YAPISAL PROGRAMLAMA DÖNEM PROJESİ

Son Teslim Tarihi: 7 Ocak 2022 Cuma 17.00

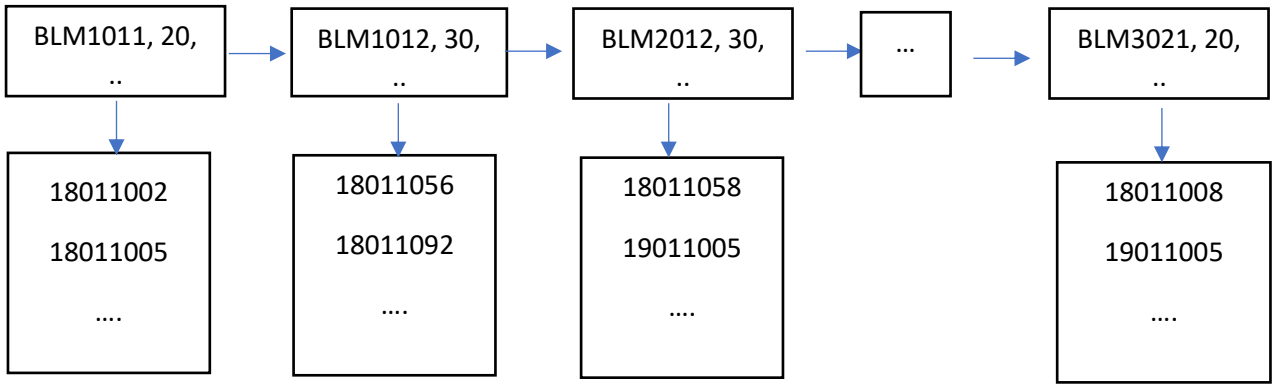
Konu: Bu projede ilişkisel veri tabanlarının mantığına dayanan, küçük ölçekli ve dosya işlemleri tabanlı ders kayıt uygulaması gerçekleştirilecektir.

İşlevsel Gereksinimler:

1- Açılan dersler dosyasında her bir dersin benzersiz olan ders kodu, adı, toplam kredisi, kontenjanı bilgisi vardır. Bu dosyadan okunacak veriler, Şekil-1'de gösterildiği gibi, TEK YÖNLÜ LİNKLİ LİSTE ile ders koduna göre sıralı bir şekilde saklanmalıdır. Her bir linkli liste düğümünde dersin kodu, adı, kapasitesi ve kredisi bilgisi olmalıdır. Ayrıca her düğümde, o derse kayıtlı öğrencilerin numarası dinamik bir dizi ile öğrenci numarasına göre artan sırada saklanmalıdır.

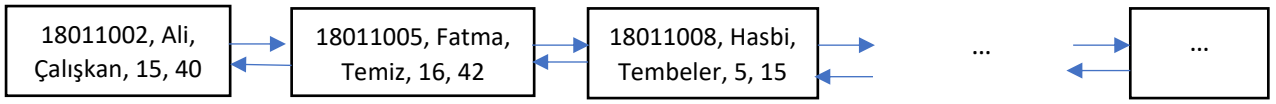
2- Dersler üzerine yapılacak olan yeni ders açma ve ders kapatma işlemleri Şekil-1'de verilen linkli liste üzerinde yapılmalıdır. Bu işlemler yapıldıktan sonra, yazacağınız dosyayaKaydet fonksiyonu yardımıyla, linkli listedeki veriler dersler isimli dosyaya kaydedilmelidir.

3- Ders açma sırasında aynı kodlu ders daha önceden açılmışsa bu işleme izin verilmemelidir, aynı şekilde kapatılmak istenen dersin kodu kullanıcıdan alındığında daha önceden açılmamış bir dersin silinemeyeceği uyarısı verilmelidir. Kapatılan bir derse kayıtlı öğrencilerin durum bilgisi 4. Maddede açıklanan ÖğrenciDersKayıt isimli dosyada DERS_KAPANDI olarak güncellenmelidir.



Şekil 1. Tek yönlü linkli liste

4- Öğrenciler dosyasında, her bir öğrencinin benzersiz olan öğrenci numarası, adı, soyadı, kaydolduğu toplam ders sayısı ve kaydolduğu toplam kredi bilgisi bulunmalıdır. Bu bilgileri karşılayan bir struct değişkeni tasarlayınız ve ÖğrenciListesi isimli ÇİFT YÖNLÜ LİNKLİ LİSTEDE (bkz. Şekil 2) öğrenci bilgilerini numaralarına göre artan düzende saklayınız. Öğrenci ekle, sil işlemleri öncelikle bu linkli listede yapılmalı sonra ilgili dosyaya veriler kaydedilmelidir.



Şekil 2. Çift yönlü linkli liste

5- Bir öğrenci çok sayıda ders alabilir, aynı zamanda bir dersi çok sayıda öğrenci de alabileceği için ÖğrenciDersKayıt isimli ayrı bir dosyada hangi öğrencinin hangi dersi aldığı <öğrenciNO-DersKodu> eşleştirmesi yöntemiyle saklanmalıdır. Her bir eşleştirme için, 1'den başlayarak otomatik artan ayrıca bir benzersiz ID bu dosyada vardır. Bu eşleştirmede ek bilgi olarak, derse kaydolma tarihi ve öğrencinin derse kayıt durumu bilgisi (KAYITLI, DERS_KAPANDI, BIRAKTI) saklanmalıdır. Örnek dosya formatı: **ID, DERS_KODU, ÖĞRENCİ_NO, TARİH, DURUM.**

6- Öğrenciler derse kaydolurken, ders açılması sırasında kullanıcıdan alınan kayıt yapılacak maksimum ders sayısı, maksimum kredi değeri kriterlerine göre ders ekleme yapabilmelidir. Bu değerlere ulaşınca ders ekleme işlemi engellenmelidir. Bunun dışında bir dersin kontenjanı dolu ise öğrencinin derse kaydolması engellenmelidir.

7- Öğrenci(ler) derse kayıt yaptığında veya bir dersi sildirdiğinde öncelikle Şekil-1'de verilen linkli liste üzerinde ekleme ve silme işlemleri yapılmalıdır. Ardından Şekil-2'deki çift yönlü linkli liste de güncellenmelidir. Bu işlem(ler) tamamlandıktan sonra ÖğrenciDersKayıt isimli dosyanın içeriği de güncellenmelidir.

8- Ders kodu verilen bir derse kayıtlı olan tüm öğrencileri listeleyen fonksiyonu yazınız. Bu fonksiyon yardımı ile tüm derslere ait sınıf listesini DERSKODU.txt isim formatında dosyaya yazdırınız.
9- Öğrenci numarası verilen bir öğrencinin aldığı tüm dersleri listeleyen ve ÖĞRENCİNO_DERSPROGRAMI.txt isminde dosyaya yazan fonksiyonu yazınız.

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.
- Öğrenci ve Ders için ekleme/silme işlemlerini yapınız, bu işlemleri önce 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 text dosya 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 Word 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 Dosyası içinde ÖğrenciNo.docx rapor dosyası, ÖğrenciNo.c kaynak kodu ve test koşumu sonucunda ortaya çıkan tüm dosyalar olmalıdır.

Tüm ödevler benzerlik denetim programından geçirilecek, benzer olduğu anlaşılan ödevler kopya olarak değerlendirilecek ve ilgili her ödeve 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 delil olarak kaydediniz. Delil olmaksızın e-posta yoluyla gönderilen ödevler kabul edilmeyecektir.

Başarılar Dileriz.