



T.C.
SAKARYA ÜNİVERSİTESİ

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

YORUM SATIRI BULUCU

B201210098 – İlknur KAYA

SAKARYA

Nisan, 2023

Programlama Dillerinin Prensipleri Dersi

YORUM SATIRI BULUCU

İlknur KAYA^{a*}

B201210098 – 1A

Özet

Ödevde istenilen bir java dosyasının adının konsol parametresi olarak alınması ile başlamaktadır. Dosyanın içerisinde sınıfları bulup bu sınıfların içerisinde yer alan her bir fonksiyonlara ait tek satırlı, çok satırlı ve javadoc yorum satırlarının bulunması istenmektedir. Fonksiyon başlamadan önce üzerinde bulunan javadoc yorumlarının fonksiyonun yorumları içerisinde eklenmesi gerekmektedir. Toplanan yorumlar her biri için ayrı bir .txt uzantılı dosya oluşturup içine yazılması ve konsol ekranında sınıf adı, fonksiyonlar ve hangi yorumdan kaç adet yoruma sahip olduklarının yazılması ile son bulur. Bu proje için regex yaklaşımından yararlandım. Regex tanımları ile bulunan yorumları tutabilmek için ArrayList kullandım. Böylece bana dosya yazma işleminde kolaylık sağlamış oldu. Fonksiyonların bitişlerini hesaplamak için sayaç kullandım. Bunun sebebi fonksiyonların içerisinde yer alan koşul veya döngü yapıları gibi regex tanımlamasında karışabilecek durumların önüne geçtim. Dosyaya yazma işlemleri için ayrı bir sınıf kullandım.

© 2023 Sakarya Üniversitesi.

Bu rapor benim özgün çalışmamdır. Faydalanmış olduğum kaynakları içerisinde belirttim. Her hangi bir kopya işleminde sorumluluk bana aittir.

Anahtar Kelimeler: Regex, ArrayList, Dosya Okuma-Yazma, Yorum Satırı, Fonksiyon, Sınıf

1. GELİŞTİRİLEN YAZILIM

Hesap sınıfı içerisinde sınıf, fonksiyon, yorum hesaplamalarını yaptım. Öncelikle yorumlar için ayrı ayrı ArrayListler oluşturdum. Daha sonrasında Regex ifadeleri tanımladım. “hesap” fonksiyonu ile işlemlere başladım. Konsol parametresi olarak gelen ifade ile dosya okumayı başlattım. Yorumlar için eğer .txt uzantılı dosya yoksa oluşturmasını yaptım. Satır satır okuma işlemine başladım. Sınıf, fonksiyon ve javadoc Matcherlarını yazdım. Sınıfın nerede bittiği bu ödevde çok önemli olmadığı için sınıf bulunduğu zaman direk ekrana yazdırdım ve başka bir işlem yaptırmadım. Fonksiyonun üzerindeki javadocları bulmak için javadoc bulan döngüyü yazdım. Bu döngü satır sonu “*/” ile eşleşene kadar dönüp farklı bir string değişkene ekliyor ve satır sonu eşleşirse döngüden çıkıp son satırı da değişkene ekliyor. En son olarak bulunan yorum ArrayListe ekleniyor. Fonksiyon bulan döngüye girildiğinde ekrana fonksiyonun adını yazdırdım ve fonksiyona ait parantezleri sayan sayacfunc arttırdım. Eğer satır “{” içeriyorsa sayaç bir arttırlıp “}” içeriyorsa bir azaltıldı. Bu durumda dizi tanımlamaları gibi bir durumda doğru hesaplayabilmesi için iki koşul da if ile tanımlandı. Eğer sayacfunc sıfır olursa fonksiyon bittiği için konsola yazdırma ve .txt dosyalarına yazdırma işlemleri burada yapıldı. Bu işlemlerin ardından yeni bir fonksiyon hesabı için ArrayListler temizlendi. Tekli, çok ve javadoc yorumları içindöngüler ve koşulları yazdım. Fonksiyon bulan döngünün sonuna geldiğimde farklı bir işlem gerekmediği için satır okumadan sonra dosya okumayı kapattım. Dosya sınıfında .txt ler bulunmuyorsa açması için bir fonksiyon ve ArrayListlerden değerleri yazdırması için ayrı bir fonksiyon yazdım. Yazdırma fonksiyonunda dosyayı ekleme modunda açtım.

* Ödev Sorumlusu. İlknur KAYA B201210098

Mail Adresi: ilknur.kaya3@ogr.sakarya.edu.tr

2. ÇIKTILAR

Çok satırlı yorumların içerisinde yer alan “/” işareti tek satırlı yorum olarak algılanmıyor. Aynı durum Javadoc yorumları için de geçerli çünkü döngü Javadoc ve çok satırlı yorumlar için dönüyor içerisindeki yazan her şey yorum ifadesi olarak alınıyor. Fokskiyon tanımlandıktan sonra “{” işaretinin ardından gelen tek satırlık yorumlar algılanmıyor. Aynı şekilde fonksiyon bitiminde bulunan “}” işaretinin ardında ya da önünde yer alant ek satırlık yorumlar algılanmıyor. Bunun sebebi o işaretler ile işlem yaptıktan sonra satır atlanması ya da sayaçın sıfır olmasından dolayı döngüden çıkılması koşullarıdır. Javadoc yorumları ve fonksiyon arasında başka bir ifade yer alsada dahi o Javadoc yorumu fonksiyonun yorumu olarak alınıyor. Yapılan denemeler sonucunda konsolda çıkan sayıların ve .txt dosyalarının içeriğinin doğruluğu tespit edilmiştir.

3. SONUÇ

Bu ödevin sonucu olarak Regex ifadelerinin kullanımını daha iyi kavradım. Öncesinde yazmakta zorlansam ve yazarken kafam karışsa bile syntax larını anladım. Tek tek ifadeler ile iç içe ifler kullanmak yerine daha kolay ve basit bir yol olduğunu öğrendim.

Kaynakça

- [1] Kerteriz. “Java Dosya Açma, Okuma, Yazma ve Düzenleme İşlemleri”. Erişim:14.04.2023. [Java Dosya Açma, Okuma, Yazma ve Düzenleme İşlemleri \(kerteriz.net\)](http://kerteriz.net)
- [2] Computer Hope. “Regular Expressions Quick Reference”. Erişim: 17.04.2023. [Regular Expressions Quick Reference \(computerhope.com\)](http://computerhope.com)
- [3] regex101. Erişim:18.04.2023. [regex101: build, test, and debug regex](http://regex101.com)