

# **Banka Otomasyon Uygulaması**

**Ad : Zehra**

**Soy ad : Aktürk**

**Numara : 21360859049**

**Ders Kodu : BLM0121**

## **ÖZET**

Bu proje bir banka sistemini taklit eden bir otomasyon sistemidir. Bu projedeki amacımız Nesneye Yönelik Programlamadaki inheritance yapısını ve polimorfizm yapısını kullanabildiğimiz bir ortam yaratmak. Projenin büyük çoğunluğu UML diyagramını baz alınarak yapılmıştır. Class lar UML diyagramına göre oluşturulmuştur, getter ve setter metotları yazılmış, her sınıfa toString() metodu eklenmiştir. Ancak bazı metotlarda parametreler değiştirilmiştir. İstenilen maddeler yorumlanmış istenilen özellikler yerine getirilmiştir. Genel olarak bu projede bir banka personeli müşteri kaydedebilir ve müşterinin bilgilerini görüntüleyebilir. Bir müşteri yeni hesap açabilir ,kredi kartı oluşturabilir, kredi kartı silebilir ve kredi alabilir. Ayrıca konsoldan herhangi bir müşterinin hesap bilgileri görüntülenebilir.

## PROJENİN ÇIKTISI

```
KrediKart.java Main.java Musteri.java VadeliHesap.java YatirimHesap.java BankaHesap.java VadesizHesap.java
45 bankaPersoneli.musteriler.add(musteri4);
29
30 BankaHesap bankaHesap= new VadeliHesap("1",4330,1,1);
31 musteriler.add(bankaHesap);
32
33 BankaHesap bankaHesap2= new VadesizHesap("2",5068.4,2);
34 musteriler.add(bankaHesap2);
35
36 YatirimHesap yatirimHesap= new YatirimHesap("1",10000,"altin", (int)1245.50,3);
37 musteriler.add(yatirimHesap);
38
39 VadeliHesap vadeli= new VadeliHesap("1", 5600, (float)0.3 ,4);
40 musteriler.add(vadeli);
41
42 ArrayList<BankaHesap> bankaHesapları= new ArrayList();
43 bankaHesapları.add(yatirimHesap);
44 bankaHesapları.add(bankaHesap2);
45 bankaHesapları.add(bankaHesap);
46 bankaHesapları.add(vadeli);
47
48
49 Date ocakBir = new Date(122, 0, 1); //bir ocak gününü tutan bir date değişkeni oluşturdum.
50
51 for (BankaHesap hesap : bankaHesapları) {
52     if (hesap.hesapAcilisTarih.after(ocakBir)) {
53         System.out.println("Musteri Numarasi: "+hesap.getMusteriNumarasi()+"\tToplam Bakiye: "+hesap.getToplamBakiye());
54     }
55 }
56
57
```

```
<terminated: Main (1) [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (7 May 2023 17:53:32 - 17:53:34) [pid: 1960]
Musteri Numarasi: 3 Toplam Bakiye: 10000.0
Musteri Numarasi: 2 Toplam Bakiye: 5068.4
Musteri Numarasi: 1 Toplam Bakiye: 4330.0
Musteri Numarasi: 4 Toplam Bakiye: 5600.0
```

Bu çıktı 1 Ocak tan sonra açılan hesapların müşteri numaralarını ve toplam bakiyelerini yazdırmıştır. Bu çıktı için 4 müşteri nesnesi yaratılmıştır.

```
public class Main {
    public static void main(String[] args) {

        Musteri musteriler=new Musteri((int) rastgele(5),1,"zehra","akturk", "zehra.akturk15@gmail.com", (int) rastgele(6),144000);
        Musteri musteriler2=new Musteri((int) rastgele(5),2,"mert","karatekin", "mertkaratekin16@gmail.com", (int) rastgele(7),620000);
        Musteri musteriler3=new Musteri(213608590,3,"hayat","yilmaz", "hayat_438@gmail.com",34000,54000);
        Musteri musteriler4 = new Musteri(213608,4,"yeliz","aydin", "yeliz.aydin@gmail.com",540360,480000);
    }
}
```

