



T.C.
SAKARYA ÜNİVERSİTESİ

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

ŞANS OYUNU

G191210305 - Adem YILMAZ

SAKARYA

Nisan, 2022

Programlama Dillerinin Prensipleri Dersi

ŞANS OYUNU

Adem YILMAZ

G191210305

Özet

Ödevimiz şu şekildedir, Elimizde Kişiler.txt dosyası var ve, bu dosyada 1000 kişi var. Bu kişilere ait bilgilerde şu şekilde; adSoyad/toplamParası/yatırılanOran/şanslı sayısı. Elimizde bir de Sayılar.txt dosyası var burada da 1 ile 10 arasında sayılar mevcut. Her sayı bu dosyadan teker teker okutuluyor. O sayıyı tutturana kişi yatırdığı paranın 10 katını kazanıyor, tutturamayan ise yatırdığı parayı kaybediyor. Kazananların parası masadan ödeniyor, haliyle kaybedenlerin parası da masaya gidiyor. Toplam parası 1000 TL'nin altına düşen oyundan eleniyor. Herkes elenince de oyun sona eriyor. Bu problem ise, C dilinde Nesne Yönelimli Benzetim yapılarak çözülecektir.

Bu problemi Struct dizisi vasıtasıyla, dosyadan okuttuğum kişileri parçalayıp tek tek Kişiler adlı ve, tipi Kişi olan Struct dizisine atayarak çözdüm. Problemdaki asıl önemli kısımda burasıdır.

© 2022 Sakarya Üniversitesi.

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

Anahtar Kelimeler: struct,dizi,pointer,dosya okuma,diziye atama,sleep,tasarım

1. GELİŞTİRİLEN YAZILIM

Geliştirmiş olduğum yazılımda Dosya, Oyun, Kişi ve Test olmak üzere C dilinde Struct yapısından Nesne Yönelimli Benzetim yapılarak oluşturulmuş 3 tane sınıf mevcuttur.

Dosya sınıfı, txt uzantılı olarak verilen Kişiler ve Sayılar dosyasında ki verileri okuyup bunları ilgili diziye aktarmaktadır. Bu diziler ise Kişiler ve sayılar dizisi olmak üzere iki tanedir. Kişiler dizisi, Kişiler.txt dosyasındaki veriler bir kişiyi temsil ettiği için Kişi tipinde bir dinamik Struct dizisidir. Bu dizinin uzunluğu ise Kişiler.txt dosyasındaki satır sayısı bulunarak hesaplanmıştır. Struct dizisi malloc ile oluşturulduktan sonra, Kişiler.txt dosyasından her satır okunup ve bu satırdaki bilgiler parçalandıktan sonra kişiler dizisinin ilgili elemanına ait kişi bu bilgiler vasıtasıyla oluşturulur. Sayılar dizisi ise int pointer dizisidir. Fakat sayılar dizisinde okuma yapıp direkt bu pointer dizisine atama yapıldığında veriler de bozulma meydana geldiği için, önce bu şekilde int sayılar[toplamUzunluk] dizisi oluşturup dosyada ki verileri bu diziye attım sonra ise bu diziyi int pointer dizisine atadım. Bunu yapmamdaki amaç bu diziye Oyun sınıfından erişebilme isteğimden kaynaklıdır.

Oyun sınıfı, oyunda gerekli işlemlerin yapılıp ekrana gerekli çıktıları vermekle görevli olan sınıftır.

Kişi sınıfı ise bir kişiye ait verilerin tutulduğu sınıftır.

İçinde main metodu bulunan Test sınıfı ise Oyun tipinde bir nesne oluşturulup, oyunun başlatıldığı kısımdır.

Öncelikli olarak Test sınıfından Oyun nesnesi oluşturulur. Oyun sınıfı ise, Dosya sınıfından kalıtım aldığı için Dosya tipinde bir üyesi vardır ve oyun nesnesi oluşturulurken dosya nesnesi de oluşturulur. Dosya oluşturulduktan sonra dosyanın KişileriAktor ve SayılarıAktor adlı iki metodu çalıştırılır ve yukarıda bahsettiğim gibi ilgili dizilere ilgili atamalar yapılır. Oyun nesnesi oluşturulduktan sonra oyun başlatılır ve Sayılar dizisindeki elemanlar tek tek okutulur. Sayılar dizisi while ile okutulur, çünkü sayılar dizisinin tüm elemanları okutulduğunda başa sarıp tekrar okutmam gerek.

