T.C.
SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

PROGRAMLAMA DİLLERİNİN PRENSİPLERİ ÖDEV RAPORU

ÖDEV 2

G221210002 – Talha İRİS

2. Öğretim B Grubu

SAKARYA

Mayıs, 2024

Programlama Dillerinin Prensipleri Dersi

C Dilinde Nesne Yönelimli Paradigma

Özet

C dilinde struct yapısı, referans ve işaretçiler kullanırlarak nesne yönelimli paradigmayi gerçeklemek asıl temeldeki amaçtı. Alt başlıklarda ise kalıtım ve soyut sınıf(fonksiyon) yapısının gerçeklenmesi önemli yer kaplıyor. Uygulama kısmında ise m satır n sütun bir matriste yer alan konumları belli "Canli" üst sınıfından kalıtım alan nesneleri (0,0) konumundan (m,n) konumuna kadar savaştırarak ölen nesneleri "X" ile gösterip hayatta kalan nesneyi ekrana yazdıran bir program yazmak amacımızdı.

1. ÖĞRENDİKLERİM

- C dilinde nesne yönelimli paradigmayı gerçeklemek.
- Java dilinden C diline olabildiğince birebir kod dönüşümü gerçekleştirmek.
- Struct işaretçilerinin kullanımında dikkat edilmesi gerekenler.
- Soyut sınıf ve fonksiyonları kullanırken dikkat edilmesi gerekenler, kullanımın sebepleri ve sağladığı kolaylıklar.
- Bir diziyi nasıl tür bağımsız bir diziye çevirebileceğimiz ve bir dizinin nasıl 2 boyutlu bir diziye benzetilebileceği.

2. EKSİK BIRAKTIKLARIM

Ödevde istenilenler açısında herhangi bir konuda eksiğim olmamasına karşın yine de programımı nasıl daha verimli ve hızlı çalışır duruma getirebileceğimi düşündüm ama elde ettiğim tüm fikirler programımda bazı sabit kalmasını istediğim parametrelerin ve gerçeklemelerin değişmesine sebep olduğundan herhangi bir değişikliğe gidip hem kod okunabilirliğini azaltmasını hem de ödevde istenilenlerin değişmesini istemedim.

3. ZORLANDIKLARIM

- En başında C dilini nesne yönelimli paradigmaya çevirmek zaten başlı başına fazlasıyla karmaşık gelmişti. Ama yazım şeklini ve gereksinimlerini anladıkça artık yavaşça kolaylaşmaya başladı. Ayrıca kodumda sıkıntı çektiğim zamanlarda herhangi bir değişiklik yapacaksam bunları benim için fazlasıyla kolaylaştırdı.
- İlk başta 1. maddedeki kısmın zoryalacağını bildiğimden bu karmaşıklıktan ilk önce Java dilinde yazıp sonrasında bunu C diline çevirerek kurtulacağımı düşündüm. Beklediğim gibi de oldu ama düşündüğüm tamamı için uygulanabilir değilmiş.

- Java dilinin sağladığı kolaylıklardan birini kullanmak C dilinde yazarken aleyhime işledi. Bu da bir diziye “Canlı” üst sınıfından kalıtım alan nesneleri ekledikten sonra hepsinin “Canlı” üst sınıfının bir türevi olmasından kaynaklı Java böyle kullanmama izin verdi. Ama C diline geldiğimde karşıma gelen zorluk büyüdü. Çünkü eğer dizide bu alt sınıflardan oluşmuş nesneleri üstlerinde bulunan “Canlı” sınıfını kullanarak tutarsam sonrasında bu nesnelerin değişken ve fonksiyonlarına ulaşmak istediğimde sorun çıkartacağını fark ettim. Böylece “void*” kullanımını denemeye karar verdim. Buradaki sorun ise dizinin boyutunu belirleme ve içine eklenen nesneleri kullanmaya karar verdiğimizde kullanacağımız nesnenin ne oladuğunu çoktan biliyor olmamız gerektiğiydi. Bunları çözümü bende vardı. Mesela 2. Sorunla ilgili “belirtec” adında bir dizi kullanılarak burada da hangi indexte hangi nesnenin durduğunu anlayabilirdik. Ama bu sorunları hepsini çözecek bir fikir kullanmaya karar verdim. O da “struct pointer” diyebiliriz. Çünkü bir stuctın içinde hem verinin kendisini hem de verinin tipini tutabiliriz. Böylece fazladan dizi kullanımına ihtiyaç kalmaz ve kod okunabilirliği artırılabilir. Bu sorunu da böyle hallettim.
- İlk başta aklıma 2 boyutlu dizi kullanımı da geldi ama bir diziyi istediğim zaman tek boyutlu istediğim zaman 2 boyutlu kullanma fikrini reddetmek istemedim. Böylece bazı işlemleri tek for döngüsüyle bazılarını çift for döngüsüyle yaptım ama bu da bir 2 boyutlu dizinin kullanımına dair bilgilere sahip olmamı sağladı.

Zamanınızı ayırdığınız için teşekkür ederim.
-Talha İRİS