



T.C.
SAKARYA ÜNİVERSİTESİ

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

1. ÖDEV

B201210387- ALEYNA ELİF ÖZKAN

SAKARYA

NİSAN,2021

Programlama Dillerinin Prensipleri Dersi

NDP 1. ÖDEV

ALEyna ELİF ÖZKAN

B201210387

1-C

Özet

Bu ödevde verilen bir C++ kodunu belli kurallara göre parçalamamız istendi. Öncelikle bu kodun içinde bulunan sınıflar, sonrasında bu sınıfların içinde bulunan public fonksiyonlar ve bu fonksiyonların kaçar parametre aldıkları ve türlerinin ne olduğunu ne tür bir şey döndürdüğünü bulmamız gerekiyordu. Aynı zamanda kalıtım alınan sınıfların da isimlerini ve kaçar kez kalıtım alındığını bulmamız istendi. Bunu yapabilmek için hem C++ dilinin kurallarını iyi bilmemiz gerekiyor -çünkü dilin izin verdiği her olasılığı düşünmemiz gerekli- hem de bu programı yazabilmek için java dilini çok iyi bilmemiz gerekiyor. Bu ödev aynı zamanda dosya okuma ve string methodları gibi konularda da alıştırma yapmamızı sağlıyor.

© 2021 Sakarya Üniversitesi.

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

Anahtar Kelimeler: java, C++, class, string

1. GELİŞTİRİLEN YAZILIM

Öncelikle class isimlerini bulabilmek için algoritma geliştirdim. Bunun için regex kullandım. C++ dilinde class adı belirtilebilmesi `class X {.....};` ifadesi ile mümkündür. Bu yüzden her class adından önce mutlaka class kelimesi yazılmalıdır. Ancak C++ dili arada birden fazla boşluk bulundurma durumunu veya arada yeni satır olması durumunu da kabul etmektedir. Bu yüzden regexi yazarken bu durumları da göz önünde bulundurdum.

Class isimlerini bulduktan sonra bu sefer public kısımları bulmamız gerekiyordu. C++ dilinde public tanımlanan fonksiyonlar `public:` ifadesi ile gösterilir. Classın içinde ve public kelimesinden sonra gelen her fonksiyon ismi bu alana dahildir. O yüzden bunu ayırmama yardımcı olacak bir regex ifadesi daha yazdım.

Bu fonksiyon isimlerinin hemen yanında (...) gösteriminin içinde fonksiyonların aldığı parametreler belirtilmektedir. Bu parametreler birbirinden virgüller aracılığıyla ayrılır. Regex yardımıyla virgüllere göre bu parametreleri ayırdım.

Parametre sayısı ise virgüllerin bir fazlasına eşit oluyordu. Ama fonksiyonları tipine ve ismine göre ayırmak o kadar kolay olmadı. Çünkü hem parametrelerde hem de fonksiyonlarda `const` kelimesi geçebilirdi. Önce bu kelimenin var olup olmadığını tespit edip ona göre işlem yapmamız gerekiyordu. Fonksiyon isimleri içinde boşluk bulunamayacağı için ‘(ifadesinden önceki kelime mutlaka fonksiyon ismi olmalıdır diye düşündüm. Ve işlemleri buna göre yaptım. Fonksiyon ismini bulduktan sonra bu fonksiyonun türüne göre farklı işlemler kullandım. Örneğin bu fonksiyon yıkıcı bir fonksiyonsa farklı, dönüş türü olan bir fonksiyonsa farklı yaklaşmam gerekiyordu. Eğer dönüş türü bulunduruyorsa bunu da çıktılarda belirttim. Süper classları bulmaya çalıştım. Eğer class bi kere kalıtım aldıysa bu classın adını ve sayısını bastırdım. Ancak bir class birden fazla classı kalıtım aldıysa o zaman bu classların adlarını bastırdım. Ancak sayılarını bastıramadım.

2. ÇIKTILAR

Ödev dosyasında verilen örnek C++ dosyasını denediğimde hata vermiyor. Başka classlarla denediğimde eğer bir sınıftan birden fazla kalıtım alındıysa bu süper classların sayılarını bastırma kısmında hata veriyor.

Test ettiğim kadarıyla fonksiyon isimlerinde < ve > bulunduğu durumlarda bu fonksiyonlara uygun bir durum yazamadığım için fonksiyon isimleri outputta hiç gözüküyor.

C++ dilinde fonksiyon ve parametreleri tanımlamada bir çok ihtimal olduğundan hepsine uygun bir durum yazabildim mi emin değilim. Özellikle de aralarda birden fazla boşluk veya yeni satır bulunması durumunda hatasız çalışıyor mu bilmiyorum. Belki bu durumların birkaçında gözümünden kaçmış kısımlar olabilir.

Const kelimesinin bulunma durumunu kontrol ettim. Onda sıkıntı çıkmayacağını düşünüyorum.

3. SONUÇ

Bu ödev sayesinde regexin ne olduğunu ve nasıl kullanıldığını öğrendim. Eğer bütün belgeyi koşullu ifadelerle parçalamaya çalışsaydım hem verimli olmayacaktı hem daha fazla vakit alacaktı hem de gözümünden bir şey kaçırma ihtimalim artacaktı. O yüzden [regex101](#) adlı siteyi kullanarak regex ifadelerini teker teker denedim. Bu sayede daha genel bir çözüm üretmiş olduğumu düşünüyorum.

Bu proje ile hem C++ dilinin kurallarına daha fazla hâkim oldum hem de java dilinde yazma pratiği elde ettim. Stringlerle alakalı methodların kullanılmasında pratik yapmış oldum ve daha önce kullanmadığım methodları kullandım. Bunları nasıl kullanacağımı araştırırken birçok bilgi edindim ve birçok örnek kod taradım. O yüzden bu ödevin programlama dillerini öğrenmek ve analiz etmek adına çok faydalı olduğunu düşünüyorum.

Referanslar

[1] Düzenli ifadeler (Regular Expressions | #Regex) Nedir? Nasıl Kullanılır? #JavaScript ile Regex – kablosuzkedi | [Youtube](#)

[2] [Regex101](#)