Sayılar dizisi okutulurken while döngüsü içinde bir for döngüsü oluşturulur ve kişiler dizisi okutulur. Bu for döngüsü içinde ise, kişiler dizisinin ilgili elemanındaki şanslı sayı bu sayıyla eşleşir ise kişinin yatırdığı paranın 10 katını kazanması, eşleşmez ise de yatırdığı parayı kaybetmesi, kazananların parasının kasadan ödenmesi, kaybedenlerin parasının kasaya gitmesi ve eğer kişinin parası 1000 TL den az ise oyundan elenmesi gibi durumlar kontrol edilip gerekli işlemler yapılır. Bu durumlar kontrol ettirildikten sonra ise en zengin kişi bulunur ve ekrana o turun en zengin kişisine ait bilgiler, tur sayısı, şanslı sayı ve masa bakiyesi gibi bilgiler yazdırılır. Oyun tüm kişiler elenene kadar devam eder.

Konsol ekranı tasarımı için ise, çok parametrelili yazdır methodu tanımladım ve, ekrana ait tasarımı boşlukları dikkate alarak bu methodun içinde oluşturdum. Her turda ekrandaki bilgilerin değişmesi için, zengin kişiyi bulup yazdır methodunu çağırdım ve bu methoda gerekli bilgileri gönderdim. Ekranda ki veriler değiştiğinde ekranda kayma olmasını engellemek için, bu bilgilerin tek tek uzunluğunu bularak bu engeli ortadan kaldırmaya çalıştım. Konsol her turda temizleniyor ve tekrar yazdırılıyor. Dolayısıyla konsol temizleme komutu ve sleep komutları kullanarak bu ekranı belli aralıklarla temizleyerek ve yineleyerek ekran tasarım problemini çözmeye çalıştım.

Son kısımda ise oyun yok edilmektedir. Oyunun yok edilmesi için öncelikle oyun oluştururken oluşturduğumuz dosya yı yok etmemiz gerekmektedir. Oyunu önceden yok edersek bu dosya çöp olarak heap bellek bölgesinde kalacaktır. Dosyayı yok ederken öncelikle dosya içinde oluşturduğumuz kişiler dizisindeki elemanları ilgili kişiye ait yoket metodunu çağırarak tek tek yok etmeliyiz ve sonrasında ise kişilerDizisini yok etmeliyiz. Sonrasında ise sayılar dizisini yok edip dosyayı yok edebiliriz. Dosya yı yok ettikten sonra oyun dosyasında oluşturduğumuz dosya sınıfında ki sayıları gösteren sayılar dizisini yok etmeliyiz. Son olarak da artık oyun sınıfını yok edebiliriz.

* ADEM YILMAZ, G191210305
adem.yilmaz10@ogr.sakarya.edu.tr

2. SONUÇ

Bu ödevdeki zorlandığım kısımlara gelicek olursam, ilk kısım Kişiler.txt dosyasında ki verileri parçalayıp bunları bir Struct dizisine atmak dı. İkinci kısım ise ekran tasarımımda her turda ekrandaki bilgiler değiştiğin de ekranda kaymaların önüne geçmekti.

Ödevi severek ve isteyerek yaptım. Dolayısıyla bana kattığı çok şey oldu. C dilinde ilk defa program yazdım ve bu yazdığım programı Nesne Yönelimli Benzetim yaparak yazmamın aşırı faydası oldu. Çünkü yüksek seviyeli dillerin neredeyse hepsi Nesne Yönelimli ve gerçek hayat projeleri ise Nesne Yönelimli Programlama baz alınarak yapılıyor. Bu nedenle bu ödev de Nesne Yönelimli Programlama yı daha iyi kavradım ve yüksek seviyeli dillerin önemini çok iyi anlamış oldum.

Referanslar

- [1] <https://stackoverflow.com/questions/19613752/how-to-properly-malloc-for-array-of-struct-in-c>
- [2] <https://www.tutorialspoint.com/explain-the-array-of-structures-in-c-language> [3] <https://www.javatpoint.com/java-regex>
- [4] <https://stackoverflow.com/questions/19823308/free-memory-for-array-of-structs>