



T.C.
SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
PROGRAMLAMA DİLLERİNİN PRENSİPLERİ 1. ÖDEV RAPORU

FONKSİYON, PARAMETRE VE OPERATÖR SAYISI SAYDIRABİLME

Grup Elemanları:

B161210367 - Beyza KARACA

B161210052 - Çiğdem Sıla UĞURLU

SAKARYA

Mart, 2019

FONKSİYON, PARAMETRE VE OPERATÖR SAYISI SAYDIRABİLME

Beyza KARACA, Çiğdem Sıla UĞURLU

B161210367 – 1. Öğretim B Grubu

B161210052 – 1. Öğretim B Grubu

Özet

Ödevimizde bizden bir Java programı yardımıyla başka bir C programını okumamız ve programın içinde bulunan fonksiyonların sayısını ve ismini, parametrelerin sayısını ve ismini son olarak da operatör sayısını bulmamız istendi.

© 2019 Sakarya Üniversitesi.

Bu rapor bizim özgün çalışmamızdır. Faydalanmış olduğumuz kaynakları içerisinde belirttik. Herhangi bir kopya işleminde sorumluluk bize aittir.

Biz bu problemi regular expressions yardımıyla çözmeye karar verdik. Regexleri internet sitelerinden öğrenerek problemimize uygun kurallar oluşturarak ödevde bizden istenilenleri uygulamaya çalıştık.

1.Geliştirilen Yazılım

Geliştirdiğimiz Java programımız Program.c dosyasını okuyor ve oluşturduğumuz regular expression kullarına göre okuduğu satırda fonksiyon, parametre veya operatör var mı diye anlamaya çalışıyor. Örneğin "((\\s)+)?(const(\\s)+)?(int|double|float|void|char)(\\s)+[a-zA-Z]+\\\\((((int|double|float|void|char)(\\s)+(&|\\s)*?[a-zA-Z]*|((int|double|float|void|char)(\\s)+(&|\\s)*?[a-zA-Z]*)))*)?\\\\(\\s)*[{}]*?((\\s)*?[\\\\\\\\n])*" regexine göre eğer satır belirli kelimelerden biri ile başlıyorsa (int, void, double, float) ardından herhangi bir isim ve “(“ parantezi varsa bunun bir fonksiyon olduğuna karar veriyor ve fonksiyon sayısı parametresini bir artırıyor. Parametre ve operatör sayıları da benzer şekilde oluşturulmuş kurallarla ayırt ediliyor ve ilgili sayaç bir arttırılabilir.

Fonksiyon ve parametre isimlerini alırken de match metodu verilen regex kurallarıyla eşleşen kelimeyi buluyor, find metodu ile var mı diye bakıyor ve group metoduyla da bulunan ve eşleşen kelimeyi alarak print ile ekrana yazdırabilmemizi sağlıyor.

2.Çıktılar

```
#include "stdio.h"
#include "stdlib.h"
void yazdirma(int *param,double g){
    int i=0;
    printf("%d ",p[i]);
}
int Toplasin(int &y,float uzunluk,char f){
    int i=0;
    int toplam=0;
    toplam += p[i];
    return toplam;
}
int main(){
    int sonuc = x+y;
    printf("Sonuc:%d\n\n",sonuc);
    int *dizi = malloc(3*sizeof(int));
    dizi[0]=x;
    dizi[1]=y;
    dizi[2]=sonuc;
    DiziYazdir(dizi,3);
    printf("\n\nToplam Deger:%d",DiziTopla(dizi,3));
    free(dizi);
    return 0;
}
```

```
run:
Toplam operatör sayısı:21
Toplam fonksiyon sayısı:3
Toplam parametre sayısı:4
Fonksiyon isimleri:
DiziYazdir-Parametreler: *p uzunluk
DiziTopla-Parametreler: *p uzunluk
main-Parametreler:

BUILD SUCCESSFUL (total time: 0 seconds)
```

Yukarıda görülen örneğimiz ve çıktısı ödevde bizde istenilen dosya ve çıktıdır.

```
#include "stdio.h"
#include "stdlib.h"
void yazdirma(int *param,double g){
    int i=0;
    printf("%d ",p[i]);
}
int Toplasin(int &y,float uzunluk,char f){
    int i=0;
    int toplam=0;
    toplam += p[i];
    return toplam;
}
int main(){
    int sonuc = x+y;
    printf("Sonuc:%d\n\n",sonuc);
    int *dizi = malloc(3*sizeof(int));
    dizi[0]=x;
    dizi[1]=y;
    dizi[2]=sonuc;
    DiziYazdir(dizi,3);
    printf("\n\nToplam Deger:%d",DiziTopla(dizi,3));
    free(dizi);
    return 0;
}
```

```
run:
Toplam operatör sayısı:15
Toplam fonksiyon sayısı:3
Toplam parametre sayısı:5
Fonksiyon isimleri:
yazdirma-Parametreler: *param g
Toplasin-Parametreler: &y uzunluk f
main-Parametreler:

BUILD SUCCESSFUL (total time: 0 seconds)
```

Burada görülen örnek ise denemelik yazılmış bir c dosyası ve çıktısıdır.

3.Sonuç

Gerçek hayatta programımıza benzer programlar programlama dili oluştururken veya programlama diline yeni özellikler eklerken dile özgü kurallar oluşturmada kullanılabilir. Bir internet sitesinde üyelerin kendilerine özgü şifre oluştururken uymaları gereken kuralları koyarken kullanılabilir. Bu ve bunlara benzer birçok uygulama alanı olabilir.

4.Ödev için kullanılan programlama: Java programlama dili

5.Ödev için kullanılan platform: NetBeans derleyicisi.

6.Referanslar

-<https://www.regexplanet.com/advanced/java/index.html>

-<https://docs.oracle.com/javase/tutorial/essential/regex/>

