**1fd75c3872a94e26ad68c7fa7667bdc82c07dd4ba85f4a6793f7a2b4e943b8fd**

**T.C.**

**SAKARYA ÜNİVERSİTESİ**

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

PROGRAMLAMA DİLLERİNİN PRENSİPLERİ ÖDEV RAPORU

**Java’da Operatör Sayımı**

**B181210076 - Muhammet Kemal Güvenç**

**SAKARYA**

**Mart, 2021**

Programlama Dillerinin Prensipleri Dersi

**Java’da Operatör Sayımı**

Muhammet Kemal Güvenç

B181210076 – B Grubu

Özet

Program, kendisine dosya yolu verilmiş Java kaynak kodu dosyasını karakter karakter okuyup içindeki operatörleri sayar ve operatörler hakkında bilgi verir. İlk önce dosya komple okunup bir String’e kaydedilir. Sonra kaydedilen String’deki tek satır ve çoklu satır yorumları, char’lar ve String’ler temizlenir. Ayrıca boşlukları ( ,\t), satır sonu karakterlerini (\r) ve satır atlama karakterlerini (\n) siler. Şablonlarda bulunan “<” ve “>” işaretleri operatör olarak sayılmaması için temizlenmiş olan String’deki şablon yapısına sahip sınıfların isimlerini kaydedilir. Sonra bu isimlere bakarak şablonlar silinir. Daha sonra sınıfların üye değişkenleri ve metotların gövdeleri hariç her şey silinir. En sonda ise operatörler sayılır. Ardından ekranda sayılan operatörler hakkında bilgi gösterilir.

© 2017 Sakarya Üniversitesi.

Bu rapor benim özgün çalışmamdır. Faydalanmış olduğum kaynakları içeresinde belirttim. Herhangi bir kopya işleminde sorumluluk bana aittir.

Anahtar Kelimeler: Operatör, Java, String, Lexical

1. Dosyanın Okunması ve Gereksiz Şeylerin Temizlenmesi

İlk önce yolu verilen dosya bulunur ve içindeki tüm yazılar karakter karakter bir String’e aktarılır. Sonra bu String’den gereksiz olan şeyler silinir.

2.1 Tek satır, çoklu satır, string’ler ve char’ların temizlenmesi

String iki karakter kullanılarak okunur ve okunan karakterler aynen başka bir String’e aktarılır. Fakat bu aktarım sırasında eğer tırnak (“) , tek tırnak (‘), // ve /\* görürse bu aktarmayı durduruyor. Bu karakterler bizim silmek istediğimiz yapıların (yorum satırları vs.) önünde geldiği için dikkatle kontrol edilir. Bu yapıların sonuna geldiğine dair karakterler okununca tekrardan aktarım devam eder. Ayrıca bu yapılar birbirlerinin içinde kullanılabildiğinden böyle bir durumun olup olmadığı da kontrol edilir.

* 1. Şablon yapısına sahip sınıf isimlerinin alınması

Başlangıçta bu isimlerin tutulması için bir koleksiyon oluşturuluyor. String’de class isimlerinin yanında “<” işareti olup olmadığına bakarak onun şablon yapısına sahip mi diye kontrol ediliyor. Eğer var ise o sınıfın ismi koleksiyona kaydediliyor. Ayrıca “ArrayList<String>” gibi şeylerde de sıkıntı çıkmaması için bu tür hazır koleksiyonların isimleri başlangıçta veriliyor.

* 1. Süslü parantezlerin içindeki ifadelerinin alınması

Bu bölümde, sadece koddaki üye değişkenleri ve metotların gövdelerini alınıp diğer yerler silinir. Bunun için String bir başka String’e karakter karakter kontrol edilerek aynen aktarılır. Süslü parantezlerin gelip gelmesine göre bazı ifadeleri alıyor ya da almıyor.

* 1. Şablon yapısına sahip sınıfların nesnelerinin silinmesi

Daha önceden şablon yapısına ait sınıfların isimlerini kaydetmiştik. Burada ise kodda bu isimlerin yanında “<” işaretinin olup olmadığına bakılır. Eğer var ise şablon silinir.

* 1. Operatörlerin sayımı

Bütün temizleme işlemlerinden sonra elimizde birçok şey arınmış bir String oluşuyor. Bu String, iki adet char veri yapısı kullanılarak karakter okunuyor. Bu iki veri yapısının aldığı değerlere göre operatörlerin ne olduğu belirleniyor. Belirlenen her bir operatör için o operatöre ait sayaç arttırılır. Bütün okuma işlemi bittikten sonra String’deki bütün operatörler sayılmış olur.

1. ÇIKTILAR

Verilen dosyadaki bütün ifadeleri yukarıda belirtilen işlemlerden geçirip ekrana o dosyanın içindeki Java kodunda kaç adet operatör olduğu hakkında bilgi çıkartır.

1. SONUÇ

Bu ödev ile birlikte derleyicilerin yaptığı lexical analiz hakkında bilgi sahibi oldum.