

# İLERİ DÜZEY PROGRAMLAMA ÖDEV 1

## Konu: Fitness Center Üyelik ve Ders Rezervasyon Sistemi (Java GUI)

**Son Teslim Tarihi:** 09.11.2025 – Pazar (23:59)

**Danışman:** Işıl Karabey Aksakallı

Bu ödevde, bir spor salonu (fitness center) için üyelik yönetimi ve ders rezervasyonlarını yöneten bir Graphical User Interface (GUI) uygulamasını Java kullanarak geliştirmeniz beklenmektedir. Ödev, sağlanan örnek dokümandaki görev akışına benzer şekilde kademeli ve anlaşılır adımlara ayrılmıştır. Aşağıdaki adımları takip ederek uygulamanızı geliştiriniz.

### 1) Başlatma ve Veritabanı Mantiği (backup.dat)

- Uygulama açılır açılmaz, çalışma dizininde bulunan "backup.dat" dosyasını okuyarak bellekteki veritabanınızı (ör. Collections) başlatınız.
- Eğer "backup.dat" dosyası YOKSA, aşağıdaki varsayılan kayıtlarla boş bir veritabanı oluşturunuz:
  - Kullanıcı: admin (şifre: password). Şifreyi hash'lenmiş olarak tutmalısınız. Kendi hashing algoritmanızı yazınız.
- Uygulama kapanırken (pencerenin yerel kapatma butonu ile), güncel verileri tekrar "backup.dat" dosyasına yazdırınız.
- Bir sonraki çalıştırmada sistemin, oluşturduğunuz bu yedek ile aynen ayağa kalktığından emin olunuz.

Örnek "backup.dat" satır formatları (sekme `\t` ile ayrılmış):

```
user kullanıcı_adi hashed_password true_if_active_false_if_not
true_if_admin_false_if_not
```

```
member üye_adi membership_type(standard/premium)
expiry_date(YYYY-MM-DD)
```

```
trainer eğitmen_adi expertise_field("Yoga", "Fonksiyonel", "Cardio" vb.)
```

```
class ders_adi eğitmen_adi max_capacity current_enrollment
```

```
booking kullanıcı_adi ders_adi date(YYYY-MM-DD)
status(active/cancelled)
```

## 2) Giriş (Login) ve Kayıt (Register) Pencereleeri

Aşağıdaki adımları izleyerek kullanıcı giriş ve kayıt akışını tasarlayınız:

- Login ekranı: Kullanıcı adı ve parola alanları bulunmalıdır.
- Parola doğrulaması: Girilen parola, veritabanındaki HASH ile karşılaştırılmalıdır.
- Başarısız girişlerde kullanıcıya uyarı dialog'ı gösteriniz (ör. "Hatalı kullanıcı adı veya şifre!").
- Kayıt ekranı: Şifrelerin eşleşip eşleşmediğini kontrol ediniz. Zaten kayıtlı kullanıcı adı girildiyse uyarı veriniz.
- Giriş başarılıysa; kullanıcı rolüne göre (admin/üye) ilgili ana ekrana yönlendiriniz.

## 3) Admin Ana Ekranı

Admin olarak giriş yapıldığında aşağıdaki bileşenler yer almalıdır:

- Karşılama metni: Kullanıcı adı ve rol bilgisi (örn. "Hoş geldiniz, Admin Isil!").
- Butonlar: "Üye Ekle", "Eğitmen Ekle", "Ders Ekle", "Rezervasyonları Yönet", "Çıkış".
- Durum/mesaj alanı: Yapılan işlemler hakkında kısa geri bildirim veriniz.

## 4) Üye (Member) Ana Ekranı

- Karşılama metni: Üye adı ve üyelik türü (standard/premium) görülsün.
- Mevcut derslerin listesi: Her ders için ad, eğitmen, tarih/saat (opsiyonel), kapasite ve kalan kontenjan bilgisi.
- İşlemler: Derse rezervasyon yapma, mevcut rezervasyonu iptal etme.
- Çıkış butonu: Oturumu kapatıp giriş ekranına döndürsün.

## 5) Pencere Detayları (Adım Adım)

### 5.1) Üye Ekle Penceresi (Admin)

- Girdi alanları: Ad Soyad, Kullanıcı Adı, Parola, Üyelik Türü (standard/premium), Başlangıç ve Bitiş Tarihi.
- Doğrulamalar: Boş alan kontrolü, kullanıcı adının benzersizliği, tarih biçimi kontrolü.
- Kayıt: Başarılıysa veritabanına eklenmeli ve backup.dat'a yansıtılmalıdır.

### 5.2) Eğitmen Ekle Penceresi (Admin)

- Girdi alanları: Eğitmen Adı, Uzmanlık Alanı (örn. Yoga, Pilates, Fonksiyonel, Cardio).

- Doğrulamalar: Boş alan ve mükerrer kayıt kontrolü.
- Kayıt sonrası: Liste yenilensin ve mesaj gösterilsin.

### 5.3) Ders Ekle Penceresi (Admin)

- Girdi alanları: Ders Adı (benzersiz), Eğitimci (dropdown), Kapasite (pozitif tamsayı).
- Doğrulamalar: Boş alan, benzersiz ad, sayısal kapasite kontrolü.
- Kayıt sonrası: Ders listesi güncellensin.

### 5.4) Rezervasyonları Yönet (Admin)

- Tüm rezervasyonları tablo olarak gösterin (kullanıcı, ders, tarih, durum).
- Filtreleme: Tarihe veya kullanıcı adına göre filtreleyebilme.
- İşlem: Seçilen rezervasyonu iptal etme (status = cancelled).

