



T.C.
SAKARYA ÜNİVERSİTESİ

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

String Parçalama

Grup Elemanları:

G171210351 - Ahmet Armutcu

SAKARYA

Mart, 2017

Programlama Dillerinin Prensipleri Dersi

String Parçalama

^a G171210351 (2C) grubu

Özet

Projem NetBeans platformunda geliştirilmiş bir java uygulamasıdır. Daha önceden yazılmış olan java kaynak dosyasının içinde istenilen bilgileri belli şartlardan geçirerek ekrana yazan bir uygulamadır. Örneğin class adı, parametre adı, metot adı vb. Bu verileri java kaynak dosyasına ararken bazı şartlara dikkat edip arıyor. Bunuda regex komutları sayesinde yaptım.

© 2017 Sakarya Üniversitesi.

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

Anahtar Kelimeler: Regex,String,java,substring,class

1. GELİŞTİRİLEN YAZILIM

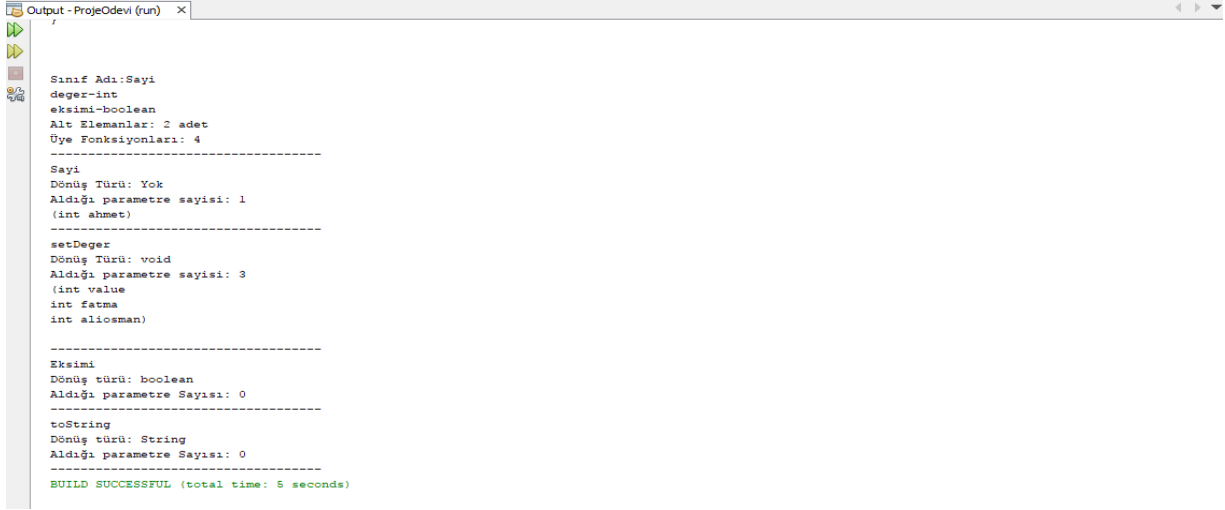
Projemde ilk olarak daha önce yazılmış olan java dosyasını satır satır BufferedReader ile okudum. İlk olarak class adını bulmaya çalıştım bunun için özel bir regex komutu kullandım. Parantez açıksa class adını yazdırdım.

Alt eleman sayısını öğrenmek istedim. Bununda özel bir regex komutu vardır. Bu komutta sonu noktalı virgül ile bitiyorsa alt elemandır dedim ve sayaç ile bu elemanları saydırarak ekrana yazdırdım. Eğer sonda noktalı virgül yoksa ekrana yazılmayacaktır.

Üye fonksiyonlarını bulmak için bazı metotları tanımladım. Dönüş tipi olan Dönüş tipi olmayan sadece public ile başlayan ve sonra aç kapa parantezi olması şartıyla tanımladım. Sonra bunları sayaç ile saydırdım. Ekrana yazdım. Sonra bu fonksiyonların isimlerini parametre sayısını parantez içindeki parametreleri substring, tim vb string kombinasyonları ile bazı işlemler yaparak ekrana yazdırdım. Örneğin parametre sayısını öğrenme için split yöntemi ile virgüle kadar aldım yani 1 virgül varsa 2 parametre olacak dedim. 2 virgül varsa 3 parametre var dedim. Eğer parametre yoksa bunun içinde metot içinde özel bir regex komutu ile match edip ekrana yazdırdım. Eğer metot isimlerinde aç kapa parantez yoksa metot olarak algılatmadım. Fonksiyon isimlerini alt eleman isimlerini ekrana yazdırırken dizi kullandım. Örneğin public int sayı; alt elemanı olsun. Regex buna uygun bir Pattern yazdım ve match ettim. Match find ile dosyada arattım. Sonra buna uygunsa 0 public,1int 2eleman oluyor. Diziye atadım ve ekrana yazdırdım. Kısaca Regex komutları ile dosyada arama yaptım ve regex uygun olanları ve istenilen bilgileri diziye aktarıp ekrana yazdırdım.

Bazı Özel Regex komutları

Örnek	Örnek
Class adı için	Pattern sinifadi=Pattern. compile("public+(\\s)*+class+(\\s)*+(\\S)+(\\s)*\\{");
Alt eleman	Pattern elemanlar=Pattern. compile("(public private)+(\\s)*(int boolean String double)+(\\s)*+(\\S)+(\\s)*;");



```

Output - ProjeOdevi (run) x
Sınıf Adı: Sayi
deger-int
eksimi-boolean
Alt Elemanlar: 2 adet
Üye Fonksiyonları: 4
-----
Sayi
Dönüş Türü: Yok
Aldığı parametre sayısı: 1
(int ahmet)
-----
setDeger
Dönüş Türü: void
Aldığı parametre sayısı: 3
(int value
int fatma
int aliosman)
-----
Eksimi
Dönüş türü: boolean
Aldığı parametre Sayısı: 0
-----
toString
Dönüş türü: String
Aldığı parametre Sayısı: 0
-----
BUILD SUCCESSFUL (total time: 5 seconds)

```

Projemin Çıktısı

2. ÇIKTILAR

Projemde çoğu istenilen işlemi yapıyor. Fakat bazı eksiklikler var. Örneğin class isminde sadece açılış parantezini kontrol ediyor sonra metod isminden sonra parantezi bir boşluk bırakması gerekiyor. Bunu yapamadım. Fakat class ismini alt eleman sayısını vb. çoğu şeyi yaptım. Projemin çıktısı yukarıdaki resimdir.

Projemde farklı bir sınıf oluşturarak burada işlemleri metodlar sayesinde parçalayarak oluşturdum. Main sınıfında sadece çalışması gereken kodları yazdım.

3. SONUÇ

Netbeans ile bir dosyadaki bir string ifadediyi istediğim komuta göre çağırmaı öğrendim. Regex ne olduğunu şimdiye kadar az biliyodum. Bu proje sayesinde regex ne olduğunu öğrendim. Bazı şeyleri yapamazsam javayı sevdim. Yeni bir platform olduğu için ilk zamanlarda zorluk çektim. Fakat alıştım.