

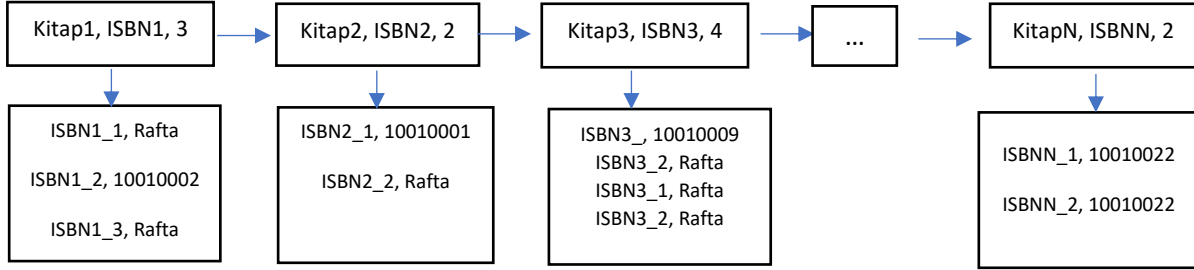
Son Teslim Tarihi: 2 Ocak 2023 PAZARTESİ 23.59 , Teslim: online.yildiz.edu.tr üzerinden olacaktır

NOT: Bu dosyada 8 Aralık 2022 tarihine kadar güncelleme yapılabilir. Güncel versiyonu Google classroom üzerinden takip ediniz. 8 Aralık 2022 tarihinde son versiyonu online.yildiz.edu.tr sistemine tanımlanacaktır.

Konu: Bu projede ilişkisel veri tabanlarının çalışma mantığına dayanan, küçük ölçekli ve dosya işlemleri tabanlı KÜTÜPHANE OTOMASYONU uygulaması gerçekleştirilecektir.

İşlevsel Gereksinimler:

- Yazar Ekleme/Silme/Düzenleme/Listeleme İşlemleri:** Her bir yazarın adı, soyadı, her yeni yazar kaydında otomatik artarak (1, 2, ..., N) verilecek yazarID bilgisi TEK YÖNLÜ LINKED LISTEDE yazarID değerine göre sıralı bir şekilde saklanmalı **yazarlar.csv** isimde dosyada tutulmalıdır. Yazarlar için kullanıcıdan alınıp dosyada saklanan bu bilgiler istenirse silinebilmeli veya düzeltilebilmelidir ve son hali aynı dosyada güncel olarak bulunmalıdır.
- Öğrenci Ekleme/Silme/Düzenleme/Listeleme İşlemleri:** Her bir öğrencinin adı, soyadı, başlangıçta 100 olan kütüphane puanı ve her bir öğrenci için benzersiz olan 8 haneli öğrenci numarası olmalıdır. Öğrencilere ait bilgiler dosyada ve uygulama içerisinde tanımlanacak struct yapısında ÇİFT YÖNLÜ LINKED LIST ve **Oğrenciler.csv** isimli dosyada saklanmalıdır. Saklanan bu bilgiler istenirse silinebilmeli veya düzeltilebilmelidir ve son hali aynı dosyada güncel olarak bulunmalıdır.
- Kitap Ekleme/Silme/Düzenleme/Listeleme İşlemleri:** Her bir kitabın adı, 13 Haneli ISBN numarası ve adet bilgisi olmalıdır. Bir kitaptan birden fazla sayıda örnek kütüphanede olabileceği için, kayıt oluşturma sırasında her bir örnek kitaba ayrıca benzersiz olan etiket numarası ISBN numarasına ISBN_1, ISBN_2, ISBN_N şeklinde otomatik eklenecek sayı ile verilmelidir. Kitaplar önce isimlerine sonra da ISBN numaralarına göre TEK YÖNLÜ LINKED LIST (**struct Kitap**) ile aşağıdaki gibi saklanmalıdır. Her kitabın örnek sayısı kadar kitabın bilgisi için ayıca TEK YÖNLÜ LINKED LIST (**struct KitapOrnek**) oluşturulmalı ve bu listelerde kitabın ETİKET numarası ile kitabın ödünç alınma durumu saklanmalıdır. Bu bağlamda, kitap ödünç alınmış ise hangi öğrenci tarafından ödünç alındı ise o öğrencinin ID numarası yer almalıdır. Eğer kitap ödünç alınmamışsa, bu alanda RAFTA bilgisi yazmalıdır. Her bir KİTAP ve bunların örnekleri arasındaki ilişki Şekil-1'de gösterildiği gibi TEK YÖNLÜ LINKED LIST'ler kullanılarak modellenmeli, ayrıca bu veriler uyumlu bir şekilde CSV türündeki dosyalarda saklanmalıdır. CSV dosyası; her bir örneğe ait verinin virgül (,) işareti ile ayrıldığı metin dosyasıdır. Örnek: Java Programlama, 1234567891011, 5



