

HAZIRLAYAN ÖĞRENCİ:

ADI SOYADI: HÜMEYRA ÇİMEN

OKUL NUMARASI: 19360859053

### **BANKA UYGULAMASI NESNEYE YÖNELİK PROGRAMLAMA PROJE ÖDEVİ**

#### **Ödev amacı:**

Bir banka uygulamasının temel olarak işlevlerini yerine getiren bir proje yaptım. Bu işlemleri kısaca tanımlayacak olursak bankaya ait bir kişi ve bu kişiden miras alan banka personeli ,müşteri ;müşterinin yapabilecekleri hesap açma kapama hesap bilgisi alma kredi kartı ekleme silme bilgisini alma, para ekleme çekme , kredi kartı borç ödeme ,para transferi yapma vb....

Proje ödev kapsamında belirtilen tüm özellikleri yetine getirecek durumda tasarlanmıştır .

#### **Banka personeli :**

Ödevde belirtilen özellikler kapsamında kişi, müşteri ve banka personeli adlı sınıfı oluşturdum; extends keyword ile banka personeli ve müşteri sınıfının kişi sınıfından miras almasını sağladım. Bu sınıflar için hepsine ayrı ayrı yapıcı blok (constructor method)larını parametrelili ve/veya parametresiz olarak tanımladım . Sınıflar içinde tanımlanan özellikleri private tanımlayarak ; getter ve setter methodları ile özelliklere (attributes) ulaşım sağlayarak kapsülleme (encapsulation)işlemi tamamladım .

Ödevde belirtildiği üzere rastgele olarak verilmesi istenen özellikler için rastgele(random) nesnesi üretmek (int)tam sayılar üretmek her seferinde çeşitli basamaklarda farklı değerler almasını sağladım .

Miras aldığı kişi sınıfına ait özellikleri barındırıyordu banka personelinin içinde personel id özelliği random olarak üç basamaklı constructor (yapıcı blok ) içerisinde oluşturuldu .(constructor newlendiği zaman çalışan blok.)

Constructor bloğu içerisinde random olarak banka personeli id si oluşturuldu.(Bunun için java oracle sayfasından random sınıfını newleme ve kullanma kalıplarını inceledim)Öncelikle bir random nesnesi oluşturdum 0 ile 1 arasında bir sayı üretmesini sağladım **üç basamaklı** olması için bunu yüzler ve onlar basamağı şeklinde atamasını yaparak üç basamaklı bir personel numara oluşturdum. Müşteri sınıfından kişi sınıfından miras aldı random olarak oluşturulan müşteri numarasını yine constructor içerisinde set metodu kullanılarak yine random sınıfından nesne üretmek iki basamaklı (integer) bir sayı oluşturarak tanımladım. Extend ettiği sınıftaki metotlara ulaşabilmek için **süper** anahtar kelimesini kullandım .

**arayüz :** müşteri ekle metodu kullanıcıdan aldığı **ad soyad Email ve telefon numarasıyla** müşteri sınıfından bir nesne üretmek yine müşteri sınıfındaki set metotlarını kullanarak kişiler sınıfından ve müşteri sınıflarından oluşturulan kişiler ve **müşteriler adlı listeye (arraylist)** ekledim .

Müşteri kişi sınıfından miras aldığı için hem kişiler sınıfına hem de müşteriler sınıfında yer alıyor kişiler sınıfında müşteriler haricinde bir de banka personelleri yer alıyor .

arayüz kısmında **müşteri bilgilerini** getireceğim bir metot oluşturdum bu metot kullanıcıdan tam sayı **bir müşteri numarası** istiyor. Bu müşteri sınıfını bulmak için **for** döngüsüyle müşteriler listesinde tek tek gezdim **if blok** ile kullanıcıdan aldığım müşteri numarasını müşteriler listesindeki ,müşteri numaraları ile eşleşen müşteriyi , ekranda mesaj olarak gösterdim .

Peki bu müşteri numarası herhangi bir müşteri listesinin içerisindeki müşteri numarası ile eşleşmedi bu durumda ise **böyle bir müşteri bulunamadı** diyerek uyarı verdim. Banka personeli ekle kısmı da hemen hemen bununla aynı şekilde yaptım.

#### **Kredi kartları ile ilgili işlemlerin Açıklamaları :**

Kredi kartı ve banka hesabı adlı iki Class müşteri sınıfı ile bağlantı halde(compozisiton) ama implement ve extent durumları yok bu durumda müşteri sınıfında yer alan hesaplar ve kredi kartları adlı arraylist bu sınıflardan oluşturulan kredi kartı ve hesap bilgilerini tutuyor.

kredi kartı klasında ara yüzde bulunan **kredi kartı ekleme metodu** ile bir kredi kartı ekleniyor kredi kartı eklemek için bir **müşteri numarasına** sahip olması gerektiği için kullanıcıdan bir müşteri numarası aldım **Müşteri bilgileri** kısmında açıkladığım gibi yine **for döngüsü kullanarak** müşterileri adlı dizide gezdim müşteri numarası ile eşleşen sınıf üzerinden müşteri sınıfı içerisinde bulan **kredi kartı ekle metodunu** çağırdım kredi kartı ekle metodu parametre olarak **bir limit** değeri alıyor kredi kartına eklemek için kredi kartı sınıfı getter ,setter ve yapıcı (const) bloklarına sahip ve random olarak atanan bir **kart numarası** değerine sahip.