Görüldüğü gibi 2 nesnenin değerleri random verilmiş,2 nesnenin değerleri girilmiştir.

```
public static long rastgele(int kacBasamak) {
    Random random= new Random();

    Set<Integer> uniqueNumbers = new HashSet<>(); //Burada uniqueNumbers.add((int) randomSayi); //icinde return randomSayi;
    long max = (long) Math.pow(10, kacBasamak);
    long min = (long) Math.pow(10, kacBasamak-2);

    long randomSayi = random.nextLong(max-min)+min;

    if (uniqueNumbers.contains(randomSayi)) {
        return rastgele(5);
    } else {
        return randomSayi;
    }
}
```

Random atama yapmamızı sağlayan fonksiyon main sınıfının altında bu şekildedir. Bu fonksiyon kaç basamaklı random sayı istediğinizi parametre olarak alır. Ve o kadar basamaklı bir sayı geriye döndürür.

```
BankaPersonel bankaPersoneli=new BankaPersonel(443564,"aysel","hakkioğlu","aysel.hakkioğlu@gmail.com",638482625);
bankaPersoneli.musteriler.add(musteriler);
bankaPersoneli.musteriler.add(musteriler2);
bankaPersoneli.musteriler.add(musteriler3);
bankaPersoneli.musteriler.add(musteriler4);

BankaHesap bankaHesap= new VadeliHesap("1",4330,1,1);
musteriler.add(bankaHesap);

BankaHesap bankaHesap2= new VadesizHesap("2",5068.4,2);
musteriler.add(bankaHesap2);

YatirimHesap yatirimHesap= new YatirimHesap("1",10000,"altin", (int)1245.50,3);
musteriler.add(yatirimHesap);

VadeliHesap vadeli= new VadeliHesap("1", 5600, (float)0.3 ,4);
musteriler.add(vadeli);
```

Bir tane banka personel oluşturulmuştur. Bilgileri girilmiştir ve resimde görülmektedir.

Ayrıca yukarıdaki resimde 4 banka hesabı oluşturulmuş ve ilgili müşterilerin hesaplar array listine eklenmiştir.

```
72
73
74     for (Musteri muster : bankaPersoneli.musteriler) {
75         for (KrediKarti kart : muster.getKrediKartlari()) {
76             if (kart.getLimit() < 5000.0) {
77                 System.out.println(muster.getAd()+" "+muster.getSoyad()+" adli musterinin 5000 tl limit altındaki kredi kartları:");
78                 System.out.println("Kart No: " + kart.getKartNumarasi() + " Güncel Borç Bilgisi: " + kart.getGüncelBorc());
79             }
80         }
81     }
82
83
```

Markers Properties Servers Data Source Explorer Snippets Terminal Console Coverage  
terminated - Main (1) [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (7 May 2023 18:13:53 - 18:13:55) [pid: 19176]  
Hayat Yılmaz adlı musterinin 5000 tl limit altındaki kredi kartları:  
Kart No: 491 Güncel Borç Bilgisi: 12000.3

Yukarıdaki çıktıda kredi kart limiti 5000 tl nin altında olan müşterilerin güncel borç bilgisi yazdırılmıştır.

```
for (Musteri muster : bankaPersoneli.musteriler) {
    for (KrediKarti kart : muster.getKrediKartlari()) {
        if (kart.getLimit() > 10000.0) {
            int secenek = JOptionPane.showConfirmDialog(null, muster.getAd()+" "+muster.getSoyad()+" Sizin için harika bir teklifimiz var!!\n Bankamız
            if (secenek == JOptionPane.YES_OPTION) {
                int cevap = JOptionPane.showOptionDialog(null, "Harika! Kredi başvurunuzu işleme alıyoruz.", "Kredi Teklifi", JOptionPane.DEFAULT_OPTI
            } else {
                int cevap = JOptionPane.showOptionDialog(null, "Üzgünüz, belki başka bir zaman.", "Kredi Teklifi", JOptionPane.DEFAULT_OPTION, JOptio
            }
        }
    }
}
```

Kredi Kartı Teklifi  
? zehra akturk Sizin için harika bir teklifimiz var!!  
Bankamızdan %1 faiz ile kredi almak ister misiniz?  
Yes No