Şekil 1 Kitap ve Örnekleri Arasındaki İlişki

- Kitap – Yazar İlişkilendirme İşlemleri:** Bir kitabın birden çok yazar tarafından yazılma, bir yazarın da çok sayıda kitap yazabilme durumu dikkate alınarak, **KitapISBN - YazarID** eşleştirmesi yapılmalıdır. Bu eleştirme bilgileri **KitapYazar.csv** dosyasında Şekil-2'de gösterildiği gibi saklanmalıdır. Bu dosya üzerinde yapılacak okuma işlemlerinde kaydedilecek veriler için bir struct tasarlanmalıdır. Bu struct yoluyla dosyadaki satır sayısı (N) kadar yer açıldıktan sonra, **N elemanlı dinamik STRUCT DİZİSİNDE** veriler program açık kaldığı sürece saklanmalıdır. Şekil-2'de bulunan örneğe göre; 1234567891011 kodlu kitap 1, 2 ve 3 ID'ye sahip yazarlar tarafından yazılmış olup, 1234567891012 kodlu kitabı ise sadece 1 ID'ye sahip yazar yazmıştır. Örnekten de anlaşıldığı gibi bir yazar çok sayıda kitap yazabilirken, bir kitap çok sayıda yazar tarafından yazılabilir.

NOT1: Madde-1 ile girişi yapılan yazar bilgisi sistemden silindiğinde aşağıdaki dosyada güncelleme yapılmalıdır. İlgili yazar artık olmayacağı için bu alanda -1 yazmalıdır.

NOT2: Kitap – Yazar eşleştirme işlemlerinde kayıtlı olmayan bir Yazar veya kitap için işlem yapılmamalı ve uyarı vermelidir. Veri tutarlılığı adına gerekli tüm kontroller yapılmalıdır.

1234567891011, 1	1234567891011, -1
1234567891011, 2	1234567891011, 2
1234567891011, 3	1234567891011, 3
1234567891012, 1	1234567891012, -1
1234567891013, 4	1234567891013, 4
1234567891014, 4	1234567891014, 4
1234567891014, 1	1234567891014, -1

Şekil 2 Kitap Yazar Eşleştirmesi (Sol: Yazar silinmeden önce, Sağ: Yazar silindikten sonra)

- 5- **Öğrenci Kitap Ödünç Alma İşlemleri:** Bir öğrenci çok sayıda kitabı ödünç alabilirken, bir kitap farklı zamanlarda çok sayıda öğrenci tarafından da ödünç alınabilir. Bu nedenle bu bilgilerin tutarlı bir şekilde saklanması gerekmektedir. Her bir örnek kitabın ödünç alınma ve teslim edilme durumlarını saklayan bir CSV dosyası ve buna uyumlu olacak bir struct tasarlayınız. Bu veri yapısında ÖğrenciID, KitapEtiketNO, İşlem Türü ve Tarih Bilgisi Şekil-3 ile gösterildiği gibi saklanmalıdır. İşlem türü bilgisi; **kitabın ÖDÜNÇ ALINMASI – 0** ya da **kitabın TESLİM EDİLMESİ – 1** olarak kodlanmalıdır. Bununla birlikte kitap ödünç alındığında Şekil-1 ve Madde-3 ile izah edildiği gibi, bir kitabın Raf Durum Bilgisi, **Rafta ya da ÖğrenciNo** olarak ilgili dosyalarda ve veri yapısında güncellenmelidir.

