**ALGORİTMA VE PROGRAMLAMA – II DERSİ**

**2022-2023 PROJE ÖDEVİ**

**RAPORU**

Proje Konusu:

Çift bağlı liste veri yapısı kullanımına dayanan ve filmleri yapım yılına göre sıralayan bir film envanter uygulaması geliştirmek.

**Programcı Kataloğu:** 2-7. Sayfalar

**Kullanıcı Kataloğu:** 8-13. Sayfalar

**Programcı Kataloğu:**

**Kullanılan Sınıf ve Metotlar:**

A) **Actor** Sınıfı:

Bu sınıftaki metotlar, **Actor** sınıfının özelliklerine erişmek, özellikleri değiştirmek ve nesneyi metin olarak temsil etmek için kullanılır. Bu sayede Actor nesneleriyle ilgili bilgiler alınabilir, değiştirilebilir ve uygun bir biçimde gösterilebilir.

**1. public Actor(String name, String gender, String nationality)**

Bu metot, **Actor** sınıfının constructor(yapıcı) metotudur. name(isim), gender(cinsiyet) ve nationality(vatandaşlık) parametrelerini alır ve bu parametrelerle bir Actor nesnesi oluşturur.

**2. getter ve setter metotları**

Getter metotları; name, gender ve nationality özelliklerine erişim sağlar. Setter metotları ise bu özellikleri değiştirmek için kullanılır.

**3. public String toString()**

Bu metot, Actor nesnesinin String temsilini döndürmek için kullanılır. name, gender ve nationality özelliklerinin değerlerini birleştirerek okunabilir bir şekilde döndürür. Bu metot, genellikle bir nesneyi metin olarak temsil etmek veya hata ayıklamak amacıyla kullanılır.

B) **Movie** Sınıfı:

Bu sınıftaki metotlar, **Movie** sınıfının özelliklerine erişmek, özellikleri değiştirmek ve nesneyi metin olarak temsil etmek için kullanılır. Ayrıca, actors özelliği bir ArrayList<Actor> türünde olduğundan, bir filmde yer alan oyuncuların listesini elde etmek ve ayarlamak için özel metotlar da sunulmaktadır.

**1.public Movie(int year, String title, String genre, String director, ArrayList<Actor> actors)**

Bu metot, sınıfın constructor metotudur. year, title, genre, director ve actors parametrelerini alır vebu parametrelerle bir Movie nesnesi oluşturur.

**2. getter ve setter metotları**

Getter metotları; year, title, genre, director ve actors özelliklerine erişim sağlar. Setter metotları ise bu özellikleri değiştirmek için kullanılır.

**3. public String soString()**

Bu metot, Movie nesnesinin String temsilini döndürmek için kullanılır. year, title, genre, director ve actors özelliklerinin değerlerini birleştirerek okunabilir bir şekilde döndürür. Bu metot, genellikle bir nesneyi metin olarak temsil etmek veya hata ayıklamak amacıyla kullanılır.

C) **Node** Sınıfı:

Bu sınıftaki metodlar, bir bağlı listedeki düğümlerin özelliklerine erişmek ve düğümleri bağlamak için kullanılır.

**1. public Node(Movie movie)**

Bu metot, sınıfın constructor metotudur. Bir Movie nesnesi alır ve bu nesneyi düğümün filmi olarak ayarlar. Başlangıçta prev ve next değerlerini “null” olarak ayarlar.

**2. getter ve setter metotları**

1. **public Movie getMovie()**
   * Bu metot, düğümün filmini (**movie**) döndürür.
2. **public void setMovie(Movie movie)**
   * Bu metot, düğümün filmini (**movie**) ayarlamak için kullanılır.
   * Bir **Movie** nesnesi alır ve düğümün filmini bu nesneyle günceller.
3. **public Node getPrev()**
   * Bu metot, düğümün bir önceki düğümünü (**prev**) döndürür.
4. **public void setPrev(Node prev)**
   * Bu metot, düğümün bir önceki düğümünü (**prev**) ayarlamak için kullanılır.
   * Bir **Node** nesnesi alır ve düğümün bir önceki düğümünü bu nesneyle günceller.
5. **public Node getNext()**
   * Bu metot, düğümün bir sonraki düğümünü (**next**) döndürür.
6. **public void setNext(Node next)**
   * Bu metot, düğümün bir sonraki düğümünü (**next**) ayarlamak için kullanılır.
   * Bir **Node** nesnesi alır ve düğümün bir sonraki düğümünü bu nesneyle günceller.

