



**T.C.**  
**SAKARYA ÜNİVERSİTESİ**

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

**DOSYADAN VERİ OKUMA**

**B201210059 - Yusuf YESİN**

**SAKARYA**

**Mart, 2022**

Programlama Dillerinin Prensipleri Dersi

**DOSYADAN VERİ OKUMA**

YUSUF YESİN

B201210059

## 1. Öğretim C Grubu

### Özet

Problem bize verilen java uzantılı dosyayı okumak ve dosya içerisindeki tekli operatör, ikili operatör, sayısal operatör, ilişkisel operatör, mantıksal operatörleri ve operandları saymamız.

İlk olarak dosya okuma kısmını yaptım. while döngüsü ile satır satır okudum. Her satırı da for döngüsü ile karakter karakter okudum. If ile karakterlerin istenen karakter olup olmadığını kontrol ettim. Eğer karakter istenen karakterse sayacı bir arttırdım. Eğer okunan satır yorum satırı ise bu satırları atlardım. Aynı şekilde tırnak işareti içerisindeki karakterleri de atlardım. Son olarak sonuçları ekrana yazdırdım.

© 2017 Sakarya Üniversitesi.

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

Emek, özveri, java

### • GELİŞTİRİLEN YAZILIM

Scanner kütüphanesini kullanarak dosyayı okudum. while döngüsü içerisinde dosyadaki her bir satırı sırayla String değişkenine attım. while döngüsünün içinde for döngüsü ile satır değişkenindeki her karakteri teker teker okudum. Burada charAt() metodunu kullandım. İlk olarak yorum satırlarını kontrol ettim. Boolean değişken tanımladım ve yorum satırı gördüğümde veya tırnak işareti gördüğümde değişken true değerini alıyor. Yorum satırı bittiğinde veya tırnak işareti kapandığında ise false oluyor. Karakterler teker teker okunurken if ile istenen operatöre eşit olup olmadığına baktım ve bunu sadece boolean değişkenimin değeri false iken yaptım. Böylece yorum satırlarını ve tırnak işareti içerisindeki string ifadeleri atlamış oldum. Eğer okunan karakter istenen operatöre eşit ise gerekli sayacı bir arttırdım ve operatör tekli operatör ise operand sayacı bir arttırdım. Eğer operatör ikili operatör ise operand sayacı iki arttırdım. Ödevde en zorlandığım kısım ArrayList<Double> gibi bir ifade ile karşılaştığımda burdaki "< , >" işaretlerini ilişkisel operatörden ayırmak oldu. Bunu regex kullanarak yaptım. Şöyle ki bu işaretler içerisinde kelime varsa bu ifadeleri atlardım. Son olarak operatör ve operand sayısını ekrana yazdırdım.

### • ÇIKTILAR

Geliştirmiş olduğum yazılımı birden fazla girdi ile test ettim. Birbirine yakın sonuçlar verdi. Fakat girdinin çok karmaşık bir şekilde yazılması hatalı sonuç vermesine sebep olabiliyor. Girdinin hatalı ifadeler içermesi durumunda da sonuç hatalı olabiliyor.

### • SONUÇ

Java dilini daha etkin bir şekilde kullanmayı öğrendim. Regex kavramını öğrendim. Bir metin içerisindeki telefon numarası, mail adresi veya bağlantı adresini okuyabilecek bir yazılım yazabilecek seviyeye ulaştım.