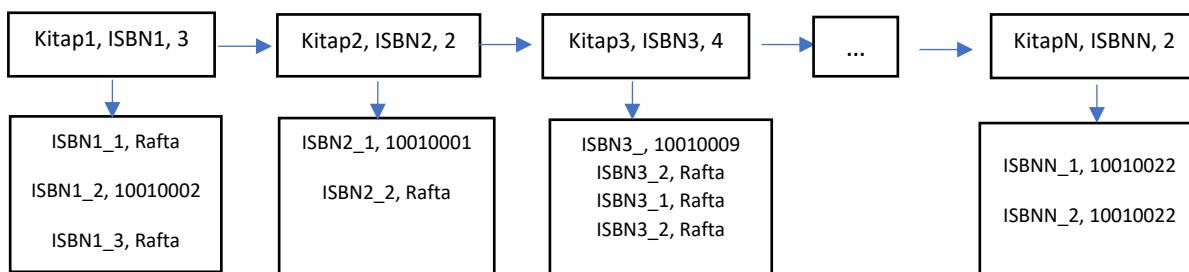


Son Teslim Tarihi: 27 MAYIS 2025 23.59 , Teslim: online.yildiz.edu.tr üzerinden olacaktır

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 olarak (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 **Ogrenciler.csv isimli dosyada** ve uygulama içerisinde tanımlanacak struct yapısında ÇIFT YÖNLÜ LINKED LIST şeklinde tutulmalıdır. 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 kitabı 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 ile aşağıdaki gibi saklanmalıdır. Her kitabı örnek sayısı kadar kitabı bilgisi için ayıca TEK YÖNLÜ LINKED LIST oluşturulmalı ve bu listelerde kitabı ETİKET numarası ile kitabı ödünç alınma durumu saklanmalıdır. Bu bağlamda, kitabı ödünç alınmış ise hangi öğrenci tarafından ödünç alındı ise o öğrencinin ID numarası yer almıştır. Eğer kitabı 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 (,) işaretleri ile ayrıldığı metin dosyasıdır. Örnek: Java Programlama, 1234567891011, 5



Şekil 1 Kitap ve Örnekleri Arasındaki İlişki

- Kitap – Yazar İlişkilendirme İşlemleri:** Bir kitabı birden çok yazar tarafından yazılma, bir yazarın da çok sayıda kitabı yazabilme durumu dikkate alınarak, **KitapISBN - YazarID** eşleştirme yapılması yapılmalıdır. Bu eşleştirme bilgileri **KitapYazar.csv dosyasında** Şekil-2'de gösterildiği gibi saklanmalıdır. Bu dosya üzerinde yapılacak okuma işlemlerinde kaydedilecek veriler için bir struct tasarılanmalıdır. 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 kitabı 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 kitabı yazabilirken, bir kitabı 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 kitabı için işlem yapılmamalı ve uyarı vermelidir. Veri tutarlılığı adına gerekli tüm kontroller yapılmalıdır.

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

Şekil 2 Kitap Yazar Eşleştirme (Sol: Yazar silinmeden önce, Sağ: Yazar silindiğinden 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 tasarılayınız. 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, **Rafya ya da ÖğrenciNo** olarak ilgili dosyalarda ve veri yapısında güncellemelidir.

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 yapılrsa 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:

- ❖ 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 tahsis, 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 çalıştırığınızda ekrana çıkan ve kullanıcının ekran üzerinden giriş yaptığı her şeyi komut satırından kopyalayıp raporunuza 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

PROJE TESLİMİ HAKKINDA:

1. Uygulamanızın çalışmasını kısa bir video (en fazla 8 dk.) çekerek video paylaşım sitesine yükleyiniz. Anlatmak istediğiniz detayları da seslendirebilirsınız.
2. Yaptığınız çalışmayı, problem tanımını, gerçekleştirdiğiniz çözüme ait ekran görüntülerini, algoritmanıza ait detayları içeren raporu ve **video linkini** ÖğrenciNo.pdf formatında saklayınız.
3. **Önemli Not:** Video çekiminin başında görüntülü olarak kendinizi tanıtmayı **UNUTMAYINIZ :)**
4. **Aşağıdaki şablonca uygun şekilde proje tesliminizi yapınız:**

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

*** **ÖğrenciNo.pdf** rapor dosyası,

*** **ÖğrenciNo.c** kaynak kodu ve

*** Test koşumu sonucunda ortaya çıkan tüm **.CSV** formatlı dosyalar.

NOT: Raporda geçen dönemki Sayısal Analiz, Yapısal Programlamaya Giriş derslerindeki raporlara benzer biçimde içindeler 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 verelecektir.

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.

Başarılılar Dileriz.

NOT: Sadece zip veya rar formatlı dosya yüklenecektir, google drive, one drive, github vb linkler kabul de ildir.