Bu kısımda yes butonuna basılınca:

```
for (Musteri muster : bankaPersoneli.musteriler) {
    for (KrediKarti kart : muster.getKrediKartlari()) {
        if (kart.getLimit() > 10000.0) {
            int secenek = JOptionPane.showConfirmDialog(null, muster.getAd()+" "+muster.getSoyad()+" Sizin için harika bir teklifimiz var!!\n Bankamız
            if (secenek == JOptionPane.YES_OPTION) {
                int cevap = JOptionPane.showOptionDialog(null, "Harika! Kredi başvurunuzu işleme alıyoruz.", "Kredi Teklifi", JOptionPane.DEFAULT_OPTI
            } else {
                int cevap = JOptionPane.showOptionDialog(null, "Üzgünüz, belki başka bir zaman.", "Kredi Teklifi", JOptionPane.DEFAULT_OPTION, JOptio
            }
        }
    }
}
```

Kredi Kartı Teklifi  
! Harika! Kredi başvurunuzu işleme alıyoruz.  
Tamam

Yukarıdaki çıktıda kart limiti 10000 tl nin üzerinde olan müşterilere kredi teklif edilmiştir.

```
100
101     bankaHesap2.hesapGoruntuleme(2, "mert", "karatekin", 245, 3456, bankaPersoneli);
102
103
104
```

Markers Properties Servers Data Source Explorer Snippets Terminal Console Coverage  
terminated - Main (1) [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (7 May 2023 21:04:48 - 21:04:48) [pid: 19952]  
musteri numarası : 2 Hesap iban: 245 Hesap bakiye: 3456.0 hesap turu: null

Bu çıktıda ise hesap görüntüleme methodu çalıştırılmıştır.

## PROJENİN OLUŞUM SÜRECİ

### 1) Kişi Sınıfı

Öncelikle UML diyagramına göre Kişi sınıfı oluşturulmuştur. Getter ve setter metotları yazılmıştır. Parametrelili constructor UML diyagramındaki belirtilen özellikler çerçevesinde oluşturulmuştur. Parametrelili constructor oluşturduğumuz için java default olarak parametresiz constructor oluşturmamıştır. Bundan dolayı parametresiz constructor oluşturulmuştur.

### 2) Müşteri Sınıfı

Müşteri sınıfı Kisi sınıfından miras alan sınıflardan bir tanesidir. Kişi sınıfından miras aldığı için parametrelili constructor ında süper() metodu eklenmiştir.

#### i) hesapEkle Metodu

Bu sınıftaki hesapEkle metodu kullanıcının seçimine göre Vadeli ,Vadesiz ve Yatırım hesabı açabilmektedir.

```
// UML diagramında istenen methodlar
public void hesapEkle(int tcKimlikNo, int musteriNumarasi, String ad, String soyad) {

    System.out.println("Hesap türünüzü seçiniz. \n 1-Vadeli hesap 2- Vadesiz hesap 3- yatırım hesabı");
    Scanner secim = new Scanner(System.in);
    int sonuc = secim.nextInt();

    if (sonuc == 1) {
        hesaplar.add(new VadeliHesap("VadeliHesap", 3500, (float) 0.05, musteriNumarasi));
        System.out.println(ad + " " + soyad + " " + musteriNumarasi + " numaralı müşteriye Vadeli Hesap açıldı. ");
    } else if (sonuc == 2) {
        hesaplar.add(new VadesizHesap("Vadesiz hesap", 1345.23, musteriNumarasi));
        System.out.println(ad + " " + soyad + " " + musteriNumarasi + " numaralı müşteriye Vadesiz Hesap açıldı. ");
    } else if (sonuc == 3) {
        hesaplar.add(new YatirimHesap("yatırım hesap", 200000, "altın", (float) 1245.7, musteriNumarasi));
        System.out
            .println(ad + " " + soyad + " " + musteriNumarasi + " numaralı müşteriye yatırım Hesabi açıldı. ");
    }
}
```

#### ii) hesapSil metodu

Bu metod önce bütün hesapları tek tek dolaşır içerisinde müşteri numarası methota gönderilen müşteri numarası ile eşit olanı bulur ve bulunca buHesapMı isimli değişkeni true olarak atar. Sonrasında banka hesabında bakiye olup olmadığını kontrol eder. Hesapta bakiye yoksa o hesabı siler.

### **3)BankaPersonel Sınıfı**

BankaPersonel sınıfı Kişi sınıfından miras almıştır. Bu sınıfta oluşturulan müşteriler, müşteriler isimli ArrayList e eklenmekte ve burada tutulmaktadır. Projemizin büyük çoğunluğu müşteri üzerinden devam etmektedir. Bundan dolayı bu sınıfta iş yapan bir method yoktur.

