

**BURSA TEKNİK ÜNİVERSİTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ**  
**BLM0121 NESNEYE YÖNELİK PROGRAMLAMA DERSİ**  
**PROJE ÖDEVİ**  
**2023**

1. Ödevin süresi 9 gündür. (05.05.2023, 23:59)
2. Ödevler e-kampüs'teki ilgili alana yüklenecektir.
3. Gecikmeler ve mailler kabul edilmeyecektir.
4. Yazılan kodlarda yorum satırları olmalı, işlemler açıklanmalıdır.
5. Proje için en az 3 sayfalık projede ne yapıldığını ve eksikleri anlatan bir rapor hazırlanmalıdır. (Sadece kod bloklarını yapıştırmayınız.)
6. Uygulamanın çalışır hali yüklenmelidir.
7. Raporun adına ve ilk sayfasında öğrencilerin isim, soy isim ve öğrenci numarası bilgileri yer almalıdır.
8. Proje raporunuzun ve kodlarınızın yer aldığı dosyalar, ad\_soyad\_ogrencino olarak isimlendirilen bir klasör içerisinde zip/rar şeklinde teslim edilmelidir.
9. Kopya çekildiğinin tespit edilmesi halinde ilgili ödev değerlendirilmeyecektir.

**Yukarıda yer alan 9 maddenin hepsine uyulmalıdır. Kodu pdf halinde gönderenlerin projeleri değerlendirmeye alınmayacaktır.**

Ödev ile ilgili sorularınız için [ceren.akman@btu.edu.tr](mailto:ceren.akman@btu.edu.tr), [zeynep.barut@btu.edu.tr](mailto:zeynep.barut@btu.edu.tr), [hasibe.candan@btu.edu.tr](mailto:hasibe.candan@btu.edu.tr) mail adreslerine ulaşabilirsiniz.

*Başarılar dileriz...*

## PROJENİN AMACI

Bu projede Nesne Yönelimli Programlama dersi kapsamında öğrendiğiniz bilgileri kullanarak bir banka uygulaması geliştirmeniz beklenmektedir.

Programı aşağıda belirtilen şartları yerine getirecek şekilde tasarlayıp kodlayınız.

## YAPILMASI İSTENENLER

Aşağıda yer alan UML diyagramında tüm sınıfların sahip olabileceği özellikler ve parametreleri verilmiştir.

- Kisi sınıfından BankaPersonel ve Musteri sınıfları miras almaktadır. Kişi sınıfında banka personeli ve müşterilerin sahip olabileceği ortak özellikler bulunmaktadır.
- Banka personelinin temsilcisi olduğu müşterileri saklamak için müşteriler isimli bir ArrayList yer almaktadır. Bu ArrayList içerisinde ilgili müşteri nesneleri saklanacaktır.
- Müşteri sınıfında, hesaplar ve krediKartlari isimli ArrayList'ler yer almaktadır. Bu ArrayList'lerde yine ilgili sınıfların nesneleri saklanacaktır.
- Musteri sınıfındaki hesapEkle isimli metot, müşterinin açmak istediği hesap türüne göre (Vadeli hesap, Vadesiz hesap veya yatırım hesabı) BankaHesap sınıfının altındaki hesapTuru isimli değişkenden elde edilecektir. İlgili sınıftan nesneler oluşturulup hesaplar isimli ArrayList'te saklanacaktır.
- Musteri sınıfındaki hesapSil metodu öncelikle hesabın bakiyesini kontrol etmeli, eğer hesap bakiyesi 0 dan büyükse "lütfen öncelikle bakiyenizi başka bir hesaba aktarınız" uyarısını göstermelidir. Eğer 0'a eşitse hesabı silmelidir
- KrediKarti sınıfındaki krediKartiEkle isimli metot, krediKartlari isimli ArrayList'e eklemelidir. krediKartiSil metodu öncelikle kartın güncel borç bilgisini kontrol etmeli, eğer 0'a eşitse kartı silmeli, değilse "lütfen öncelikle borç ödemesi yapınız" uyarısını göstermelidir. kullanilabilirLimit metodu ile müşteriye güncel kullanilabilir limitini yazdırınız.
- Krediler sınıfının içinde yer alan Krediler isimli metot müşteri numarasını, krediID'sini ve krediMiktarını parametre olarak alacak. Bu metodun içinde musterNumarasi verisi kullanılarak kişinin bakiye verilerine göre yıllık kazancının %50'si kadar kredi verilebilecek.
- KrediKarti sınıfının içinde yer alan KrediKarti metodu kartNumarasi, limit, kullanilabilirLimit ve guncelBorc bilgilerini parametre olarak alacaktır.
- BankaHesap sınıfının içindeki hesapBilgisi hesap türü bilgilerini tutacaktır. Bu hesap türleri maaş hesabı ve normal hesap olacaktır. 1 maaş hesabını temsil ederken, 2 normal hesabı temsil edecektir.
- Hesapbilgisi verileri kullanılarak maaş hesabı seçildiyse vadeli hesap içinde yer alan faiz oranı %20 işletilecek, vadesiz hesap altında yer alan para transferi kısmında ise 0 TL kesinti olacak. Eğer normal hesap seçildiyse o zaman vadeli hesap altında faiz oranı %10 işlenecek, vadesiz hesap altında yer alan para transferi metodunda ise 8 TL kesinti olacaktır. (Göndericinin iban bilgisi BankaHesap sınıfından elde edilecektir.)
- krediKartiBorcOdeme metodu vadesiz bakiye ve guncelBorc miktar bilgilerini parametre olarak almalıdır. Ardından hesabın bakiyesinden bu miktar düşülmeli, kredi kartının guncelBorc değeri de ödeme miktarı kadar azaltılmalıdır.

- krediOdeme metodu krediID ve Kredi bilgilerini parametre olarak almalıdır. Ardından hesabın bakiyesinden bu miktar düşülmeli, kredinin guncelBorc değeri de ödeme miktarı kadar azaltılmalıdır.
- YatirimHesap sınıfının içindeki paraEkle metodu, yatirimBakiye, yatirimTuru ve kur u parametre olarak alacak, girilen kur miktarı ile yatırılan miktarı çarparak hesaplayıp yatirimBakiye hesabının üzerine ekleyecektir. paraBoz benzer şekilde yatirimBakiye miktarından düşecektir.
- PersonelID, tcKimlikNo, musteriNumarasi, kartNumarasi, iban gibi değerleri random number generator kullanarak otomatik olarak türetiniz. Bunların aynı olmaması için kontrolünü yapınız.
- BankaHesap açılış tarihi 1 Ocak'tan sonra açılan tüm hesapların toplamBakiye'lerini ekrana yazdırın.
- KrediKarti limiti 5000 TL altında olanların guncelBorc bilgisini yazdırın.
- KrediKarti limiti 10000 TL üstünderse kredi teklifi yapsın.
- musteriNumarasi kullanıcıdan alınan bir kişinin bütün hesap bilgilerini ekrana yazdırınız.
- Aşağıda verilen UML diyagramında ve yukarıda yapılması gerekenler kısmında metotların alması gereken parametreler belirtilmemiştir.
- Ayrıca her sınıf için ilgili get ve set metotlarını yazınız.

## Sınıf Diyagramı

Projenin sınıf diyagramı genel haliyle aşağıda verilmiştir. Bu diyagramı temel alarak, ihtiyaç duyduğunuz yerlerde gerekli eklemeleri yaparak projenizi hazırlayınız.

