



**T.C.**  
**SAKARYA ÜNİVERSİTESİ**

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

**1. ÖDEV**

**G191210066 - Furkan İsmet Tufan**

**SAKARYA**

**Nisan, 2021**

Programlama Dillerinin Prensipleri Dersi

Java Kullanarak Cpp Dosyasının Analizi

Furkan İsmet Tufan \*

g191210066 – 2.Öğretim C Grubu

---

## Özet

Problem, java kullanarak cpp dosyasını açıp analiz etmek ve ekrana sınıf , super sınıf , metot bilgilerini listelemek. Problemin çözümü için ağırlıklı olarak regex kullandım. Uygulamada Program.cpp dosyasını açıp hazırladığım regexler yardımıyla istediğim yapıları buldum ve ekrana belirli bir formatta yazdırdım.

© 2021 Sakarya Üniversitesi.

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

Anahtar Kelimeler: regex, java, c++

---

## 1. GELİŞTİRİLEN YAZILIM

### 1.1 Ödevin Amacı

Ödev; programlama dillerinin parçalarını daha iyi anlamamız, kurallarını daha net öğrenmemiz, regex mantığını öğrenmemiz ve bir cpp dosyasını analiz ederek sınıfları ve metotları yakalayabilmemiz için verilmiş olabilir.

### 1.2 Ödeve Başlangıç

Ödevde istenilen parçalamayı yapabilmek için metot ve sınıf tanımlamalarını diğer tanımlamalardan ayırt eden özellikleri düşünmem gerekti. Bu ayırt edici özellikleri not aldım ve bunları kullanarak regex yardımıyla istenilen parçalamayı yapabileceğimi düşündüm. Regex'in nasıl kullanıldığını öğrendim ve istediğim regex'i yazabilmek için [regexr.com/](http://regexr.com/) sitesinden denemeler yaptım. Bulduğum regex kodlarını Pattern ve Matcher sınıflarını kullanarak java dosyama ekledim.

### 1.3 Ödevdeki Kontroller

Regex ile ulaştığım ifadelerle bolca oynama yapmam gerekti. Ulaştığım metot isminin if, for, switch, while gibi dilin öğelerinden olup olmadığını anyMatch() metoduyla toplu olarak kontrol ettim. Metodun yıkıcı olup olmadığını ulaştığım metod isminde '~' sembolünü arayarak öğrendim. Metodun kurucu metod olma durumunu öğrenmek için, yeni class bulunduğunda kaydedeceğim bir değişken oluşturdum ve bulunan metot isimlerini mevcut class ismiyle kıyaslayarak kurucu metod olma durumunu kontrol ettim.

---

\* Ödev Sorumlusu. Furkan İsmet Tufan, g191210066,  
Mail Adresi: [furkan.tufan@ogr.sakarya.edu.tr](mailto:furkan.tufan@ogr.sakarya.edu.tr)

Fonksiyon parametrelerini ve çoklu kalıtım durumlarını virgüllere göre ayırdım. Çoklu kalıtım durumunda; public, private, virtual, protected gibi sınıf isimlerinden bağımsız öğeleri replace() fonksiyonuyla kaldırdım ve geri kalanları virgüllere göre aldım.

Kalıtım alınan sınıfların sayılarını bulabilmek için bulduğum super sınıfları bir arraylist içerisine ekledim ve programın sonunda arraylist içerisindeki string ifadeleri equals() fonksiyonuyla kıyaslayarak super sınıfların kalıtım alınma sayılarını buldum.

## 2. ÇIKTILAR

Metot isminde '~' varsa metodun dönüş türüne void yazıyor.

"int \*topla" gibi pointer dönüş değeri olan metotlarda metot ismindeki "\*" işareti gözükmüyor ve dönüş değerinin sonuna "\*" ekleniyor. Aynı durum metotların parametre kısımlarında da uygulanıyor.

Class ismiyle aynı isimde bir metot olursa dönüş değeri olarak "Nesne Adresi" yazdırılıyor.

## 3. SONUÇ

### 3.1 Ödevde Öğrendiklerim

- Daha öncesinde hiç java kodu yazmamıştım. Ödevi yapabilmek için java'nın, bildiğim programlama dillerinden farklarını öğrendim ve javayla konsol uygulaması yazabilecek duruma geldim.
- String değişkenler konusunda daha fazla bilgi sahibi oldum. indexOf() metoduyla istenilen karakterin string ifadede kaçınıcı indexte olduğu bilgisini alabileceğimi ve bu metodun -1 döndürmesi durumunda istediğim karakterin string ifadede olmadığı bilgisini alabileceğimi öğrendim. replace() metoduyla string ifadeden istediğim kelimeleri ve karakterleri çıkarabileceğimi öğrendim ve regex yoluyla ulaştığım satırdan gereksiz bulduğum kısımları bu metodu kullanarak çıkarttım. contains() metoduyla string bir ifadenin içinde aradığımız kısmın olma bilgisini alabileceğimizi öğrendim.
- Regex'in kurallarını, ne şekilde uygulandığını öğrendim.

### 3.2 Ödevde Zorlandıklarım

- Daha önce hiç regex kullanmamıştım. Başlangıçta bu kadar sembol çok korkunç geldi ve alışması zor oldu ancak araştırarak ve denemeler yaparak alıştım.
- C++ dilinde çok fazla yazım şekli olduğu ve çok sayıda boşluk koyulabilmesinden dolayı tüm yazım şekillerini düşünmek zor oldu. Regex ifademde boşluk olma ihtimali olan her yere \\s\* eklemem gerekti.
- Bulduğum metodun public metot olma durumunu kontrol etmek en çok zorlandığım kısımdı.
- Super sınıfların kalıtım alınma sayılarını bulmakta çok zorlandım.

## Referanslar

- [1] <https://www.geeksforgeeks.org/>
- [2] <https://www.w3schools.com/>
- [3] <https://stackoverflow.com/>