### **4) Krediler Sınıfı**

Bu sınıf Müşterinin kredi bilgilerinin tutmaktadır. Parametrelili constructorümüzde müşteri numarası kontrolü yapılmıştır. BankaPersonel sınıfında bulunan Müşteri nesnelerini tutan ArrayListten yararlanılmıştır. İlgili müşterinin yıllık kazancının %50 si kadar kredi miktarı belirlenmiştir.

#### **i)Kampanya Methodu**

Bu methodda kullanıcının kart limiti güncel borcuna eşitse o müşteriye özel %0.1 faizle ihtiyaç kredisi fırsatı sunuluyor. Ve kart limiti 5000 tl ve üzerinde olan müşterilere özel her internet alışverişinde 100 bonus para fırsatı sunuluyor.

Eğer kişinin kart limiti 5000 tl altında ise bonus para kazanabilmek için limit arttırma fırsatı sunuluyor.

### **5)KrediKarti Sınıfı**

Bu sınıfta Kredi kartı özelliklerini ve methodlarını tutmaktadır. Parametrelili constructor kullanılabilirLimit değişkenine kullanılabilirLimit methodunu atanmıştır.

#### **i) krediKartiEkle Methodu**

Bu method UML diyagramından farklı olarak Müşteri nesnesini ve KrediKarti nesnesini parametre olarak almaktadır. Böylece belirli müşteri nesnesinin krediKartlari ArrayList ine belirli kredi kartı başarı ile eklendi.

#### ii) krediKartiSil Methodu

Bu method da UML diyagramından farklı olarak belirli kredi kartını belirli müşteri nesnesinden silmek için krediKarti ve Müşteri nesnesi parametre olarak alınmıştır. Güncel borç, limit ve kartNumarasi parametreleri silinmiştir. Bu parametrelere krediKarti nesnesi üzerinden ulaşılabilir. Belirli kredi kartı belirli müşterinin krediKartlari ArrayList inden başarılı bir şekilde silinmiştir.

#### iii)kullanilabilirLimit Methodu

Önce kartNumarasi adlı parametre müşteri nesnesinde bulunan krediKartlari ArrayListinden kontrol edilmiştir. Bu method müşterinin kullanılabilir limitini döndürmektedir.

```
public double kullanilabilirLimit(int kartNumarasi ,double guncelBorc) {  
    Musteri muster=new Musteri();  
    double kullanilabilirLimit=0;  
    int i;  
    for(i=0;i<musteri.krediKartlari.size();i++) {  
        if(musteri.krediKartlari.get(i).kartNumarasi==kartNumarasi) {//burada kart numaramiza ait musteriler sinifi  
            kullanilabilirLimit+=(double)musteri.krediKartlari.get(i).limit-musteri.krediKartlari.get(i).guncelBorc;  
            return kullanilabilirLimit;  
        }  
        else {  
            System.out.println("Yanlis kart numarasi girdiniz. Boyle bir kart bulunamadi.");  
            return (double)0.0;  
        }  
    }  
    return kullanilabilirLimit;  
}
```

## 6)BankaHesap Sınıfı

Bu sınıf müşterilerin her hesap açış işleminde kullanılmaktadır. Bütün banka hesaplarının ata sınıfıdır. Parametrelili constructor ın çağırıldığı tarih hesap açılış tarihi olarak kabul edilir. Constructor da hesap bilgisine göre o müşteri için maaş müşterisi ise farklı normal müşteri ise farklı faiz oranı ve kesinti ücreti hesaplanmıştır.

```

public BankaHesap(long iban, double toplamBakiye, String hesapbilgisi) {
    this.iban = iban;
    this.toplamBakiye = toplamBakiye;
    this.hesapbilgisi = hesapbilgisi;

    Calendar zaman = Calendar.getInstance();
    Date suan = zaman.getTime();
    this.hesapAcilisTarih = suan; //burada hesapAcilisTarihi isimli deđiskeni bilgissayarin suanki tarihi olarak kabul ediyorum.

    ibanlar.add(iban);

    VadeliHesap vadeliHesap = new VadeliHesap();
    VadesizHesap vadesizHesap=new VadesizHesap();

    if (hesapbilgisi == "1")
    {
        vadeliHesap.faisOrani=(float)0.20;
        vadesizHesap.paraTransfer("1", iban, toplamBakiye, musterNumarasi);
        System.out.println("Bankamizin maas musterisi oldugunuz icin para transferi islemlerinizde islem ucreti yok!!! ");
    }
    else if(hesapbilgisi == "2")
    {
        vadeliHesap.faisOrani=(float)0.10;
        vadesizHesap.paraTransfer("2", iban, toplamBakiye, musterNumarasi);
        System.out.println("Bankamizin normal hesap turu ile calistiginiz kayıt altına alınmistir.");
    }
    else {
        System.out.println("hic biri uymuyor");
    }
}

public void hesapGoruntuleme(int musterNumarasi, String ad, String soyad, long iban, double toplamBakiye,
    BankaPersonel bankaPersonel) {

