



PROJE RAPORU: Banka ATM Sistemi

1. Proje Adı

ATM

2. Projenin Amacı

Bu projenin amacı, bir bankanın ATM sistemini simüle ederek para çekme, para yatırma, bakiye görüntüleme, şifre değiştirme ve QR kod gibi işlemleri Java programlama dili ve OOP (nesneye yönelik programlama) prensipleriyle gerçekleştirmektir. Kullanıcı arayüzü (GUI) ile birlikte kullanıcı dostu bir deneyim sağlanmıştır. Kullanıcıya modern ve kullanıcı dostu bir deneyim sunmak amacıyla farklı dil desteği eklenmiş ve QR kod işlemleri için **ZXing kütüphanesi** kullanılmıştır.

3. Kullanılan Teknolojiler

- Java (JDK 23)
- Eclipse IDE
- Java Swing (GUI için)
- Dosya İşlemleri
- OOP Prensipleri (Encapsulation, Inheritance, Polymorphism, Abstraction, Interfaces)
- **ZXing (Zebra Crossing) Kütüphanesi (Versiyon 3.5.1)**
QR kod oluşturma ve okuma işlemleri için kullanılan açık kaynak kütüphane. Proje dosyaları arasında bulunan lib klasöründe core-3.5.1.jar ve javase-3.5.1.jar dosyaları ile projeye entegre edilmiştir.

→ core-3.5.1.jar: ZXing'in temel barcode işleme bileşenlerini içerir.

→ javase-3.5.1.jar: Java SE ortamında QR kod işlemlerini kolaylaştıran yardımcı sınıfları içerir.

4. Sınıfların ve Yapının Açıklaması

Main	ATM uygulamasını başlatan ana sınıftır. Uygulamayı çalıştırır ve ilk olarak GirişEkranı'nın açılmasını sağlar.
GirişEkranı	<p>Kullanıcıların ATM sistemine kullanıcı adı ve şifre bilgileriyle giriş yaptığı grafiksel kullanıcı arayüzü (GUI) ekranıdır.</p> <ul style="list-style-type: none">Dil seçimi, şifre göster/gizle, bloke kontrolü, hatalı giriş sayacı gibi işlevler burada bulunur.
AnaMenü	<p>Başarılı giriş sonrası kullanıcıya sunulan ana işlem ekranıdır.</p> <ul style="list-style-type: none">Banka hesabı ile ilgili işlemler (bakiye görüntüleme, para çekme, para yatırma vb.) bu ekranda yapılır.
DigerİslemlerPenceresi	<p>Kullanıcıya ATM üzerinde ekstra işlemler sunan alt penceredir.</p> <ul style="list-style-type: none">QR kod oluşturma ve göstermeŞifre değiştirmeKullanıcı bloke kaldırma işlemleri gibi seçenekler içerir.
DilSecimEkranı	<p>Kullanıcının sistemde kullanmak istediği dili seçmesine olanak sağlayan ekran.</p> <ul style="list-style-type: none">Seçilen dile göre uygulamanın arayüz dili değiştirilir.
QRKodIslemleri	QR kodların oluşturulması, gösterilmesi ve doğrulanması işlemlerini gerçekleştiren yardımcı sınıftır.

	<ul style="list-style-type: none"> Kullanıcının hesap bilgilerini QR kod formatına çevirir ve güvenli işlemler için kullanır.
RoundedButton	<p>GUI’de kullanılan, köşeleri yuvarlatılmış özel bir buton bileşenidir.</p> <ul style="list-style-type: none"> Standart JButton’a göre daha estetik görünüm sağlamaktadır.
TesekkurEkranı	<p>İşlem sonrasında kullanıcıya teşekkür mesajı gösteren ekran.</p> <ul style="list-style-type: none"> Kullanıcı deneyimini artırmak için bilgilendirme ve kapanış mesajları sunar.
Kullanici	<p>ATM sistemine kayıtlı tüm kullanıcıların temel özelliklerini tanımlayan sınıf.</p> <ul style="list-style-type: none"> Kullanıcı adı, şifre, bloke durumu gibi ortak özellikleri içerir.
Musteri	<p>Kullanici sınıfından türetilmiş, banka müşterilerini temsil eden sınıf.</p> <ul style="list-style-type: none"> Hesap bakiyesi, hesap numarası, müşteri özel işlemleri gibi bilgiler içerir.
IslemYapilabilir	<p>İşlem yapılabilen sınıflar için tanımlanmış bir arayüz (interface) veya soyut sınıf.</p> <ul style="list-style-type: none"> Para çekme, para yatırma gibi işlemlerin imzalarını içerir.
MusteriIslemleri	

	<p>Müşteri hesapları üzerinde yapılabilen banka işlemlerini yöneten sınıf.</p> <ul style="list-style-type: none"> Bakiye kontrolü, para transferi, işlem geçmişi gibi işlevleri barındırır.
Dosyaİslemleri	<p>Kullanıcı bilgileri, işlem kayıtları, bloke durumları gibi verilerin dosya sistemi üzerinde okunması ve yazılması işlemlerini gerçekleştirir.</p> <ul style="list-style-type: none"> Dosya tabanlı veri yönetimi sağlar.
DosyaYardimcisi	<p>Dosya işlemlerini kolaylaştıran yardımcı sınıf.</p> <ul style="list-style-type: none"> Kullanıcı doğrulama, bloke kontrolü, işlem kaydı tutma gibi yüksek seviye fonksiyonlar içerir.

