



T.C.
SAKARYA ÜNİVERSİTESİ

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

ŞANS OYUNU

B201210101- YUNUS EMRE AKINCI

SAKARYA

Mayıs, 2022

Programlama Dillerinin Prensipleri Dersi

ŞANS OYUNU

Yunus Emre Akıncı

B201210101-1C

Özet

Ödevde C dilinde nesne yönelimli benzetimle birlikte verileri dosyadan çeken bir şans oyunu yapmamız isteniyordu. C dilinde NYP olmadığı için ödevi önce C# dilinde istenenlere göre yaptım. C# da oluşturduğum sınıfları C dilinde struct yapısı ile NYP'ye benzetmeye çalıştım.

© 2022 Sakarya Üniversitesi.

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

Anahtar Kelimeler: struct, pointer, NYP, bellek, dosya

1-) GELİŞTİRİLEN YAZILIM

1.1 Dosya.h ve Dosya.

Yer alan fonksiyonlar

- new_Dosya():Dosya ==>Bellekten dosya yapısı için yer tahsis eder
- kisileriDondur():Kisi* ==>Dosyadaki kişileri döndürür
- sayilariDondur():int* ==>Dosyadaki şanslı sayıları döndürür
- uzunlukBul():int ==>Parametre olarak verilen dosyanın kaç satır olduğunu hesaplar
- delete_Dosya():void ==>Bellekte kendisi için ayrılan alanı serbest bırakır

Oluşturulan struct yapısının kurucu methoduna parametre olarak Kisiler.txt ve Sayilar.txt 'nin klasör yolları verildi. kisileriDondur() fonksiyonunda while döngüsü ile dosya null değerini gösterene kadar satır satır okundu ve okunan bilgiler her satır satında bir kişi oluşturularak kişiler dizisine atandı. Benzer işlemler sayilariDondur() fonksiyonunda yapıldı ve böylece dosyadan veri çekme işlemleri tamamlandı.

1.2 Kisi.h ve Kisi.c

Yer alan fonksiyonlar

- new_Kisi():Kisi ==>Bellekten kişi yapısı için yer tahsis eder
- delete_Kisi():void ==>Bellekte kendisi için ayrılan alanı serbest bırakır

Kisi struct yapısında dosyadan gelen bilgileri tutacak değişkenler oluşturuldu.

⇒ isim:char* para:double oran:float sansliSayi:int

1.3 Oyun.h ve Oyun.c

Yer alan fonksiyonlar

- new_Oyun():Oyun ==>Bellekten oyun yapısı için yer tahsis eder
- oyna():void ==>Şans oyunun başlatan ve sürdüren fonksiyon
- delete_Oyun():void ==>Bellekte kendisi için ayrılan alanı serbest bırakır

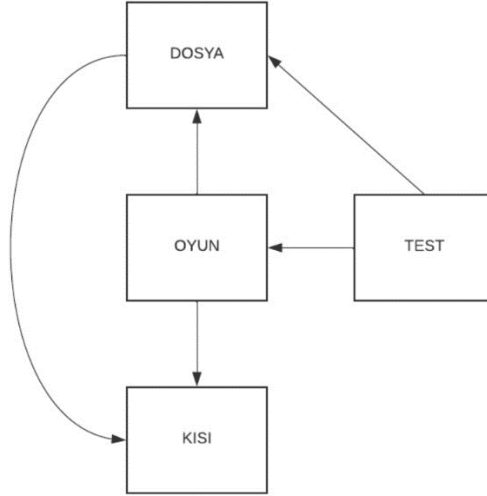
Yapının kurucu fonksiyonu parametre olarak Dosya yapısı verildi. Aldığı parametrenin fonksiyonlarını kullanarak oyun yapısının üye dizilerine (kisiler:Kisi* ve sayilar:int*) atama yapıldı. Bu diziler oyna() fonksiyonu içerisinde çağrılarak şans oyunu oynatıldı.

Oyunun mantığı: Her turda gelen şanslı sayı her kişinin şanslı sayısı ile tek tek karşılaştırılacak. Kişi sayıyı tutturursa yatırdığı paranın (para*oran) 10 katını kazanacak ve kazanılan para masa tarafından karşılanacak.

Tutturamazsa yatırdığı para masanın parasına eklenecek. Parası 1000'in altına düşen kişi elenecek. Oyun herkes elenene kadar devam edecek.

1.3 *Test.c*

main() fonksiyonunu burada oluşturuldu.



Kısaca yapılar arası ilişkiler

2-) ÇIKTILAR

```

#####
##          SANSLI SAYI: 8          ##
#####
##          TUR: 28                 ##
##          MASA PARASI: 41456561.24 ##
##          -----                ##
##          EN ZENGİN KİSİ         ##
##          Jennell Daniel         ##
##          BAKİYESİ: 214243.61    ##
##          -----                ##
#####

```

Oyun devam ederken (28. tur)

```

#####
##          TUR: 278                 ##
##          MASA PARASI: 48528976.56 ##
##          -----                ##
##          OYUN BİTTİ              ##
##          -----                ##
#####

```

Oyun sonu (Tüm kişilerin parası 1000'den az)

3-) SONUÇLAR

Bu ödev bana C dilini daha yakından tanıma fırsatı sağladı. C de C#, Java gibi dillerde olduğu gibi garbage collector olmadığı için belleği benim yönetmem gerekti ve bunun için free() fonksiyonu hakkında bilgi sahibi oldum. Aynı zamanda C de string olmadığı için char* ve char* dizileri hakkında araştırma yapma fırsatım oldu.

Ekran Görüntüsü: <https://www.youtube.com/watch?v=IEiCikI875M>

Referanslar

- [1] <https://www.muratoksuzer.com/2019/07/c-programlama-ile-txt-dosyayi-satir-satir-okuma.html>
- [2] <https://www.delftstack.com/howto/c/c-array-of-structs/#:~:text=There%20is%20another%20way%20to,memory%20with%20the%20specified%20size.>
- [3] <https://stackoverflow.com/questions/19068643/dynamic-memory-allocation-for-pointer-arrays>
- [4] <https://www.youtube.com/watch?v=yMHGt2Bcsts>