# Nesneye Yönelik Programlama

2020-2021 Bahar Dönemi Proje Ödevi Raporu

Batuhan Şimşek ----------- 20360859055

Projede genel hatlarıyla bir çevrimiçi banka uygulamasının temel back-end kodlaması istenmişti.Raporda, projede hazırladığım her bir sınıfı ve metotları ayrı ayrı açıklayarak ilerleyeceğim.

Kisi.java

Sistemdeki bütün insanların -müşteri,personel vb.- hepsinde bulunabilecek temel özelliklerin yer aldığı ana sınıflardan biridir.Ad,soyad,e-mail ve telefon numarası değişkenleri burada tutulur,diğer sınıflara miras verilir.İçinde kendi Constructor’ ı ve toString() metodu dışında metot bulunmaz.

BankaPersoneli.java

Kişi sınıfından miras aldığı nitelikler dışında personelID niteliği ve musteriler isimli, oluşturulan müşteri nesnelerinin saklandığı bir ArrayList bulundurur.personelID niteliği rastgele olarak üretilir ve atanır.

Musteri.java

Kişi sınıfından miras aldığı nitelikler dışında musteriNumarasi niteliği, hesaplar ve krediKartlari isimli, ilgili nesnelerin saklandığı ArrayList’lere sahiptir.İçerisinde toString() metodu haricinde dört farklı metot daha bulundurur.

*hesapEkle()* Metodu

Parametre olarak açılacak hesabın bakiyesini, ve açılacak hesap türünü belirlemek için ilgili hesap türünüe ait nesneyi alır.

Alınan bakiyeyi açılan hesap bakiyesine atar, açılan hesabı ise ilgili ArrayList’e ekler.İki hesap türü olduğu için aynı isimle iki farklı şekilde yazılarak “Overload” edilmiştir.

*hesapSil()* Metodu

Parametre olarak silinecek hesap nesnesini alır. Alınan hesabın bakiyesini kontrol eder, bakiye boş ise ArrayList’ten nesneyi siler,bakiyede para var ise bakiye aktarım uyarısı verir.

*krediKartiEkle()* Metodu

Parametre olarak eklenecek kartın borç - limit bilgisini ve ilgili nesneyi alır.(KrediKarti sınıfından herhangi bir nesne)

Alınan borç – limit bilgilerini eklenen kart limiti ve bocu olarak atar, eklenen kartı ise ilgili ArrayList’e ekler.

*krediKartiSil()* Metodu

Parametre olarak silinecek kart nesnesini alır.

Kartın güncel borç durumunu kontrol eder, eğer borç varsa borç ödeme uyarısı verir,borç yok ise ilgili nesneyi ilgili ArrayList’ten siler.

KrediKarti.java

İçerisinde kartNumarasi, limit, guncelBorc, kullanılabilirLimit değişkenleri, Constructor ve toString() metodu bulunur.Kart numarasının rastgele türetilmesinde 16 haneli üretilmesine dikkat edilmiştir.

BankaHesabi.java

İçerisinde iban ve bakiye değişkenlerini, Constructor’I ve toString() metodunu barındırır. Iban değişkeninin uluslararası kurallara uygun olacak şekilde “TR\*\* \*\*\*\* \*\*\*\* \*\*\*\* \*\*\*\* \*\*\*\* \*\*” formatında rastgele üretilmesine dikkat edilmiştir.

VadesizHesap.java

BankaHesabi sınıfından miras alır, içerisinde hesapTürü değişkeni, Constructor,toString() ve iki metot daha bulunmaktadır.

paraTransferi() Metodu

Transfer yapılacak hesapların nesnelerini ve aktarılacak tutarı parametre olarak alır, birinci sırada yazılan hesaptan ikinci sırada yazılan hesaba transferi gerçekleştirir.

krediKartiBorcOdeme() Metodu

Borç ödenecek hesap ve kredi kartı nesnelerini ve ödenecek miktarı parametre olarak alır, ödenen miktarı ilgili kartın guncelBorc değerinden düşerek borç ödemesini gerçekleştirir.

YatirimHesabi.java

BankaHesabi sınıfından miras alır, içerisinde hesapTürü değişkeni, Constructor,toString() ve iki metot daha bulunmaktadır.

paraEkle Metodu

Para eklenecek hesap nesnesini ve yatırılacak para miktarını parametre olarak alır. İlgili hesabın bakiyesini verilen parametre doğrultusunda günceller.

paraÇek Metodu

Para çekilecek hesap nesnesini ve çekilecek para miktarını parametre olarak alır. İlgili hesabın bakiyesini verilen parametre doğrultusunda günceller.

Frame.java

Projenin istenilen GUI kısmı için oluşturulmuştur.Bu sınıf ve içerikleri oluşturulurken Eclipse IDE’sine ait WindowBuilder eklentisi kullanılarak -NetBeans için JFrame kullanılarak yapılmasıyla aynıdır.- yapılmıştır. Dizayn panelinden tasarımı oluşturulmuş, ilgili kodlar ise otomatik olarak yazılmıştır bu eklenti sayesinde.Ancak projede istenilen müşteri bilgilerinin tamamının bir liste şeklinde sırayla gösterilmesini tam olarak başaramadım, bunun yerine en azından GUI’ye o esnada girilen bilgilerle oluşturulan müşteri nesnesinin bilgilerini, müşteri bilgileri butonuna tıklandığında bir uyarı penceresi açarak gösterecek şekilde bir kod yazdım.

Test.java

Sınıflarda yazılan metotların çalışıp çalışmadığının, çıktılarının ne olduğunun görülebilmesi için oluşturulan test sınıfı. Gerekli tüm nesneler bu sınıf içerisinde oluşturulup gerekli metotlarda kullanılmıştır. Neredeyse her metodu kullanarak test ettim.