T.C.
SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

PROGRAMLAMA DİLLERİNİN PRENSİPLERİ 2. ÖDEV RAPORU

C Dilinde Nesne Yönelimli Yapı Benzetimi

G191210066 - Furkan İsmet Tufan

SAKARYA

Mayıs, 2021

Programlama Dillerinin Prensipleri Dersi

C Dilinde Nesne Yönelimli Yapı Benzetimi

Furkan İsmet Tufan[a]*, g191210066, 2.Öğretim C Grubu, furkan.tufan@ogr.sakarya.edu.tr

Özet

Ödevde istenen; c dilinde json dosyalarını parse edip, nesne yönelimliye benzetilecek yapılar yardımıyla emirleri uygulamak. Ödevi yaparken json dosyalarını okumak için oluşturulan ekstra kütüphaneleri kullanmadım. json dosyalarını txt dosyası okur gibi okudum. *strstr*, *memcpy* gibi fonksiyonlar yardımıyla dosyalardan istenen bilgileri aldım ve oluşturduğum *portfoy*, *hisse* ve *emir* yapılarına kaydettim. Sonrasında ise *banka* yapısındaki fonksiyonlarla emirleri yerine getirdim. Ekrana satışlardan elde edilen kar ve zararları, güncel portfoyu bastırdım. Son olarak, malloc ile oluşturduklarımın belleğini free() komutuyla işletim sistemine iade ettim.

Anahtar Kelimeler: c, struct, files

1. GELİŞTİRİLEN YAZILIM

1.1 Ödevin veriliş amacı

Ödevin veriliş amacı, birçok durumda işimize yarayan nesne yönelimli programlamanın c dilinde benzetimini öğrenmemiz ve kullanmamız olabilir.

1.2 Oluşturduğum yapılar

Okunacak dosyaların içindekileri kaydetmek için Emir, Hisse ve Portfoy yapıları oluşturdum. Bu yapıları, derste anlatıldığı şekilde nesne yönelimliye benzeterek yazdım. Banka yapısına ise, yukarıda bahsettiğim yapıların dizilerini özellik olarak ekledim. Ayrıca, bankayı parametre olarak alan ve banka üzerinden işlemleri yapan EmirleriUygula isimli bir fonksiyon ekledim.

1.3 Dosyaları okuma yöntemim

İlk olarak dosya göstericisi oluşturdum ve hisseler dosyasını *fopen* fonksiyonuyla açtım. Dosyayı satır satır okumaya başladım. Toplam hisse sayısını öğrenmek için içinde "_id" olan satır sayısını bulmaya çalıştım. *strstr* fonksiyonuyla, bulunulan satırın "_id" içerme durumunu kontrol ettim. Fonksiyon NULL döndürmediği sürece, başlangıç değeri 0 olan hisseSayisi değişkeninin değerini bir artırdım. Dosyadaki toplam hisse sayısını öğrendim ve malloc komutuyla hisseSayisi boyutunda Hisse dizisi oluşturdum. *fseek* yardımıyla dosyanın başına döndüm ve Hisse dizisine atamak için dosyadaki değerleri en baştan okumaya başladım.

* Ödev Sorumlusu. Furkan İsmet Tufan, g191210066,
Mail Adresi: furkan.tufan@ogr.sakarya.edu.tr

Id değeri, "_id" yazısının başındaki indisten 7 karakter sonra başlıyordu. Satırları parçalamak için bunun gibi bilgileri kullandım. Örneğin; "sembol" yazısının, bulunduğum satırdaki indisini *strstr* fonksiyonunun yardımıyla buldum ve 10 karakter sonrasındaki indisi ilkSutun isimli değişkene kaydettim. Sonrasında ilkSutun'da yazan indisten itibaren gördüğüm her karakteri çift tırnak sembolüyle karşılaştırdım. Çift tırnak sembolünü görmeden önceki son karakterin indisini de sonSutun değişkenine kaydettim. Son olarak memcpy fonksiyonunu kullanarak ilkSutun ve sonSutun arasındaki karakterleri çektim ve geciciSembol değişkenine atadım. Bu işlemleri id, ad ve fiyat için de yaptım. En sonunda elimdeki bu değerleri HisseOlustur fonksiyonuna parametre olarak yolladım ve önceden oluşturduğum Hisse dizisinin bir indexini oluşturmuş oldum. Bu işlemi her satır için tekrarladım ve Hisse dizimi doldurmuş oldum.

Yukarıda hisseler.json dosyası için detaylı olarak anlattığım işlemleri emirler.json ve portfoy.json için de yaptım. Okuma işlemlerinin sonunda elimde Hisseler dizisi, Portfoy dizisi ve Emirler dizisi oluştu. Açtığım her dosyayı, okuma işlemi bitince kapattım.

1.4 Emirleri yerine getirme

Banka yapısından bir banka oluşturdum ve elimdeki dizileri içerisine kaydettim. EmirleriUygula metodunu, oluşturduğum bankayı parametre olarak yazarak çağırdım.

Bu metot öncelikle emrin sembolünü hisseler dizisindeki ve portfoy dizisindeki sembollerle karşılaştırıyor ve alakalı yerden guncelFiyat, maliyet gibi bilgileri alıyor. Daha sonrasında emrin işlemini strcmp ile kontrol ediyor ve alış&satış durumuna göre ayrı işlemler yapıyor. Alış işleminde alakalı hisse portfoyde yoksa *realloc* ile portfoy dizisinin eleman sayısını artırıyor ve portfoy dizisine yeni elemanı ekliyor, portoyde varsa maliyet, adet gibi değerlerini güncelliyor. Satış durumunda kar&zarar durumunu hesaplayıp ekrana yazdırıyor. Bankanın emirler dizisindeki her emir için bu işlemler tekrarlanıyor. Son olarak istenen bilgiler ekrana yazdırılıyor ve dizi üyelerinin belleği YokEt metotlarıyla boşaltılıyor.

2. ÇIKTILAR

- *hisseler.json* dosyasında olmayan bir şey için alım emri varsa alım yapılmıyor.
- *portfoy.json* dosyasında olmayan ancak hisseler.json dosyasında olan bir hisse için alım emri varsa alım yapılıyor ve portfoy dizisine ekleniyor.
- Portfoy dizisinde olmayan bir şeyin satış emri olursa işlem gerçekleştirilmiyor.
- Bir hissenin portfoydeki miktarından fazla sayıda satış işlemi olursa satış yapılmıyor ve ekrana uyarı mesajı yazdırılıyor.

3. SONUÇ

Yaptığım bu çalışma sayesinde gelecekte c diliyle nesne yönelimliye benzer yapılar oluşturabilirim.

3.1 Öğrendiklerim:

- strstr ve memcpy ile dosya parçalayabilmeyi öğrendim.
- C dilinde, bool değişkenin benzerini oluşturmak için typedef kullanımını öğrendim.
- C dilinde struct ile nesne yönelimliye benzer yapılar oluşturmayı öğrendim.

3.2 Zorlandıklarım:

- C dilinde çoğu şey için hazır fonksiyon olmaması, ödevi planlarken düşündüğüm şeyleri yaparken zorlanmama sebep oldu. İlk defa C kullandım ve şu ana kadar kullandığım diller arasında en karmaşığı buydu.

Referanslar

[1] https://www.geeksforgeeks.org/

[2] https://stackoverflow.com/

[3] https://www.techonthenet.com/