



T.C.
SAKARYA ÜNİVERSİTESİ

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

Java Parser

B181210076 - Muhammet Kemal Güvenç

SAKARYA

Nisan, 2023

Programlama Dillerinin Prensipleri Dersi

Java Parser

Muhammet Kemal Güvenç

B181210076 – I/C

Özet

Program, aldığı dosyadaki metotları ve içlerindeki yorumları algılar. Sonra bunlarla ilgili bilgileri ekrana çıkartır. Ayrıca elde ettiği yorumları dosyalara kaydeder. Bunu şu şekilde yapar: İlk önce parametre olarak dosyanın adını alır. Dosyayı adından bulup tamamını okur. Okunan dosyadaki yorum satırları geçici olarak silinir. Sonra içindeki metotların “header” kısımlarını bulur. Ardından metotların gövde kısımları bulunur. Eğer üzerlerinde Javadoc yorum satırları varsa onlar da dahil edilir. Böylece dosyadaki metotların hepsi bulunmuş ve ayıklanmış olur. Devamında yorumları geri koyar. Ayrıca hangi yorumun hangi metoda ait olduğu anlaşılabilmesi için de yorumlar geri koyulurken metotlarda buna uygun şekilde yeniden ayarlanır. Her bir metot için elde ettiği bilgileri ekrana çıkartır. Ayrıca yorumları da dosyalara kaydeder.

© 2023 Sakarya Üniversitesi.

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

Anahtar Kelimeler: Java, Parser, Metot, Yorum Satırı

1. GELİŞTİRİLEN YAZILIM

1. Parser Sınıfı: Bu sınıfta yapıcı metot dışında 8 adet metot bulunmaktadır. Bunlar şu şekilde çalışmaktadır:

- a. public Parser(String text): Parser sınıfının yapıcı metodudur. Bu metot parametre olarak ayıklanacak (parse) metni alır.
- b. private void clearComments(): Bu metot ile ayıklanacak (parse) metin bütün yorum satırları Regex kullanılarak geçici olmak üzere ayıklanır.
- c. private void getMethodHeaders(): Yorum satırlarında arınmış olan metin üzerinden metotların baş (header) kısımlarını Regex ile bulur. Bulunan metotların metin üzerindeki başlangıç ve bitiş konumları LimitsOfMethod sınıfı ile saklanır.
- d. private void extendMethodsToBodies(): Şimdiye metotların sadece baş (header) kısımları elde edilmiş oldu. Bunların gövde kısımlarının da alınması gerekir. Bunun için şöyle bir algoritma çalışır.
 - İlk önce bir sayaç oluşturulur.
 - Metodun baş (header) kısımlarından sonra gelen ilk “{“ parantezi bulunur. Ve sayaç 1 arttırılır.
 - Bu şekilde her “{“ parantezi görüldüğünde sayaç 1 arttırılır her “}” parantezi görüldüğünde 1 azaltılır.
 - Sayaç 0 olduğunda metodun sonuna geldiği anlaşılır ve metodun bitiş konumu güncellenir. Bu süreç her fonksiyon için tekrar edilir.
- e. private void fixComment(): Geçici olarak silinen bütün yorum satırlarını geri yerine koyar.
- f. private void fixMethodLimitsAccordingToComments(): Yorumlar eklendikten sonra metotların başlangıç ve bitiş konumları kaymış olma problemi oluşur. Bunu düzeltmek için bütün metotların başlangıç ve bitiş konumları yeniden ayarlanır. Şöyle ki,
 - Eğer bir yorumun başlangıç konumu bir metodun başlangıç konumundan önceyse metodun her iki konumu da ileri alınır.
 - Fakat yorumun başlangıç konumu metodun başlangıç konumundan ileri ama bitiş konumundan önceyse sadece bitiş konumu ileri alınır. Bu süreç her yorum ve her fonksiyon için kontrol edilerek yapılır.
- g. private void extendMethodsToJavadoc(): Metotların üzerindeki Javadoc yorumlarını da metodun içine alabilmek için her metodun üstü kontrol edilir. Eğer metodun üstünde bir Javadoc yorumu tespit edilirse metodun başlangıç konumu geriye alınır.
- h. public void parse(): Yukarıda anlatılan bütün metotları sırasıyla çağırıp metni ayıklayar. (parse)

- i. public List<Method> getMethods(): Bütün metin ayıklandıktan (parse) sonra elde edilen bütün metotların başlangıç ve bitiş konumları “LimitsOfMethod” nesneleri ile saklanır. Fakat metotlar üzerindeki yorumları ayıklayabilmek için metotlar “Method” nesnelerinde saklanmalıdır. Bütün bulunan metotları “Method” nesnesi olarak geri döndürür.
2. LimitsOfMethod Sınıfı: Sadece bulunan metotların metin üzerindeki başlangıç ve bitiş konumlarını saklar.
3. Method Sınıfı: Bu sınıfta yapıcı metot dışında 8 adet metot bulunmaktadır. Bunlar şu şekilde çalışmaktadır:
 - a. public Method(String text): Parametre olarak yazılı metodu alır. Sınıftaki diğer metotları kullanarak verilen metottaki yorumları bulup değişkenlerde saklar.
 - b. public void show(): Metodun içindeki yorum satırları ile alakalı bilgileri ekrana çıkartır.
 - c. public String getContentOfSingleLineComments(): Tek satırlı yorum satırlarını döndürür.
 - d. public String getContentOfMultiLineComments(): Çok satırlı yorum satırlarını döndürür.
 - e. public String getContentOfJavadocComments(): Javadoc yorum satırlarını döndürür.
 - f. public String getMethodName(): Metot isimini döndürür.
 - g. private String createContent(List<String> comments): Metodun içindeki yorum satırları ilk önce bir listede saklanır. Bunlar sonra tek bir String’e çevirmek için bu fonksiyon çağrılır.
 - h. private String findMethodName(): Metodun ismini Regex ile bulur.
 - i. private void findComments(...): Metodun içindeki yorum satırlarını Regex ile bulur. Bulduğu yorum satırları hangi tür yorum satırına aitse ona ait değişkene kaydeder.
4. MethodFileWriter: Yapıcı metot dışında 1 tane fonksiyonu vardır.
 - a. public MethodFileWriter(): Kaydedilecek dosyaları hazırlar.
 - b. public void writeMethodToFiles(Method method): Kendisine parametre olarak verilen metodun içinden yorum satırlarını alır ve uygun olan dosyaya yazar.

2. ÇIKTILAR

Programaya verilen herhangi bir dosyada bulunan bütün metotları bulur. Onlara ait yorum satırlarının bilgisini ekrana çıkartır ve yorum satırlarını dosyalara kaydeder.

3. SONUÇ

Bu program ile bir Java dosyasında bulunan metotları hızlıca görülebilir ve bu metotlar ile alakalı bütün yorum satırları görülebilir.