NOT1: Kitap ödünç alma ve teslim işlemlerinde kayıtlı olmayan bir öğrenci veya kitap için işlem yapılmamalı ve uyarı vermelidir. Veri tutarlılığı adına gerekli kontroller yapılmalıdır. Ayrıca, bir öğrencinin Puan Bilgisi negatif durumda olursa kitap verilemez uyarısı ile işlem iptal edilmelidir.

NOT2: Kitap teslimi kitabın ödünç alındığı tarihten 15 günden sonra yapılırsa ilgili öğrenciye -10 ceza puanı verilmeli ve bu bilgiler ilgili dosya ve veri yapısında güncellenmelidir.

NOT3: Kitap ödünç alma işlemleri sırasında, bir kitaba ait tüm örnekler başka öğrenciler tarafından ödünç alınmışsa İŞLEM BAŞARISIZ uyarısı verilmelidir.

```
1234567891011_1,10011088,0,12.10.2022
1234567891018_2,10011048,0,12.10.2022
1234567891011_1,10011088,1,22.10.2022
1234567891012_1,10011048,1,27.10.2022
1234567891018_2,10011088,0,28.10.2022
```

Şekil 3 Kitap Ödünç Alma İşlemleri

PROGRAMDA İSTENEN MENÜLER

ÖĞRENCİ İŞLEMLERİ:

- 1- **Öğrenci Ekle, Sil, Güncelle:** Öğrenciye ait tüm bilgilerin kullanıcıdan alınıp, dosyalar ve linkli listeler üzerinde **Ekle/Sil/Güncelle** işlemlerinin ayrı ayrı gerçekleştirilmesi gerekmektedir.
- 2- **Öğrenci Bilgisi Görüntüleme:** ID bilgisi veya Ad-Soyad bilgisi alınan öğrencinin kişisel bilgileri (Ad, Soyad, ID, Puan vb) ve tüm kitap hareketleri listelenmelidir.
- 3- **Kitap Teslim Etmemiş Öğrencileri Listele:**
- 4- **Cezalı Öğrencileri Listele**
- 5- **Tüm Öğrencileri Listele**
- 6- **Kitap Ödünç Al/Teslim Et**

KİTAP İŞLEMLERİ:

- 7- **Kitap Ekle, Sil, Güncelle:** Kitaplara ait tüm bilgilerin kullanıcıdan alınıp, dosyalar ve linkli listeler üzerinde **Ekle/Sil/Güncelle** işlemlerinin ayrı ayrı gerçekleştirilmesi gerekmektedir.
- 8- **Kitap Bilgisi Görüntüleme:** Kullanıcıdan alınan Kitap Adı bilgisine göre her bir kitabın ve bu kitabın örnek kopyalarına **ait tüm bilgilerin listelenmesi** işlemleri yapılmalıdır.
- 9- **Raftaki Kitapları Listele:**
- 10- **Zamanında Teslim Edilmeyen Kitapları Listele:**
- 11- **Kitap-Yazar Eşleştir :** İlgili dosya ve struct dizisi üzerinde güncelleme yapılmalıdır
- 12- **Kitabın Yazarını Güncelle:** İlgili dosya ve struct dizisi üzerinde güncelleme yapılmalıdır

YAZAR İŞLEMLERİ:

- 13- **Yazar Ekle, Sil, Güncelle:** Yazarlara ait tüm bilgilerin kullanıcıdan alınıp, dosyalar ve linkli listeler üzerinde **Ekle/Sil/Güncelle** işlemlerinin ayrı ayrı gerçekleştirilmesi gerekmektedir.
- 14- **Yazar Bilgisi Görüntüleme:** Kullanıcıdan alınan Yazar Adı bilgisine göre her bir yazarın bilgilerinin ve bu yazara ait kitaplara **ait tüm bilgilerin listelenmesi** işlemleri yapılmalıdır.

