T.C.
SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

PROGRAMLAMA DİLLERİNİN PRENSİPLERİ ÖDEV RAPORU

BORSADA EMİRLER SONRASI KAR/ZARAR VE GÜNCEL PORTFOY LİSTELEME

G191210303 - Barış YILMAZ

SAKARYA

Mayıs, 2021

Programlama Dillerinin Prensipleri Dersi

BORSADA EMİRLER SONRASI KAR/ZARAR VE GÜNCEL PORTFOY LİSTELEME

Barış YILMAZ[a]*

[a] *G191210303 2/C*

Özet

Ödevde C dilinde Struct ile nesne yönelimli benzetimi yapılarak borsadaki emirler sonrası satışı hisselerin sembollerinin ve kar zarar durumları ile emirler sonrası güncel portföyün listelenmesi istendi. Genel olarak JSON dosyalarını okuduktan sonra parse işlemi yaparak nesne listelerini oluşturarak ilerledim ve son olarak banka yapısında elde edilen nesneler ile alım-satım işlemlerini yaptım. Ardından ekrana çıktılarını verdim.

Anahtar Kelimeler: OOP, Struct, C, Dinamik, Json

1. GELİŞTİRİLEN YAZILIM

Öncelikle emirler, hisseler ve portföyleri okuyacağım JSON tipi dosyayı parse edebilmek için github üzerinden cJSON isimli kütüphaneyi indirdim. JSON dosyalarınının okuma ve parse fonksiyonlarına sahip olacak Dosya yapısı oluşturdum ve bu yapıda Hisse, Portfoy veya Emir tipinde nesne listeleri ile JSON parse işleminde lazım olacak bir root ve dosya okuma hatalarının oluşabileceğini öngördüğüm için bir de jumper oluşturdum. Kurucusuna ise parametre olarak dosya yolunu aldım. Dosya nesnesi oluşturulduktan sonra hisse, portföy, emir dosyalarının nesnelerinin Oku metodunu fonksiyon göstericisi aracılığıyla(oku) çağırarak dosya datalarını aldım ve ardından alım-satım işlemlerini yapacak olan oluşturduğum banka yapısının fonksiyon göstericisi yardımı ile alimSatim metoduna hisse, portföy ve emir dosya nesnelerini parametre olarak gönderdim. Banka yapısı içerisinde oluşturduğum hisseObj, emirObj ve portfoyObj dosya nesnelerine gelen parametre dosya nesnelerini atadım. Ardından bu dosya nesnelerinde bulunan parse metotlarını gerekli dosya için gerekli parse yöntemini(hisse için HisseParseEt gibi) fonksiyon göstericisi yardımı ile(hisse için hisseParse gibi) çağırarak orada Dosya yapısında bulunan ilgili nesne listesini dinamik olarak ilgili nesne sayısı kadar yani struct oluşturarak JSON dizisinde dönerek ilgili nesne dizisinde ilgili nesnenin kurucusunu çağırarak parametre olarak parse ettiğim bilgileri verdim. Hisse, Portfoy ve Emir yapılarında benzer yapılar oluşturdum ve sembol, varsa adet gibi değişkenler tuttum. Bu değişkenler içerisinde bulunan stringleri ise kurucuda parametre alınan stringlerin boyutu kadar dinamik olarak oluşturdum ve atamalarını yaptım. Yalnızca emirler için satışları ayrı olarak tutup listeleyeceğim için emir dosyasını parse ederken işlemi satış olanların sayısını da tuttum ve işlemlerin ardından emirlerdeki satış sayısı değerini dinamik olarak hafızada yer ayıracağım için tuttuğum banka yapısındaki satisSayisi değişkenine atadım. Bu satış sayısı yardımıyla satisSemboller ve satisKarZarar'ları adet kadar dinamik olarak hafızada oluşturdum. Ardından emir dosyasındaki obje sayısı kadar döngü oluşturdum ve döngüde işlemin satış veya alış olma durumuna göre if ekleyerek içeride portföy ve hisseler için ayrı ayrı sayıları