5. Kullanıcı Arayüzü (GUI)

GirişEkranı: Kullanıcıların sisteme erişimi için kullanıcı adı ve şifre bilgileri ile giriş yapabilecekleri güvenli bir ekran sunar. Yanlış girişlerde uygun uyarılar verir.

- Giriş ekranında kullanıcı giriş ekranı (kullanıcı adı, şifre) bulunur.
- Ana menüde işlem menüsü: Para çek, yatır, bakiye sorgula, çıkış.
- “Diğer Seçenekler” butonu: Bloke kaldırma, şifre değiştirme, QR kod işlemleri gibi ek işlemler içerir.
- Turuncu tuşlar, beyaz ve gri arka plan (tasarım özellikleri) bulunur.

6. OOP Kullanımı

- Kalıtım (Inheritance):**
Musteri sınıfı, ortak özellik ve metotları paylaşmak için Kullanici sınıfından türetilmiştir. Ayrıca, İşlemYapilabilir arayüzünü (interface) implemente ederek işlem yapma yeteneği kazanır.
- Soyutlama (Abstraction):**
Kullanici sınıfı soyut (abstract) olarak tanımlanmıştır. Bu sınıf, ortak kullanıcı

özelliklerini ve temel metodları içerir, ancak detaylı işlemler alt sınıflar (Musteri) tarafından override edilir.

- **Arayüzler (Interfaces):**
IslemYapilabilir arayüzü, müşteri işlemleri için gerekli metotları tanımlar. Bu sayede farklı kullanıcı tipleri bu arayüzü kullanarak işlemlerini gerçekleştirebilir.
- **Çok Biçimlilik (Polymorphism):**
Programda IslemYapilabilir arayüzü referansları kullanılarak, farklı kullanıcı nesneleri (örneğin Musteri) üzerinde aynı işlemler gerçekleştirilebilir. Böylece esnek ve genişletilebilir bir yapı sağlanır.

7. Dosya İşlemleri

Projede kullanıcıların kimlik doğrulama, bakiye yönetimi ve işlem kayıtlarının tutulması için üç ayrı dosya kullanılmaktadır:

- **kullanici.txt:**
Kullanıcıların giriş bilgileri (kullanıcı adı, şifre ve bloke durumu) bu dosyada tutulur. Sistem giriş ekranı, bu dosyadan kullanıcı bilgilerini okuyarak doğrulama yapar. Ayrıca, şifre değişikliği veya bloke kaldırma işlemleri de burada güncellenir.
- **bakiye.txt:**
Her kullanıcıya ait banka hesap bakiyeleri bu dosyada saklanır. Para yatırma, çekme ve bakiye sorgulama işlemleri bu dosya üzerinden gerçekleştirilir. İşlemler sonrası bakiye bilgisi güncellenir.
- **log.txt:**
Kullanıcıların yaptığı giriş işlemleri tarih ve saat bilgisi ile birlikte bu dosyada kaydedilir. Böylece giriş işlemi geçmişi tutulur ve gerektiğinde geriye dönük kontrol sağlanabilir.

8. Ek Özellikler

Bu projede yalnızca temel bankacılık işlemleri değil, kullanıcı deneyimini zenginleştiren çeşitli görsel ve işitsel detaylar da uygulanmıştır.

- **Görsel Arayüz:** Arka plan tasarımı ve ikonlar ile ekranlar daha anlaşılır ve profesyonel hale getirilmiştir.

- **Sesli Geri Bildirim:** Kullanıcının tekrar giriş yap butonuna basması ve giriş ekranına geçişi esnasında ses eklenmiştir.
- **İkon Kullanımı:** İşlem butonunda yer alan ikonlar, işlemlerin ne işe yaradığını görsel olarak ifade ederek özellikle ilk kez kullananlar için kolaylık sağlamıştır.

9. Sonuç

Bu proje sayesinde, hem Java dilinde ileri seviye programlama teknikleri hem de nesne yönelimli programlamanın temel prensipleri uygulanmıştır. Ayrıca kullanıcı dostu bir arayüz ile sistemin kullanılabilirliği artırılmıştır.