



T.C.
SAKARYA ÜNİVERSİTESİ

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

KONSOLDAN VERİLEN *.JAVA DOSYASINI
ANALİZ ETME

G191210065 - Mehmetali Demirtaş

SAKARYA

Mart, 2022

Programlama Dillerinin Prensipleri Dersi

.JAVA DOSYASINI ANALİZ ETME

Mehmetali Demirtaş

mehmetali.demirtas@ogr.sakarya.edu.tr

G191210065 - 2.Öğretim C grubu

Özet

Konsol üzerinden komut satırı parametresi olarak verilen .java dosyasını analiz etmemiz istenmektedir. Analiz sonucunda tekli operatör sayısı, ikili operatör sayısı, sayısal operatör sayısı, ilişkisel operatör sayısı,, ilişkisel operatör sayısı, mantıksal operatör sayısı ve toplam operand sayısı bulunmaktadır.

© 2017 Sakarya Üniversitesi.

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

Anahtar Kelimeler: Operatör , operand , komut satırı parametresi , runnable jar

1. GELİŞTİRİLEN YAZILIM

Program.java >> İlk olarak komut satırından alınan dosyadaki yorum satırları silinmektedir. Switch case ile 4 farklı durum kontrol edilmektedir. Durumlar : Yorum Değil , Tekli Yorum Satırı , Çoklu Yorum Satırı , Çoklu Yorum Satırı İlk Satır.

İlk olarak yorum değil durumu kontrol edilmektedir. Dosyadan okunan karakter / ise , 3 ihtimal vardır. Sonraki karaktere göre durum belirlemesi yapılır. Eğer sonraki karakter / ise, tekli yorum satırıdır, * ise çoklu yorum satırının ilk satırıdır, bu ikisinden biri değilse okunan karakter yorum satırı değildir ve metin değişkenine eklenir.

Durum tekli yorum satırı ise, yeni satıra geçildiğinde (\n) yorum satırı sonlanmış, durum yorum değil durumuna eşitlenir.

Durum çoklu yorum ilk satır ise, yeni satıra geçildiğinde (\n) durum çoklu yorum satırına eşitlenir ve metine eklenir.

Durum çoklu yorum satırı ise, karakter * ve sonraki karakter / olana kadar, yani yorum satırı sonlanana kadar okunur, sonlandığı zaman durum yorum değil durumuna çevirilir.

Dosya okuma tamamlandıktan sonra oluşan yorumsuz yeni metin ifadesindeki string ifadeler de, operand ve operatör sayısına eklenmeyeceği için silinmektedir. Örneğin ; String x = "a+b*c" ifadesi, hesaba katılmamalıdır. Bu yüzden iki tırnak işareti arasındaki ifadeleri replaceAll metodu ile silinmektedir. Daha sonra Lexical sınıfında tanımlanan regex ifadelerin düzgün çalışması için metni tek satır haline getirilir.

Metnin son düzenlenmiş hali, Lexical sınıfından oluşturulan nesne ile operatorVeOperandBul fonksiyonuna parametre olarak verilir. Son olarak Operatör ve operand bilgileri Lexical sınıfından getter ile çekilerek ekrana yazılmaktadır.

Lexical.java >> İlk olarak sayaçlar tanımlandı ve Program.java sınıfından erişebilmek için sayaçlar için getter metodları oluşturuldu.

Daha sonra operatör ve operandları bulacak bir fonksiyon oluşturuldu.

Tekli operatörleri bulmak için iki farklı regex ifadesi oluşturuldu. İlk regex x++ ve ++x ifadelerini buluyor. İkinci regex -x ve +x gibi (pozitif/negatif) ifadeleri buluyor. Tekli sayaçları artırdığımız gibi, ikili regex tekli

ifadeleri de bulduğu için yardımcı bir sayaç oluşturulup artırıldı. Tekli operatörler tek operad barındıracağı için operand sayacına tekli operatör sayısı eklendi.

Mantıksal operatör bulmak için 2 farklı regex oluşturuldu, ilk regex && ve || ifadelerini buluyor. İkinci regex ise ! ifadesini buluyor. Mantıksal operatörler 2 operand içereceği için operatör sayacına mantıksal sayacın 2 katı eklendi.

İlişkisel operatör bulmak için her birine ayrı regex oluşturuldu. İlk regex == , ikinci regex <= ve >= , üçüncü regex != ve dördüncü regex < ile > ifadelerini bulmaktadır. == ifadesini bulduktan sonra ikili için oluşturulan yardımcı sayaçda artırılıyor. İlişkisel operatörler 2 operand içereceği için operatör sayacına ilişkisel sayacın 2 katı eklendi.

İkili operatör bulmak için tek bir regex oluşturuldu. Bu ifade ilişkisel operatörlerden == ve tekli operatörleri de bulduğu için ikili sayacından yardımcı sayaç çıkarılmaktadır. İkili operatörler 2 operand içereceği için operatör sayacına ikili sayacın 2 katı eklendi.

Sayısal operatör sayısı da tekli sayaç ve ikili sayaç toplamına eşittir.

ZORLANDIĞIM KISIMLAR : Regex ifadelerde ikili operatörler için özel bir regex oluşturamadım. Tekli ve ilişkisel operatörlerin bir kısmını da bulduğu için yardımcı bir sayaç oluşturdum ve ikili operatör sayısından yardımcı sayacı çıkarttım. Operand hesaplama da zorlandım.

EKSİK BIRAKTIĞIM YERLER : Operatör sayma esnasında eğer bir ifade de bir operatör birden fazla kullanıldıysa (örn: x+y+z) , operatör sayısı 2 ve operand sayısı 3 olmalı. Yaptığım fonksiyonda bu ayrımı yapamadığım için operand sayısını 4 olarak bulmaktayım.

2. ÇIKTILAR

```
Seç C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19043.1586]
(c) Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\Mehmetali Demirtaş\Desktop\PDPOdev\PDP_Odev1\dist>java -jar Program.jar deneme.java
Operatör Bilgisi:
    Tekli Operatör Sayısı: 1
    İkili Operatör Sayısı: 6
    Sayısal Operatör Sayısı: 7
    İlişkisel Operatör Sayısı: 3
    Mantıksal Operatör Sayısı: 2
Operand Bilgisi:
    Toplam Operand Sayısı:23

C:\Users\Mehmetali Demirtaş\Desktop\PDPOdev\PDP_Odev1\dist>java -jar Program.jar deneme.java
Operatör Bilgisi:
    Tekli Operatör Sayısı: 2
    İkili Operatör Sayısı: 7
    Sayısal Operatör Sayısı: 9
    İlişkisel Operatör Sayısı: 4
    Mantıksal Operatör Sayısı: 4
Operand Bilgisi:
    Toplam Operand Sayısı:32
```

Verilen deneme.java dosyası doğru bir şekilde analiz edilmekte, deneme.java üzerinde değişiklik yapıldığında da doğru sonuçları vermektedir. Yorum satırları ve string ifadeler analize dahil edilmemekte, sorunsuz çalışmaktadır.

3. SONUÇ

Sonuç olarak komut satırından verilen parametredeki operatör ve operand sayıları regex kullanılarak bulunabilmektedir.

Referanslar

- [1] <https://regex101.com/>
- [2] <https://www.javatpoint.com/operators-in-java>