```

### i)hesapGoruntuleme Methotu

Bu methotta bankaPersonel nesnesi parametre olarak alınmıřtır. Belli bir bankaPersonel müşterilerin numarasında göre hesap görüntüleme yapabilmektedir.

## 7) VadeliHesap Sınıfı

VadeliHesap sınıfı müşterinin vadeli hesap bilgilerini tutan bir sınıftır. BankaHesap sınıfından miras almaktadır.

## 8)VadesizHesap Sınıfı

Bu sınıf müşterinin vadesiz hesap bilgilerini tutmaktadır. BankaHesap sınıfından miras almaktadır.

### i)paraTransfer Methotu

Bu methotda önce verilen ibanın kontrolü yapılmaktadır. Sonrasında hesabın maaş hesabı ya da normal hesap olmasını sorgular. Maaş hesabı ise para transferi sırasında kesinti miktarı 0, normal hesap ise kesinti miktarı 8 tl olarak atanmıřtır.

```

public double paraTransfer(String hesapTuru, long iban, double vadesizBakiye, int islemMiktar) {
    BankaHesap bankaHesap=new BankaHesap();
    boolean ibanDogruMu=false;
    for (Long iban1 : bankaHesap.ibanlar) {
        if(iban1==iban) {
            ibanDogruMu=true;
        }
        else {
            ibanDogruMu=false;
        }
    }
    if(ibanDogruMu==true) {
        if(hesapTuru=="1") {
            return 0.0;
        }
        else if(hesapTuru=="2"){
            return 8.0;
        }
        else {
            return 0.0;
        }
    }
    else {
        System.out.println("yanlis iban");
        return 8.0;
    }
}

```

## ii) krediKartiBorcOdeme Methotu

Bu methot da müşterinin kredi kartı borç ödemesi sağlanmıştır. Belli bir kredi kartı ile işlem yapabilmek için kredi kartı nesnesi parametre olarak alınmıştır.

## iii)krediOdeme Methotu

Bu methot da müşterinin kredi ödemesi yapabilmesi sağlanmıştır. Kredi nesnesi parametre olarak alınmıştır. Kredi miktarı ödenen miktar kadar azaltılmış, taksit miktarı bir taksit azaltılmıştır.

```

public double krediOdeme(int krediID, Krediler kredi)//Kredi nesnesini parametre olarak aldığımız için rahatlıkla kredi miktarına ve
{
    //taksit miktarına bu nesne üzerinden erişebilirim.
    vadesizBakiye=-kredi.taksitMiktari; //bundan dolayı bu parametreleri sildim.
    kredi.krediMiktari=-kredi.taksitMiktari;//vadesiz bakiyeden kredinin bir taksidini ödemiş oluyoruz.
    return kredi.krediMiktari; //bu yüzden kredi miktarını da bir taksit azalmış oluyoruz.
}

```

## 9)YatirimHesap Sınıfı

Bu sınıf yatırım hesabı bilgilerini tutmaktadır. BankaHesap sınıfından miras almaktadır.

### i)paraEkle Methotu

Bu methot belli bir kur şeklinde belli miktar para eklememiz sağlanmıştır. Yatirilan miktar yatırımBakiye değişkenine eklenmiş, ve yatırımBakiye değişkeni döndürülmüştür.

### ii)paraBoz Methotu



Bu method girilen kur deęerinde istenilen miktarda para bozma işlemi yapmaktadır. Bozulan miktar yatırımBakiye deęişkeninden çıkartılmış ve yatırımBakiye deęişkeni döndürölmüştür.

**Not: Proje Eclipse de yazılmıştır.**