Kodlama Gereklilikleri:

- ❖ **KULLANMANIZ GEREKEN STRUCT YAPILARI ŞU ŞEKİLDE OLMALIDIR:** Bu yapılardaki üye değişkenlerin değiştirilmesi yasaktır. Ayrıca bunlardan başka struct yapıları EKLENMEYECEKTİR.
 - `struct Ogrenci { char ogrID[8], char ad[30], char soyad[30], int puan, struct Ogrenci *next, struct Ogrenci *prev}`
 - `struct Yazar { int yazarID, char yazarAd[30], char yazarSoyad[30], struct Yazar *next}`
 - `struct KitapOrnek { char EtiketNo[20], char Durum[8], struct KitapOrnek *next }`
 - `struct Kitap {char kitapAdı[30], char ISBN[13], struct Kitap *next, struct KitapOrnek *head }`
 - `struct KitapYazar{ char ISBN[13], int YazarID}`
 - `struct KitapOdunc{ char EtiketNo[20], char ogrID[8], byte islemTipi, struct Tarih islemTarihi}`
- ❖ Yukarıdaki maddelerde verilen bir menü ve her bir işlem için gerekli fonksiyonları tasarlayınız. **Benzer mantıkla çalışan fonksiyonlarda tekrardan kaçınmak için fonksiyon pointer kullanınız.**
- ❖ Yapılması gereken tüm ekle, sil ve güncelleme işlemler için fonksiyon tanımlamanız gerekmektedir.
- ❖ Ekle, Sil, Güncelle gibi veri üzerinde değişiklik yapılan işlemleri ilgili veri yapısında yaptıktan sonra dosyaya mutlaka kaydediniz.
- ❖ Programlarınızda static ve global değişken kullanımı yasaktır.
- ❖ Her türlü bellek tahsisi, dinamik bellek yönetimi fonksiyonları ile yapılmalıdır.
- ❖ Yukarıda detayları verilen her bir dosyadaki nesne için uygun structure tanımlamalarını yapınız.
- ❖ Oluşturulacak dosyalar CSV dosyası formatında olmalıdır.

Projenin Raporlanması:

- ❖ Raporunuzda, yazdığınız her bir fonksiyonu test etmek için yeterli sayıda öğrenci ve ders bilgisi girerek yazdığınız fonksiyonların doğru çalıştığını gösteriniz.
- ❖ Bu amaçla programı çalıştırdığınızda ekrana çıkan ve kullanıcının ekran üzerinden giriş yaptığı her şeyi komut satırından kopyalayıp raporunuzu oluşturacak PDF dosyasına ekleyiniz.
- ❖ Program sona erdiğinde verilerin saklandığı tüm dosyaların içeriklerini de raporunuza ekleyiniz.
- ❖ Yararlandığınız kaynakları ve bu projenin size kazandırdıklarını raporun sonuna ekleyiniz

TESLİM EDİLECEKLER:

ÖĞRENCİNO.RAR/ZIP Dosyası içinde: Örnek: 10011089.rar veya 10011089.zip

- *** ÖğrenciNo.pdf rapor dosyası,
- *** ÖğrenciNo.c kaynak kodu ve
- *** Test koşumu sonucunda ortaya çıkan tüm .CSV formatlı

dosyalar olmalıdır.

NOT: Raporda geçen dönemki Sayısal Analiz, Yapısal Programlamaya Giriş derslerindeki raporlara benzer biçimde içindekiler tablosu, iletişim bilgileri vb detaylar dahil tüm çıktıları içermelidir.

Tüm ödevler benzerlik denetim programından geçirilecek, benzer olduğu anlaşılan ödevler kopya olarak değerlendirilecek ve ilgili her ödevde 0 (sıfır) notu verilecektir.

NOT: Projenizi son teslim tarihine kadar online.yildiz.edu.tr üzerinden yükleyiniz. Sistem otomatik kapanacağı için geç teslimler mail vb yoluyla mümkün değildir. Projenin yüklenmesinin başarılı olduğunu gösteren ekran görüntüsünü kanıt olarak kaydediniz. Kanıt olmaksızın sonradan e-posta yoluyla gönderilen ödevler kabul edilmeyecektir.

KONTROL İŞLEMLERİ ve SORULARINIZ: Proje teslim sürecinin sonuna kadar sorularınızı dersin laboratuvar grup sorumlusuna sorabilirsiniz. Teslim süreci sonrasında grup sorumlusu yüz yüze veya online kontrol için sizinle iletişim kurmak isteyebilir. Güncel mail ve telefon numarası bilginizi raporunuza mutlaka ekleyiniz.

Başarılar Dileriz.