Bu kart numarası için farklı bir metot kullandım constructor (yapıcı blok) içerisinde **generatör kart numarası** adlı bir metoda çağırdım bu metot random sınıftan bir nesne oluşturarak **for döngüsü ile 3 tane int** sayı ürettiyor **fakat kart numarasını String** olarak tutmayı tercih ettim.

**kredi kartı ekle** metodun yanı sıra tabii ki **de kredi kartı silme** metodunda da hemen hemen aynı işlemler tekrarlandı fakat **silinecek kredi kartını bulmak** için kullanıcıdan **bir kart numarası** istedim Bu kart numarası ile kredi kartları içerisindeki kart numaraları ile **eşleştirerek** kullanıcıdan alınan kart numarasına ait kredi kartı hesabını aldım edindiğim bu bilgiyi **Remove metodu** ile kredi kartları adı listeden sildim. Fakat burada kodu sadeleştirmek için elimde kredi kartı bilgisinin bulunduğu listedeki indeks numarasını remove mothoduna bir **index** verdim(**bu index ilgili bilgileri tutan dizi içindeki konumu** ) daha sonra **kredi kartının** silindiğine dair kullanıcıya bir mesaj verdim .

**kredi kartı bilgileri:** Kredi kartı bilgileri için kredi kartın müşteri sınıfın ile bağlantı kurabileceğim için kullanıcıdan bir **müşteri numarası girmesini ve bir kart numarasını** girmesini isteyerek ulaşmak istediği kredi kartı bilgilerini to String metotları ile **verdim** .

**kredi kartı Ekle ve kredi kartı sil** metotları **müşteri sınıfının** içerisinde yer aldığı için ekleme ve silme işlemleri o metotlar içerisine yapıldı. framede (arayüz) sadece **hangi hesaba ait kredi kartı bilgisi eklenecek veya silinecek tespitini yaptı** .

**Banka hesabı** adlı sınıf **kredi kartı hesap bilgileri, hesap ekleme, hesap silme** metotlarına sahip, bu komutlar içerisinde neler yaptım :

**Hesap ekleme kodu ile başlıyorum hesap ekleme metodu** tıklandığında **masum mosion listener ve actionperformed** metotları ile bu tıklamalara bir anlam yükleyebiliyoruz ve tasarım olarak hesap ekleyebilmem için bir

müşterinin olması gerekiyor bu yüzden kullanıcıdan bir müşteri numarası istiyorum Banka hesabı iki tane mirasçıya sahip .Bunlar vadesiz hesap ,yatırım hesabı ; banka hesabını miras alıyorlar Bu yüzden kullanıcıya oluşturmak istediğim hesap türünü soruyorum J-optionconfirmdialog ile oluşturmak istediği hesap türünü vadesiz ise "1" ;yatırım hesabı ise "0" ile Giriş yaparak belirtmesinin istiyorum .Müşteri numarasını yine müşteriler listesinden kontrol edip eşleşen müşteri ailesine ait müşteri hesabını müşteri sınıfı içerisinde bulunan **Hesap ekle fonksiyonunu** metot olarak çağırıyorum ,eşleşmemesi durumunda bir uyarı mesajını kullanıcıya gösteriyor. Müşteri sınıfı içerisindeki hesap ekle metoduna değinecek olursak kullanıcıdan alınan 1 veya 0 vadesiz veya yatırım hesabı olarak ilk blokları ile Kontrol ettim bir girişi için vadesiz hesaptan Yeni bir hesap oluşturdum ve müşteri sınıfın içerisindeki hesapların adları listeye ekledim eklenen hesap bilgilerini kullanıcıya gösterdim iki durumda 0 girdiği durum oluyor burada yatırım hesabı oluşturarak yine hesaplar adı liste ekleyerek bilgileri gösterdim kullanıcının farklı bir giriş yaptığı durumu kontrol etmek için en son olarak else ile bir uyarı mesajı göndererek 1 veya 0 ne için girmesi gerektiğini belirttim

**Evet hesap ekledik peki hesap silme işlemi nasıl oluyor ?**

Yine bir müşteri numarası ile işlem yapılmak istenen müşteri buluyorum banka hesabında Random olarak oluşturduğumuz **String IBAN** değerini burada silmek istenen *Hesabı kontrol etmek* için kullanıyorum yani kullanıcıdan silmek istenen hesabın IBAN numarasını alıyorum ardından for ile müşteriye ulaşıyorum ve elde ettiğim bu IBAN numarasına ait hesabı "**hesap sil metodunu**" kullanarak ibanı parametre olarak vererek hesabın Hesap sil işlemi gerçekleştiriliyor. Fonksiyonun içeriğine bakacak olursak parametre olarak aldığı IBAN bilgisine ait hesap içerisinde bir index numarası oluşturmasını sağladım bu index numarasında Remove metoduna parametre olarak verdim Bu şekilde silindi tabiki yine IBAN numarasının yanlış girilmesi ve müşteri numarasını yanlış girilmesi gibi durumları if blokları ile kontrol ederek kullanıcı uyarı mesajları yolladım en sonunda silinen hesabı da 'silinen Hesap 'olarak bilgilerde gösterdim .