D) **DoublyLinkedList** Sınıfı:

Bu sınıftaki metotlar, çift yönlü bağlı listenin temel işlevlerini gerçekleştirmek için kullanılır ve filmlerin eklenmesi, silinmesi, sıralanması, yazdırılması ve dosyaya kaydedilmesi gibi işlemleri sağlar.

**1. isEmpty()**: Bağlı listenin boş olup olmadığını kontrol eder. Eğer baş düğüm null ise, liste boştur ve true değeri döndürür.

**2. addToEmptyList(Movie movie)**: Boş bir listeye film ekler. Baş ve son düğüm olarak yeni bir düğüm oluşturulur.

**3. addSorted(Movie movie)**: Sıralı bir şekilde filme göre düğümü listeye ekler. Eğer liste boş ise addToEmptyList() metodunu çağırarak filmi ekler. Aksi halde, listenin uygun konumuna göre düğümü ekler. Filmin yılına ve başlığına göre sıralama yapar.

**4. addMovie(Movie movie)**: addSorted() metodunu çağırarak filmi listenin uygun konumuna ekler.

**5. removeMovie(String movieTitle)**: Belirtilen film başlığına sahip düğümü listeden kaldırır. İlgili düğümün bağlantıları güncellenir.

**6. printForward()**: Listeyi baştan sona doğru ileriye doğru yazdırır. Her düğümde bir pencere açarak film bilgilerini gösterir.

**7. printBackward()**: Listeyi sondan başa doğru geriye doğru yazdırır. Her düğümde bir pencere açarak film bilgilerini gösterir.

**8. printMoviesBeforeYear(int year)**: Belirtilen yıldan önceki filmleri yazdırır. Her düğümü kontrol ederek, film yılını kontrol eder ve belirtilen yıldan küçük olan filmleri gösterir.

**9. saveToFile(String filename)**: Listedeki filmleri belirtilen dosyaya kaydeder. Dosya adı parametre olarak alınır ve her filmi dosyaya yazarken CSV formatında saklar. Her satırda bir film bulunur ve sütunlar arasında film bilgileri yer alır.

E) **MovieApp** Sınıfı (Main):

Bu sınıf, film envanteri uygulamasının çalışmasını sağlayan bir main metodu içerir. İşlevlerin gerçekleştirilmesi için kullanıcı arayüzü olarak **JOptionPane** sınıfı kullanılmıştır.

**1. main(String[] args)**: Film envanteri uygulamasının ana metodu. Kullanıcıya bir menü gösterir ve kullanıcının seçimine göre ilgili işlevleri çağırır. Seçenekler arasında metin dosyasından filmlerin okunması, yeni film eklenmesi, film bilgilerinin gösterilmesi, film silinmesi, filmlerin tarihlerine göre sıralanması ve istenilen tarihten önceki filmlerin görüntülenmesi gibi işlemler bulunur. Kullanıcı çıkış seçeneğini seçene kadar döngü devam eder.

**2. displayMenu()**: Kullanıcıya hoş geldiniz mesajını gösteren bir iletişim kutusu görüntüler.

**3. readFromFile(DoublyLinkedList filmInventory, String filename)**: Metin dosyasından film bilgilerini okuyarak bir çift yönlü bağlı listede depolar. Dosya adı ve film envanteri nesnesi parametre olarak alınır. Dosya okunurken her satırda bir film bilgisi bulunur ve bu bilgiler ayrıştırılarak Movie ve Actor nesneleri oluşturulur. Oluşturulan film nesnesi, filmInventory nesnesine eklenir.

**4. addFilm(Scanner scanner, DoublyLinkedList filmInventory)**: Kullanıcıdan film bilgilerini alarak yeni bir film ekler. Scanner ve film envanteri nesnesi parametre olarak alınır. Kullanıcıdan film yapım yılı, adı, türü, yönetmeni ve oyuncularının bilgileri istenir. Oyuncuların bilgileri birden fazla kez alınabilir. Girilen bilgiler kullanılarak Movie ve Actor nesneleri oluşturulur ve film envanteri nesnesine eklenir.