* Ödev Sorumlusu. Barış YILMAZ, G191210303,
Mail Adresi: baris.yilmaz5@ogr.sakarya.edu.tr

kadar başka bir döngüde kar ve zararları hesaplamak için gerekli olan bilgileri çekip gerekli işlemleri yaptım ve bu işlemleri yaparken satış için satış emri verilen hissenin portföyde bulunmaması veya bulunan adetten fazla olması durumunun kontrolü ile emir verilen adedin 0 veya daha küçük olma durumlarını, alım için ise alım emri verilen hissenin hisselerde bulunup bulunmadığının kontrolünü sağladım. Eğer alım emri verilen hisse portföyde bulunmuyorsa daha önce oluşturulan portföy nesnesi listesine yeni nesneyi ekleyemeyeceğim için bu durum için öncelikle portföy nesnesi listesini +1 adet realloc ettim ve yeni eklenen hisse bilgilerini parametre olarak kurucusuna verdim. Emirlerin döngüsü bittikten sonra ise banka yapısındaki satışların sembollerini ve kar-zararlarını tutan dizileri eğer önceden emirler parse edilirken satış olarak okundu fakat banka yapısı içerisinde satışı yapılacakken eklenememe durumu gibi durumlarda bu satış dizilerinin boyutunun listelenecek adetten fazla olmasını engellemek için indeks sayısından kontrolünü yaparak eğer böyle bir durumda varsa diziyi realloc ederek dinamik olarak küçülttüm. Sonrasında bankanın fonksiyon göstericisi yardımı ile toString metodunu çağırdım ve satış sembol ve kar-zarar dizilerini döngü yardımı ile dolaşarak satışları ve portföy nesnelerinin adedi kadar başka bir döngü ile de portföy nesnelerinin fonksiyon göstericisi yardımı ile toString metotlarını çağırarak güncel adedi 0 olmayan nesnelerin bilgilerini yazdırdım. Bu bilgileri yazdırırken banka ve portföy yapıları içerisinde fiyat bilgilerinde .'dan sonra maksimum 2 digit olacak şekilde ödev dosyasında örneği gösterilen gereksiz 0'ları silen SifirsizDondur fonksiyonlarını private benzetimi yaparak kaynak kodlarında oluşturdum. Bu fonksiyon içerisinde öncelikle herhangi bir tmp dizisi yardımı ile float'ın karakter sayısını alarak yalnızca ilk seferde malloc sonrasında, realloc ederek dinamik olarak karakter sayısı adedi kadar dinamik bellekte yerlerini ayırdım ve bu yapıdaki sifirsizSayi değişkenini döndürdüm. Bu dosya okuma işleminden yazdırma işlemine kadar olan tüm işlemleri try-catch yapısı benzetimi olan jumper koşulu içerisinde tuttum ve dosyanın bulunamaması veya bulunan dosyasının JSON dosyası olmaması durumlarını kontrol ettim. Son olarak ise oluşturduğum hisse, portföy, emir dosya nesneleri ve banka nesnelerinin fonksiyon göstericileri yardımı ile yıkıcı metotlarını(yokEt) çağırdım. Oluşturulan hisse, portföy, emir nesne listeleri ve bu listelerde bulunan her bir nesne dosya yıkıcısı içerisinde hangi nesnenin dosyası olduğunun kontrolü listenin NULL durumuna göre kontrol edilerek orada o nesnelerin yıkıcıları çağırılarak yok ediliyor.

2. ÇIKTILAR

Farklı birçok JSON dosyalarını deneyerek hazırladığım bu ödevde çıktılarda herhangi bir problem yaşamadım. Program düzgün bir şekilde çalıştı.

3. SONUÇ

C'de nesne yönelimli benzetimi, dinamik bellek yönetimi, JSON dosyalarını okuma ve parse etme hakkında tecrübeler edindim.

Referanslar

[1] https://github.com/DaveGamble/cJSON/blob/master/README.md

[2] https://riptutorial.com/c/example/27330/object-based-programming-using-structs