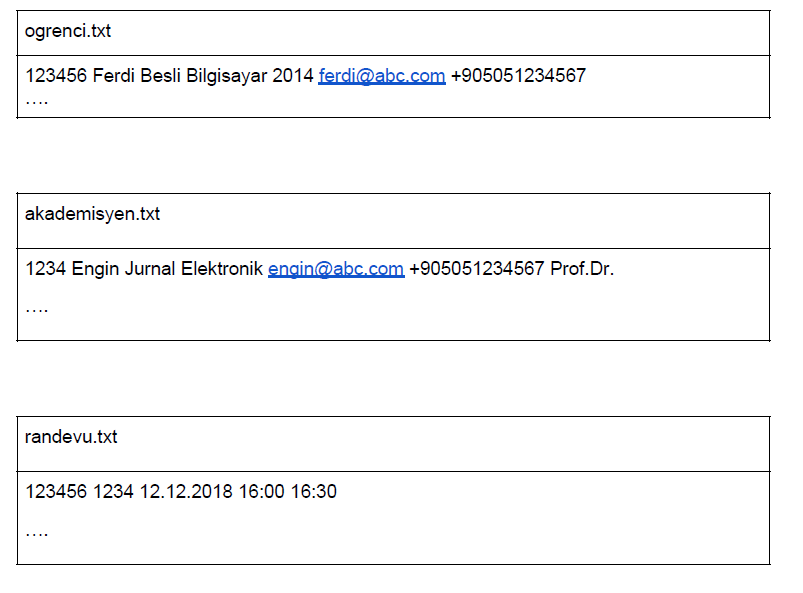
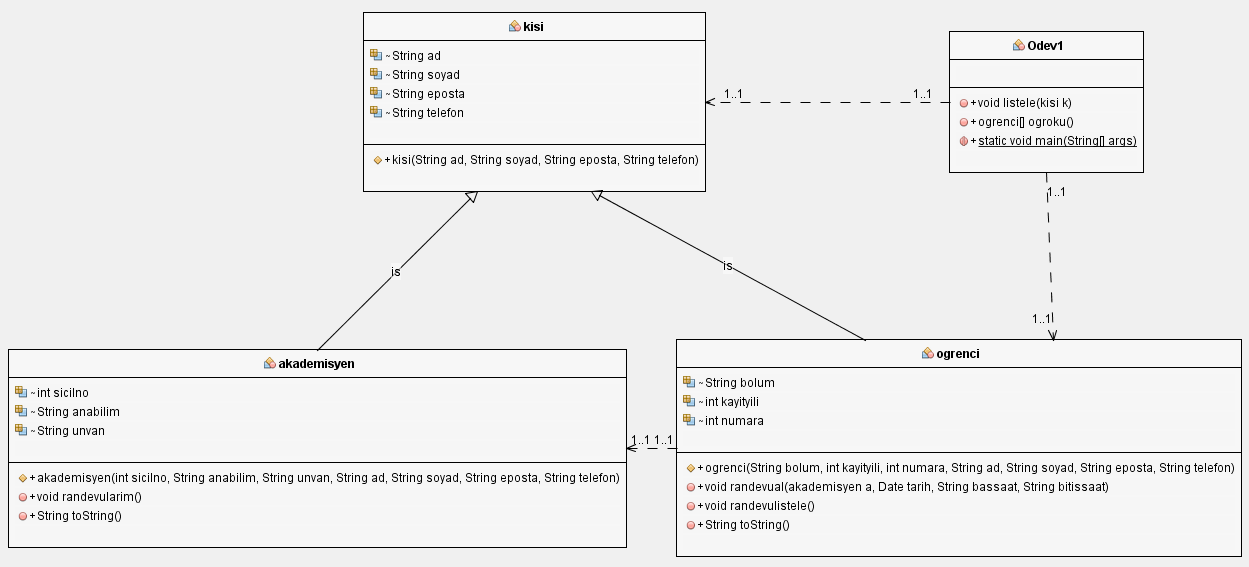
**Soru:**

Sizden bir üniversitede öğrencilerin akademisyenlerden randevu almasını sağlayan bir sistem geliştirmeniz istenmektedir. Öğrencinin numara, ad, soyad, bölüm, kayıt yılı, e-posta, telefon bilgileri tutulacaktır. Akademisyenin sicil numarasi, ad, soyad, anabilim dalı, e-posta, telefon, ünvan bilgileri tutulacaktır. Randevunun da öğrenci, akademisyen, tarih, başlangıç saati, bitiş saati bilgileri olmalıdır. Program başlangıcında ogrenci, akademisyen ve randevu bilgileri dosya[lar] dan okunmali ve program çıkışında dosya[lar] a kaydedilmelidir. Dosya örnekleri aşağıda açıklanmaktadır. Öğrenci numara bilgisi öğrenci için, sicil numarasi bilgisi akademisyen için benzersiz bir bilgidir. Tasarlanan bir konsol ekran menüsü sayesinde, sırasıyla öğrenci, akademisyen ve randevu ekleme ve listeleme işlemleri yapılabilmelidir. Randevu için hem öğrencinin hem de akademisyenin randevuları sorgulanmalı ve ayni kisi[ler]e ait takvim çakışması engellenmelidir. Ayrıca öğrenci ve akademisyenlerin randevu programı ekrana çıktı olarak verebilmelidir. Sınıf tasarımları yapılırken miras ve kompozisyon mekanizmaları kullanılmalı, nesne yönelimli programlama prensipleri ihmal edilmemelidir.

Örnek dosya içerikleri aşağıdaki gibi olmalıdır:





Buna göre;

Yukarıda UML diyagramı verilen sınıfları oluşturarak gerekli kodlamaları yapınız?

* Öğrenci sınıfındaki randevual metodu ilgili öğrenci için parametre olarak gelen akademisyenden randevu ilgili tarih ve saatte öğrenci veya akademisyenin randevusu yok ise randevu alacak ve randevu.txt dosyasına yazacaktır.
* Öğrenci sınıfındaki Randevulistele() ilgili öğrencinin randevularını gösterecektir.
* Hem öğrencihem de akademisyen sınıfındaki toString() metodu öğrenci bilgilerini string olarak döndürecektir.
* Akademisyen sınıfındaki randevularım() metodu akademisyenin randevularını listeler.
* Odev sınıfı çalıştırılabilir psvm sahibi sınıftır. Buradaki listele metoduna ogrenci gönderilirse öğrenci.txt dosyası akademisyen tipinden bir nesne gönderilirse akademisyen.txt dosyası okunarak listelenecektir.
* Ogroku() metodu ogrenci.txt dosyasındaki öğrenci bilgilerini bir dizi olarak döndürmelidir.
* Tasarlanacak bütün sınıflar odev1.java dosyasında olmalıdır.
* Kurucu metotlar ilgili öğrenci/akademisyen için girilen bilgileri ilgili dosyalara kayıt etmelidir.

ÖDEV gönderimi: Ödev dosyasında çalışır kodlar ve bir Word dokumanı olarak rapor eklenmelidir. Her bir sınıf için kodlarınız ve metotların çalıştırma sonuçları bu Word dokümanında gösterilmelidir. Bütün ödevler turnitin intihal programında depo özelliği ile kontrol edilecektir. Kopya ödevler sıfır olarak değerlendirilecektir.

S0N gönderim tarihi: 15. 04.2020 23:59