



T.C.
SAKARYA ÜNİVERSİTESİ

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

PDP ÖDEV 1

B201210038 - Beyza ERKAN

SAKARYA
Mart, 2022

OPERATÖRLER

Beyza Erkan

B201210038 1.Öğretim C Grubu

Özet

Komut satırı argümanı olarak verilen java dosyasını okuyarak içindeki operatörlerin sınıflarına göre sayılarını ve operandların sayısını bulan konsol uygulaması.

© 2022 Sakarya Üniversitesi.

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

Regex, operatörler, operand, komut satırı argümanları, jar

1. PROGRAMLAMA ADIMLARI

1.0 Lexical Sınıfı

Regexlerin ve sayaclarinin tanımlandığı sınıf

1.1 Tanımlamalar

Lexical sınıfında tekli, ikili, mantıksal, ilişkisel operatörlerin regexlerini tanımladım. Ve bu regexleri sayacak olan ayrı ayrı sayaçlar tanımladım.

Sayısal operatörlerin sayısını tekli ve ikili operatörleri topladığımızda bulabiliyoruz o yüzden sayısal operatörler için yeni bir pattern tanımlamadım.

İlişkisel, mantıksal, sayısal, ikili ve tekli operatörler dışında operand sayısını bulmayı kolaylaştıracak iki regex daha yazdım. singleOperands adlı pattern tek operand içeren (++ -- !) operatörleri bulur. dualLogicalOperands adlı pattern iki operand içeren mantıksal operatörleri (&& ||) bulur.

Bu iki regex yardımıyla kolayca operand sayısına ulaşabileceğiz.

2.0 Program Sınıfı (Ana Sınıf)

.java uzantılı dosya okuma işleminin, dosyadaki operatörler ve operandların sayısının hesaplandığı ana sınıf.

2.1 Tanımlamalar

Dosyayı satır satır okuması için “line” adlı string, dosyayı tek bir string içinde tutması için “key” adlı iki string tanımladım.

2.2 Dosya Okuma İşlemleri

Komut satırından verilen java dosyasını dosya okuyucuya parametre olarak verdim. Dosyayı line adlı değişkenle satır satır okuyor ve key adlı değişkene atıyor. Yani key, okunan dosyayı tek bir stringde tutuyor. Amacı blok halindeki yorum satırlarını silmeyi daha kolay hale getirmektir.

2.3 Yorum Satırlarını Silme

Daha sonra yorum satırlarını bulduğunda o kısımları boşluk karakteriyle değiştiriyor. Bunu, `/* * * */`, `//` yorum satırlarıyla ilgili regexleri `replaceAll` fonksiyonuna vererek boşluk karakteriyle değiştirmesini sağlayarak yaptım. Sonucunda yorum satırları silinmiş olacak ve arama rahat bir şekilde yapılabilecek.

2.4 Lexical Objesi, Patternlar, Matcherlar ve Sayaçlar

Lexical sınıfından bir obje çağırdım. Objedeki patternleri yakalaması için operatörlerin sınıflarına göre matcherlar tanımladım. Matchlerlar key değişkeninde eşleşme yapacak. While döngüsü ile find fonksiyonuyla operatörleri bulduğunda ilgili sayacı artıracak. Sayısal operatörler için tekli ve ikili operatörlerin sayısını toplamamız yeterli olacaktır. En sonunda sayaçları konsola yazdıracak.

2.5 Operand Sayısı

- İlişkisel operatörlerin hepsi iki operand içerir.
- İkili operatorlerin hepsi iki operand içerir
- Yukarıdakilerin dışında iki operand içeren `&&` `||` operatörleridir.
- Tek operand icerenler `++` `--` `!` operatörleridir.

İkili operand içerenleri ikiyle çarpıp tek operand içerenlerle topladığımızda operand sayısını buluruz.

```
(operators.singleOperandCounter + operators.dualLogicalOperandCounter*2 +
operators.dualCounter*2 + operators.relationalCounter*2)
```

3.0 Hatalar

İstenmeyen bir durum ortaya çıkarsa hata fırlatır.

2. ÇIKTILAR

Okutulan içerik:

```
package odev;
public class Deneme {
    public int x;
    public int y;
    public Deneme(int x,int y) {
        this.x = x;
        this.y = y;
    }
    /*
    *
    * x*x + y*y // işlemi x > 0 ve y > 0
    */
    public int x2y2() {
        return x * x + y * y;
    }
    /*
    * Karakök işlemi yapan fonksiyon
    * karakok(x*x + y*y)
    */
    public double karakok() {
        return Math.sqrt((double)x2y2());
    }
    // log2(x/++y)
    public double log2() {
        ++y;
        return Math.log(((double)x)/y);
    }
    @Override
    public boolean equals(Object obj) {
        if(obj != null && obj instanceof Deneme) {
            return ((Deneme)obj).x == this.x && ((Deneme)obj).y == this.y;
        }
        return false;
    }
}
```

Çıktı:

```
PS C:\Users\Beyza\Desktop\B201210038\dist> java -jar Program.jar "C:\Users\Beyza\Desktop\text.java"
Operator bilgisi:
    Tekli Operator Bilgisi: 1
    İkili Operator Bilgisi: 6
    Sayısal Operator Bilgisi: 7
    İlişkisel Operator Bilgisi: 3
    Mantıksal Operator Bilgisi: 2
Operand bilgisi:
    Toplam Operand Sayısı: 23
```

3. SONUÇ

Komut satırı argümanı olarak verilen dosya okunarak içindeki sınıflarına göre operator sayıları ve operandların sayısı ekrana yazıldı.

Günlük hayatta kullanılabilecek yerler:

Kullanıcı kayıt işlemlerinde mail, telefon numarası formatının kontrol edilmesi

Google arama sonuçlarında yazım hatası yapıldığında “Bunu mu demek istediniz” ibaresi vererek öneriler çıkarması.

Referanslar

- [1] <https://regex101.com>
- [2] <https://www.primacsi.com/regular-expressions-regex-nedir-nasil-kullanilir/>
- [3] https://www.youtube.com/watch?v=bF_zEzFQZuA
- [4] <https://github.com/gkandemi/regex>
- [5] <https://support.smartbear.com/alertsite/docs/monitors/web/selenium/export-eclipse-java-project-as-runnable-jar.html>