



T.C.
SAKARYA ÜNİVERSİTESİ

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

ÖDEV BAŞLIĞI
Program.cpp Dosyasını Bölütleme

G191210068 - Ömer Faruk Güzel

2B

SAKARYA

Mart, 2021

Programlama Dillerinin Prensipleri Dersi

Program.cpp Dosyasını Bölütleme

Ömer Faruk Güzel

^a G191210068 – 2B

Özet

Problemim örnek bir cpp dosyasını sınıflara bölütleyerek bu sınıflar içerisindeki public metodları, metodların parametreleriyle birlikte geri dönüş tiplerini ve kalıtım alınan sınıfları sayıları ile isimlerini ekrana yazdırmak. Problemin çözümünde cpp dosyasını karakter karakter okutarak istenilen kodlara başlangıç karakteri kontrolüyle ulaştım ve bu kodun doğrumu kontrolüyle birlikte gereken sınıf isimlerini sınırlayıcı karakterler yardımı ile buldum ve metodlar içinde aynı işlemleri uygulayarak metodları ve parametreleri sınırlayıcı karakterler yardımı ile bularak ekrana yazdırdım

© 2021 Sakarya Üniversitesi.

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

Anahtar:Karakter,Sınırlamak,Koşul,Boşluk

1. GELİŞTİRİLEN YAZILIM

Öncelikle dosyayı okuturken karakter karakter okutmayı tercih ettim. Bu tercihimin sebebi cpp dosyalarının boşluk hassasiyeti olmaması ve bu hassasiyetin olmayışından kaynaklanan kelime bitişikliği veya ayrılığıydı.

Bende karakter karakter okutarak öncelikle class için sınırlayıcı baş karakter olan c harfine ulaştım ve boşluğa kadar okuttum. Okunan kelime class ise devam ettim ve bir sonraki boşluktan sonra okunacak kelime ise sınıfın adı oldu. Bu okumadan sonra eğer sınırlayıcı karakterim ':' ise kalıtım alınan sınıf var olduğu kontrolünü yaptım ardından da bu kalıtım alınan sınıfın sayısını bir sözlüğe atarak tuttum. Daha sonra sınıfın içine aç süslü parantez koşulu ile girerek aynı işlemleri public için de uyguladım ve okunan kelimenin public olması durumunda devam ettim ve sınıf sonuna veya yeni bir kapsüllemeye gelene kadar metodları okuttum. Metodları okuturken aç parantez sınırlayıcı karakter yardımı ile metdoun geri dönüş tipi olup olmadığını veya bu metdoun yapıcı, yıkıcı metod olup olmadığını kontrol ederek parametreleri varsa kapa süslü parantez koşuluna kadar okuttum. Bu esnada metod içinde başka bir süslü parantez bloğu olup olmadığını da kontrol ederek gereken süslü parantez sayısına kadar okuttum ve metodun kapama süslü parantezine ulaştım.

Bu Problemimin çözümünde en etken rol oynayan 4 fonksiyon ile gereken koşullu okumaları sağladım. Bu fonksiyonlar kod okuma, isim okuma, boşluk okuma ve sınırlayıcı karaktere kadar okuma.

Sistemin işleyişi:

- Class kelimesine gelmek
- Class ismini okumak
- Varsa kalıtım sınıfını okumak
- Sınıf içi public kapsüllemeyi okumak
- Public metod okumak
- Metod geri dönüş tipini tanımlamak
- Metod parametresini okumak
- Metod içi süslü parantez bloğu kontrolü
- Public dışı kapsülleme veya sınıf sonu kontrolü
- Dosya içi class bitene kadar döngüde kalmak
- Her bir kontrol arası boşluk kontrolü yapmak

2. ÇIKTILAR

Problems Javadoc Declaration Console
 <terminated> FirstClass (1) [Java Application] C:\Program Files\Java\jdk-15.0.2\bin\javaw.exe (Apr 11, 2021, 10:07:53 PM – 10:07:53 PM)

Sınıf: Kisi

```
Kisi
  parametre : 3 ( string, int, double )
  Donus Turu : Nesne Adresi
Kisi
  parametre : 0
  Donus Turu : Nesne Adresi
getIsim
  parametre : 0
  Donus Turu : string
getDogumYili
  parametre : 0
  Donus Turu : int
getKilo
  parametre : 0
  Donus Turu : double
YemekYe
  parametre : 1 ( double )
  Donus Turu : void
Kos
  parametre : 1 ( double )
  Donus Turu : void
```

Sınıf: Arac

```
Arac
  parametre : 4 ( string, int, double, double )
  Donus Turu : Nesne Adresi
Arac
  parametre : 0
  Donus Turu : Nesne Adresi
Sur
  parametre : 1 ( double )
  Donus Turu : void
~Arac
  parametre : 0
  Donus Turu : void
```

Sınıf: Tekne

```
Tekne
  parametre : 3 ( Kisi*, double, MALZEME )
  Donus Turu : Nesne Adresi
Tekne
  parametre : 0
  Donus Turu : Nesne Adresi
MalzemeToString
  parametre : 0
  Donus Turu : string
operator<<
```

3. SONUÇ

Çalışma Düzenimde kağıt kullanımını ve olasılıklarla birlikte algoritma tasarımı geliştirmekte etkili olan bu ödev hataları nasıl çözmem gerektiğini, patlama noktalarını ve hata noktalarını bulmakta gelişim sağladı. Ayrıca Java dilini daha iyi anlamam ve bu dili kullanırken ki yaklaşımlarımı belirlememde yardımcı oldu.