### 5.5) Ders Rezervasyon Penceresi (Üye)

- Ders listesi: Uygun dersler listelensin, kapasite doluyorsa rezervasyon butonu pasif olsun.
- Rezervasyon: Üye boş kontenjan varsa rezervasyon yapabilsin.
- İptal: Üye kendi rezervasyonunu iptal edebilsin.
- Geri/İleri: Önceki ekrana dönüş ve onay adımları hatasız çalışsın.

## 6) Medya/İçerik (Opsiyonel)

Sistemde bir tanıtım videosu veya sınıf tanıtım klipleri oynatmayı düşünüyorsanız (opsiyonel), oynat/duraklat, 5 sn ileri/geri sarma gibi kontrolleri sağlamalı ve medya sonunu/aşımını güvenli şekilde yönetmelisiniz. Bu bölüm zorunlu değildir.

## 7) Hata Yönetimi ve Kullanılabilirlik

- Tüm input'lar için uygun hata mesajları gösteriniz (boş alan, hatalı format, zaten mevcut kayıt vb.).
- Kritik işlemler (silme/iptal) için kullanıcıdan onay alınız.
- Ekran geçişlerinde durumun (ör. seçili ders/rezervasyon) doğru taşındığını doğrulayınız.
- Kapanışta backup.dat güncellendiğinden emin olunuz.

## 8) Bonus Özellik (Zorunlu Değil, +10'a kadar)

Aşağıdakilerden biri gibi YARATICI ve FAYDALI bir ekstra özellik ekleyiniz. Raporunda amacını, faydasını ve nasıl kullanıldığını açıklayınız. Aynı ekstra özelliği kullanan ödevler bonus alamaz. Örneğin:

- Yoklama/katılım takibi (üyenin derse katılımı otomatik işaretlensin).
- Üyelik bitişi yaklaşanlara hatırlatma (uyarı penceresi).
- Basit istatistikler: en popüler ders, en yoğun eğitmen, haftalık rezervasyon sayısı vb.

## 9) Rapor Kuralları

- Kapak ve Dizin sayfaları.
- Sorun Tanımı: Geliştirdiğiniz sistemin amacını özetleyiniz.
- Çözüm Yaklaşımı: Mimari, veri yapıları, akış ve ana sınıflar.
- Karşılaşılan Sorunlar ve Çözümler: Teknik engeller ve giderme yöntemleri.
- Sistemin Faydaları ve GUI'nin Faydaları.
- Kaynaklar ve (varsa) Ekler.
- Metin uzunluğu: Min 5–Max 20 sayfa (Times New Roman 12).
- Turnitin ve AI kontrolü: Benzerlik > %20 veya AI > %1 ise ödev 0 olarak notlandırılır.

## 10) Notlandırma

Ana Kategori	Alt Kriter	Açıklama	Ağırlık
<b>Fonksiyonellik</b> (40%)	Temel Gereksinimlerin Karşılanması	login, kayıt, üyelik işlemleri, ders ekleme, rezervasyon akışının eksiksiz çalışması	15%
	Veritabanı Yönetimi	backup.dat dosyasının doğru okunması, yazılması ve verilerin tutarlılığı	10%
	Hata Yönetimi ve Doğrulamalar	Boş alan, hatalı format, mükerrer kayıt vb. durumlarda doğru hata mesajları	10%
	Ekstra Senaryoların Yönetimi	Kullanıcı çıkışı, ekran geçişleri, backup kaydı, veri güncellemeleri	5%
<b>GUI Tasarımı</b> (25%)	Arayüz Düzeni	Ekran hizalamaları, okunabilirlik, uygun font/buton boyutu	10%
	Kullanıcı Etkileşimi	Geri bildirim mesajları, butonların işlevselliği, sezgisel kullanım	10%

Ana Kategori	Alt Kriter	Açıklama	Ağırlık
<b>Kod Kalitesi (15%)</b>	Görsel Uyum ve Tutarlılık	Renk uyumu, ikon kullanımı, genel estetik bütünlük	5%
	Kod Organizasyonu	Sınıf, metot ve değişken isimlendirmeleri, modülerlik	7%
	Yorum Satırları ve Dokümantasyon	Kodun açıklanması, karmaşık fonksiyonların açıklama satırlarıyla desteklenmesi	4%
	Hashing / Dosya İşlemleri Doğruluğu	Kendi hashing algoritmasının güvenli ve doğru uygulanması	4%
<b>Rapor (20%)</b>	İçerik ve Yapı	Sorun tanımı, çözüm yaklaşımı, faydalar ve GUI'nin açıklanması	8%
	Teknik Doğruluk	Sistem davranışlarının raporda doğru şekilde ifade edilmesi	8%
	Yazım Dili ve Biçim	Dilbilgisi, biçimsel bütünlük, sayfa düzeni	4%
<b>Bonus (+10%)</b>	<b>Özellik</b> Yaratıcılık ve Fayda	Ekstra özelliğin özgünlüğü, sistemle bütünleşmesi ve kullanıcıya katkısı	+10%

### 11) Teslimat ve Kurallar

- Kaynak kodlar ve çalıştırılabilir JAR/çalıştırma talimatı.
- backup.dat dosyası (oluşuyorsa).
- Rapor (DOCX veya PDF).
- Kod ve raporlarınız benzersiz olmalı; paylaşım/kopya tespitinde ödeviniz 0 olarak notlandırılacaktır.

Sorularınız için: ders sayfasındaki duyuru kanallarını takip ediniz. Başarılar!