regex ve java hakkında birçok bilgi kattı ve string yöntemlerini geniş çaplı öğrenmemi sağladı.