



T.C.
SAKARYA ÜNİVERSİTESİ

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

C++ DOSYASI OKUYARAK SINIFLARI VE PUBLIC
METODLARINI LİSTELEME

G191210303 - Barış YILMAZ

SAKARYA

Nisan, 2021

Programlama Dillerinin Prensipleri Dersi

C++ DOSYASI OKUYARAK SINIFLARI VE PUBLIC METODLARINI LİSTELEME

Bariş YILMAZ^{a*}

^a G191210303 2/C

Özet

Ödevde Java ile C++ dosyasının okutularak bu dosyadaki sınıfların, bu sınıflara ait public metodların isim, aldığı parametre sayısı, aldığı parametrelerin türlerinin, metodun ne türden bir şey döndürdüğünün ve son olarak ise kalıtım alınan sınıf isimleri ile kaç defa aldıklarının listelenmesi istendi. Genel olarak regex kullanımı ile sınıf, metod ve parametre okuma ifadeleri tasarlayarak gruplama yoluyla istediğim bilgilere erişerek bazı kontroller sonrasında ArrayListlere aktararak son olarak ise ekrana çıktılarını verdim.

© 2021 Sakarya Üniversitesi.

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

Anahtar Kelimeler: Java, Regex, C++, Dosyalama

1. GELİŞTİRİLEN YAZILIM

Öncelikle src klasörü içerisinde bulunan Program.cpp dosyasının yolunu Path ile aldıktan sonra Files ile birlikte dosyayı bir string değişkenine aktardım. Sonrasında dosyadan satırlarda yorum bulunuyor ise bunları replaceAll ile sildim. Satır satır okumama sebebim ise ifadelerin sonraki satırlarda devam edebilme olasılığı. Sonraki satırlarda devam eden ifadeleri de regex ile yakalayabilmek içinse oluşturduğum regexlerde DOTALL flagini kullandım. Öncelikle sınıfları yakalayabilmek için class'ı bir bütün olarak yakalayan(};'e kadar) bir regex pattern'ı hazırladım. Bu patternda grup1 sınıf adı, grup3 kalıtım alınan sınıf/ları, grup5 public: sonrası sınıf bitimine kadar olan kısmı yakalamaktadır. Sınıfa ait pattern'ın matcher'ına dosyanın string değişkenini atayarak while döngüsü ile sınıfları bulduğu sürece devam etmesini sağladım. Sınıf bulunduğunda öncelikle siniflar ArrayList'ine sınıfın adını ekleyip sonrasında kalıtım alınan sınıf/lar var ise bu süper sınıfları superSiniflar HashMap'ine eğer daha önceden eklendiyse value değerini 1 arttırarak, eklenmediyse yeni ekleme yaparak ekliyor. Sınıf içerisinde public: bulunuyorsa public bulduktan sonra sınıf sonuna kadar yakaladığı için sonrasında private: veya protected: var ise o kısımları çıkarması için substring ile o kısım kırılıyor. Bu kontrol sonrası bazı metodların içerisinde diğer bir metodun çağırılması vb. sıkıntılı durumlar için en sağlıklısının bu olduğunu düşünerek public: içerisindeki tüm metodları süslü parantez aç ve kapat sayılarını sayarak metod metod ayırarak bu metodları metodlarListesi ArrayList'ine aktardım. Grup1'i ilk kelimeyi, grup3'ü ikinci kelimeyi, grup4'ü parametreleri yakalayan metod regexi yardımıyla bu metodlarListesi ArrayList'ini gezerek metod regexinin matcherına gezdiğim elemanı vererek metod kontrolü yaptıktan sonra eğer metod nesne adresi döndüren veya yıkıcı metod değil ise yani tek kelime sonrası parametreler gelmiyorsa 3. grubun null olup olmamasından bunu yakalayıp ona göre dönüş türlerini ve ayrıca metod isimlerini ekledim. Her iki durum için ise parametreleri tutan grup 4 null ise parametrelere 0'ı boş değil ise parametreleri , 'den split ile bölerek stringde toplayıp tek bir

* Ödev Sorumlusu. Bariş YILMAZ, G191210303,
Mail Adresi: baris.yilmaz5@ogr.sakarya.edu.tr

string olarak parametreler ArrayList'ine ekledim. Sınıf ait tüm metot ve parametrelerinin eklenmesinin ardından sınıflara ait metotların ayrıştırılabilmesi için metotIndeksTutucu adında bir ArrayList oluşturarak yazdırma esnasında 1. sınıfa metot sayısını metotIndeksTutucunun 1. indeksindeki adet kadar metotlar ArrayList'inde giderek bulabileceğim yapıyı oluşturdum. Aynı mantıkla 2. sınıfa ait metot sayısı da bu sebeple metotIndeksTutucunun 1. indeksindeki sayı+yeni metot sayısı oluyor. En son yazdırırken örneğin 2. sınıfı yazdıracakken metotIndeksTutucu'nun 1. İndeksindeki sayıdan 2. İndeksindeki sayıya metotlar ArrayList'inin indeksinde gittiğimde sınıfa ait metotları ve aynı mantıkla parametrelerini yakalayabilmiş oldum. While döngüsü bitmeden önce yani yeni sınıfı aramadan önce ise metot sayacını 0'a eşitledim ve public olan metotları böldüğüm metotlarListesi ArrayList'ini temizledim. En son ise dokümanda bahsedildiği şekilde sınıfları ve bundan sonra var ise süper sınıfları yazdırdım.

2. ÇIKTILAR

Elimden geldiğince olası tüm yazım şekillerini de değerlendirerek hazırladığım bu ödevde karmakarışık dosyalarda dahi çıktıda herhangi bir problem yaşamadım. Program düzgün bir şekilde çalıştı.

3. SONUÇ

Regex'in kullanımıyla ilgili pek çok şey tecrübe ettim.

Referanslar

[1] <https://docs.oracle.com/javase/tutorial/essential/regex/>