**5. displayFilm(Scanner scanner, DoublyLinkedList filmInventory)**: Kullanıcıdan bir film adı alarak ilgili film bilgilerini gösterir. Scanner ve film envanteri nesnesi parametre olarak alınır. Film envanterindeki her düğüm kontrol edilir ve girilen film adına sahip film bulunursa bilgileri kullanıcıya gösterilir.

**6. removeFilm(Scanner scanner, DoublyLinkedList filmInventory)**: Kullanıcıdan bir film adı alarak ilgili filmi listeden siler. Scanner ve film envanteri nesnesi parametre olarak alınır. Film envanterindeki removeMovie() metodu çağrılarak film silinir.

**7. displayFilmsBeforeYear(Scanner scanner, DoublyLinkedList filmInventory)**: Kullanıcıdan bir yıl alarak belirtilen yıldan önceki filmleri gösterir. Scanner ve film envanteri nesnesi parametre olarak alınır. Film envanterindeki printMoviesBeforeYear() metodu çağrılarak belirtilen yıldan önceki filmler listelenir.

Bu şekilde, **MovieApp** sınıfı, kullanıcıdan girdiler alarak film envanteriyle ilgili işlemleri gerçekleştiren bir uygulama sağlar.

Tasarım aşamasında kullanılacak olan sınıfların özellikleri belirlendi, bu sınıflara yazılacak metotlar planlandı ve bağlantılar düzenlendi.

Kod yazma aşamasında belirlenen özellikler ve metotlar kodlandı ve uygulandı.

Test aşamasında ise kodun doğru çalıştığından emin olmak için deneme ve hata ayıklama süreci gerçekleşti.

Hata uyarısı çıkmadan karşılaşılan ve programın düzgün bir şekilde çalışmasını engelleyen hatalar debug modu çalıştırılarak adım adım izlendi ve hatanın nereden kaynaklandığı tespit edilip çözüldü.

**Tasarım süresi:** 2 Saat 15 Dakika

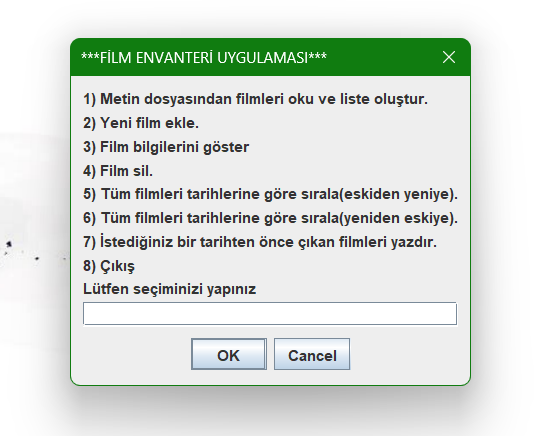
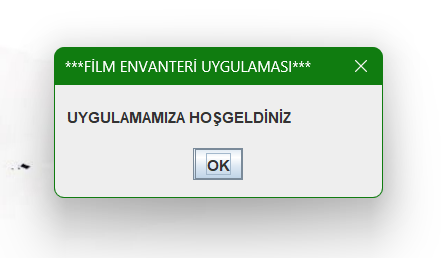
**Gerçekleştirme süresi:** 1 Saat

**Test süresi:** 6 Saat 30 Dakika

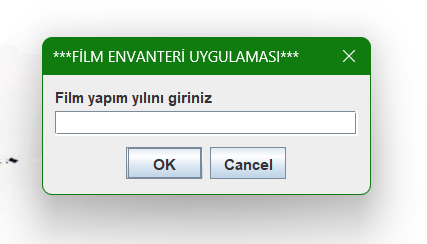
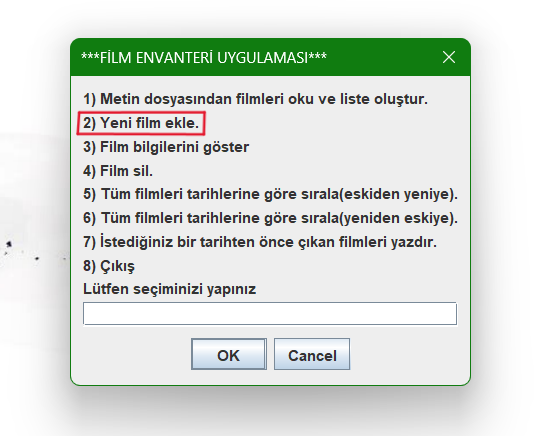
**Kullanıcı Kataloğu:**