Burada dikkat etmem gereken nokta hesabı sildikten sonra hesaba ait bilgileri göstermeye çalışmam olurdu ,fakat ben hesabın kontrolünü sağlayan silinen adlı bir geçici değişken ile bu sorundan kurtuldum bunu yapmazsam farklı bir yöntemde silme işlemi yapmak için kullandığın kod satırın hemen üzerinde hesap bilgisini göstermem gerekirdi .

Para çek fonksiyonu UML diyagramında belirtildiği üzere para ekleme ve çekme yatırım hesapları ile ilgili bir fonksiyon Bu yüzden kullanıcıdan aldığım iban bilgisinde **hesabın yatırım hesabı olduğunu kontrol etmem gerekti** .para eklenecek hesap ve para çekilecek olan hesabın aslında yatırım hesabı olması gerekiyordu .İç içe iki tane for ve if döngüleri kullanarak yatırım hesabı olmasını kontrol ettim .burada farklı

olarak iki tane for döngüsü içerisinde yer alan yatırım hesabına eşit olmasını sağlamam gerekiyordu .Aldığım String IBAN bilgisi bir banka hesabı, fakat benim bu Banka hesabını tutacak bir yatırım hesabı olmasını sağlamam gerektiği için ilk blog içerisinde in instanceof keyword ile ilgili ibanın hesabı vadesiz hesap mı yoksa yatırım hesabı mı olduğu kontrolünü yaptım ve yatırım hesabı üzerinden geçici bir hesap değişkeni oluşturarak bu hesap değişkeni üzerinden Para çek fonksiyonunu kullanıcıdan aldığım miktar bilgisi ile çağırdım yatırım hesabı içerisindekin Para çek fonksiyonuna bakacak olursak önce kullanıcıdan para çekmek üzere gönderdiği miktar Double değişkenini kontrol etmem gerekiyordu eğer bu miktar kullanıcının bakiye bilgisinden daha küçükse kullandığın bakiye bilgisini güncelledim Aksi takdirde bakiyeniz yetersiz olduğu durum olarak bir uyarı mesajı gönderdim para ekle metodu yine bir Müşteri bilgisi ve eklenmek istenen miktar Bir de eklenecek olan IBAN bilgisini alarak bu bilgilerin listeleri içerisindeki kontrollerini sağladım tüm durumların üretebileceği Hatta mesajlarını “switch Case” yapısı ile belirttim burada değişik olarak instance of ile IBAN bilgisinin yatırım hesabına ait olup olmadığı kontrolünü yaptım yine yatırım hesabı içerisindeki para ekle fonksiyonuna miktar bilgisini parametre olarak vererek miktar güncellemesini tamamladım.

Para transferinden farklı olarak iki tane IBAN bilgisi aldım bu IBAN bilgilerinin ikisini de yatırım hesabı olmasının kontrolünü yaparak bakiyelerinde güncelleme yapmalarına olanak sağladım.

Kredi kartı borç ödeme için müşteri numarası ve ödeme miktar ödeme yapmak hesabın IBAN'ı gerekiyordu tüm abonelikleri kontrol ederek kredi kartı borcunu güncelle metodunu çağırdım vadesiz hesap içerisinde kredi kartı borç ödeme fonksiyonu olduğu için ayrı olan İban bilgisini bir vadesiz hesap olup olmadığının kontrolünü sağladım

#### NOT:

Projeyi “frameleme” adlı fram den “run file” şeklinde çalıştırınız !

Yazılan kodlar içinde önceden “denediğim kötü(temel)” kodlar yorum satırları olarak bulunmaktadır .

Girdiğiniz değerlere lütfen dikkat ederek projeyi çalıştırınız !(miktar bilgisi double olarak girilmeli ve iban bilgisi TR ifadesiyle birlikte yazılmalı vs.)

ÖDEVE EK OLARAK KENDİM KREDİ KARTINDAN PARA ÇEKME İŞLEMİ YAPAN METHODU TANIMLADIM TANIMLANA KREDİ KART LİMİT DEĞERİNİ BAZ ALARAK İSTENEN MİKTARIN ÇEKİLİP ÇEKİLMEME DURUMU KONTROL EDİLİYOR EĞER MİKTAR LİMİT DEĞERİNDEN DÜŞÜK İSE İŞLEM VADESİZ HESAP İÇİNDE TANIMLANMASI GEREKEN KREDİ KARTI PARA ÇEK METHODU ÇAĞIRILIYOR İŞLEM TAMMALANIYOR BU EK METOTLAR ÖDEV PROJESİNE DAHİL DEĞİLDİR EK DOSYA(NOT DEFTERİ) İLE GÖNDERİLMİŞTİR .