



**T.C.**  
**SAKARYA ÜNİVERSİTESİ**

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

**Java Projesi**

**B201210302 - Rabia Abdioğlu**

**SAKARYA**

**Mart, 2022**

Programlama Dillerinin Prensipleri Dersi

# Java Proje Ödevi

Rabia Abdioğlu

B201210302

## Özet

Eclipse IDE ortamında, Java programlara dilinde olan tüm operatörler ve programda kullanılmış operandları bulan program yazdım. Yazdığım program runnable jar olarak export edildi. Dist klasörü içerisinde bulunan test edilecek java dosyasını parametre olarak alarak, yapılan hesaplamaları ekrana yazdırıyor. İstenilen tüm özellikler yapılmıştır. Eksik veya fazla parametre (argument) alması durumunda gerekli hataları veriyor.

© 2022 Sakarya Üniversitesi.

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

Anahtar Kelimeler: Regex, Java, Runnable, Jar

## 1. KOD KISMI

### a. Program.java

Io ve util.Scanner kütüphaneleri eklendikten sonra main fonksiyonu içerisinde değişkenler belirlenir ve ilk değerleri verilir. Çalıştırılabilir jar dosyası olarak çalışacak programın, argüman sayısı kontrolü yapılır ve adı verilen (dist klasörü içerisinde olması gereken) java dosyasının içeriği satır satır yedek stringe atılır.

```
//shell'den gelecek parametre sayısı kontrolü
if(args.length==2) System.out.println("Birden çok parametre girildi");
else if(args.length==0) System.out.println("Dosya parametresi girilmedi");
else {
    // parametre olarak verilen dosya yolu ile
    //java dosyası içeriğini satır satır okumak için
    try {
        Scanner scanner = new Scanner(new File(System.getProperty("user.dir")+"\\\\"+args[0]));
        while (scanner.hasNextLine()) tmp+=scanner.nextLine()+"\n";
        scanner.close();
    }
}
```

Yedek string (tmp) kontrol edilir. Yorum satırı veya string ifadelerin içerisinde operatör bulunması programın hatalı çalışmasına sebep olabilir. Bu sebeple istenmeyen durumlar kontrol edilerek , kod değişkenine atılır. Lexical sınıfındaki control fonksiyonuna, code parametresi yollanır. Hesaplamalar sonucu string ekran çıktısı döndürecektir.

*Lexical.control(code)*

### b. Lexical.java

Operatör ve operand hesaplamalarının yapılacağı bu sınıfta Regex kütüphanesine ait Matcher ve Pattern kullanılacaktır. Regular Expression, kod içerisinde operatörleri bulmak için belirli kurallar ile arama yapmamızı sağlayacak söz dizimidir. Match ile regular ifadeyi sağlayan karakterler bulunuyor.

```
//SAYISAL OPERATÖR

pattern=Pattern.compile("\\*|\\/|[%]|\\w\\d\\s|=\\w\\d\\s|\\w\\d\\s\\+|\\w\\s\\d|\\w\\d-\\w\\d");
match = pattern.matcher(line);
while(match.find()) {ikili++; sayisal_S++;}

match.reset();

//tekli ve sayisal

pattern=Pattern.compile("\\+|\\+|--|^[^\\w\\d\\+|\\+|\\w\\d|^[^~\\w\\d-\\d\\w]");
match = pattern.matcher(line);
while(match.find()) {tekli++; sayisal_S++; }
```

## 2. ÇIKTILAR

‘Deneme.java’ ödev dökümanındaki kodları, ‘Deneme2.java’ farklı örnek kodu içerir.

```
C:\Users\rabia\Desktop\B201210302\Odev_Java\dist>java -jar Program.jar Deneme2.java

Operatör Bilgisi:
  Tekli Operatör Sayısı:  0
  İkili Operatör Sayısı: 22
  Sayısal Operatör Sayısı: 22
  İlişkisel Operatör Sayısı: 12
  Mantıksal Operatör Sayısı: 3
Operand Bilgisi:
  Toplam Operand Sayısı: 74
```

```
C:\Users\rabia\Desktop\B201210302\Odev_Java\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
```

## 3. SONUÇ

Regex üzerine araştırma yapıldı, öğrenildi. Başarılı bir şekilde operatör ve operand sayıları ekrana yazdırıldı.

## 4. NOT

Genel olarak internet üzerinden araştırma yapıldığından site linklerini yazdım. Kod içerisinde yorum satırlarını bol tuttuğum için raporu kısa tutmaya çalıştım.

Ödevi elimden geldiğince yapmaya çalıştım. Beni zorlayan kısım Regexi baştan öğrenmeye çalışmak oldu. Çünkü çok karışık bir yapısı var. Bu sebeple ufak bir hatayı bir kaç saate ancak farkedebildim. Runnable jar kısmında hiç zorlanmadım. Ödev genel olarak iyi ama muhtemelen eksiklerim vardır.

## Kaynaklar

- [1] <https://github.com/rabiaabdioglu/Notlar/blob/main/Java.md>
- [2] <https://regexr.com/>
- [3] <https://docs.microsoft.com/tr-tr/dotnet/api/system.text.regularexpressions.regex.match?view=net-6.0>
- [4] <https://www.softwaretestinghelp.com/java/java-operators-arithmetic-unary-bitwise-operators-in-java/>