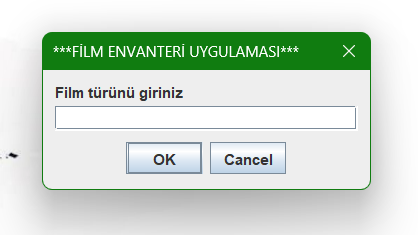
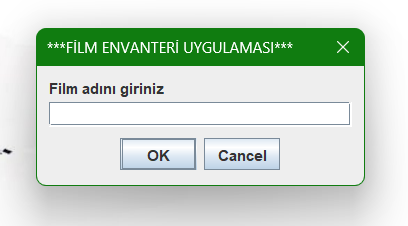
Programın kullanım kılavuzunu aşağıdaki ekran görüntüleriyle birlikte hazırladık:

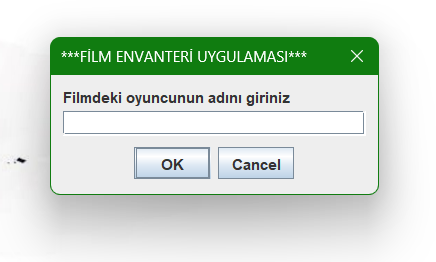
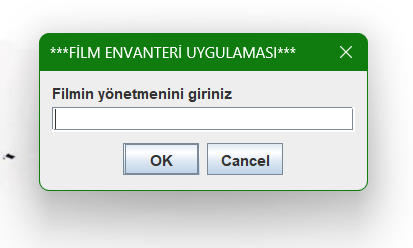
* [Ekran görüntüsü 1: Programın ana ekranı]:

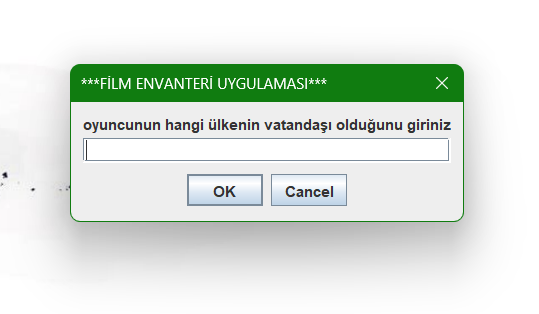
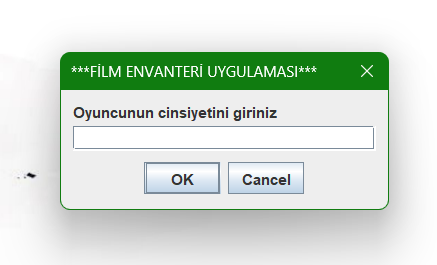


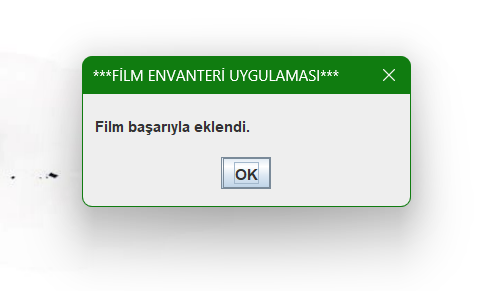
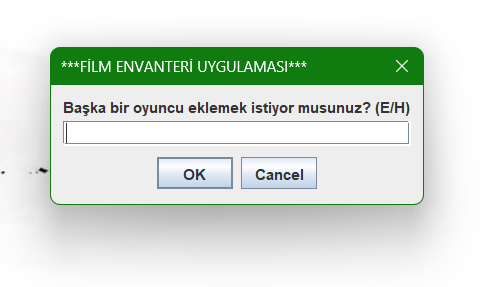
* [Ekran görüntüsü 2: Yeni bir film eklemek için kullanım adımları]:



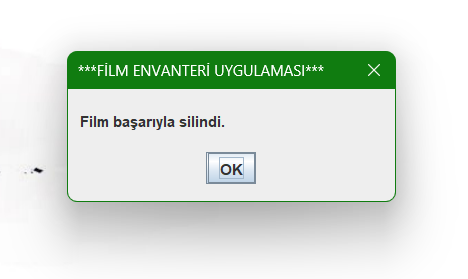
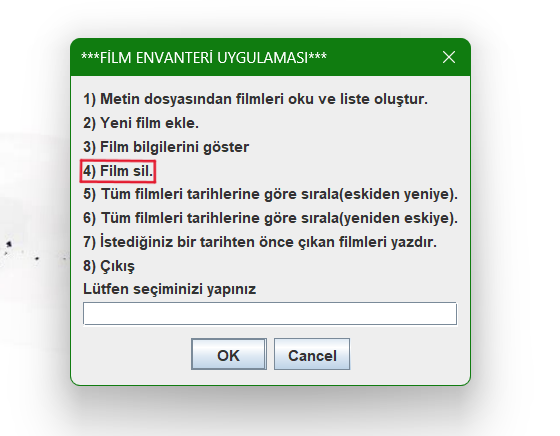




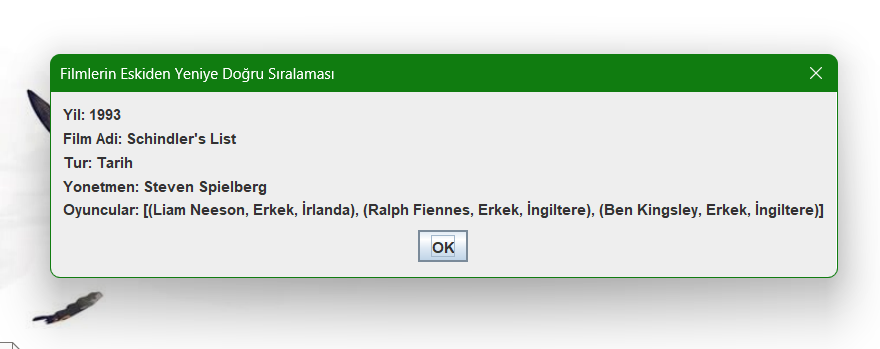
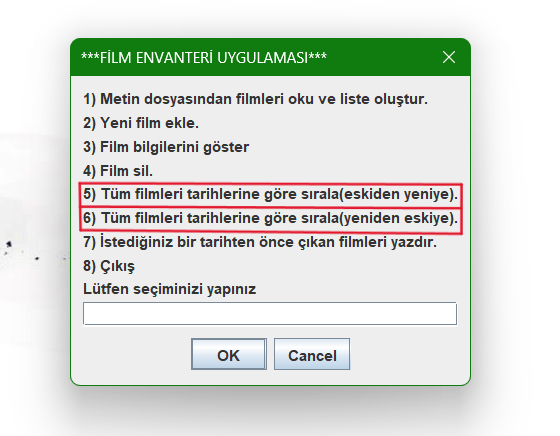




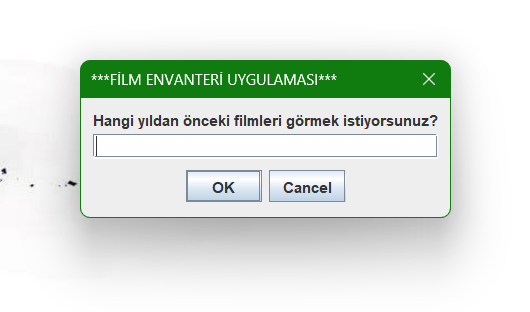
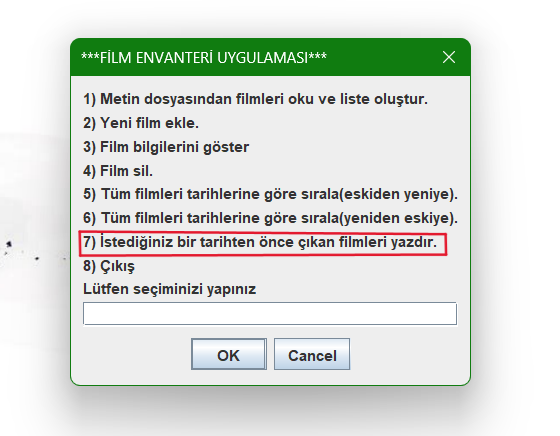
* [Ekran görüntüsü 3: Film silme işlemi için kullanım adımları]:

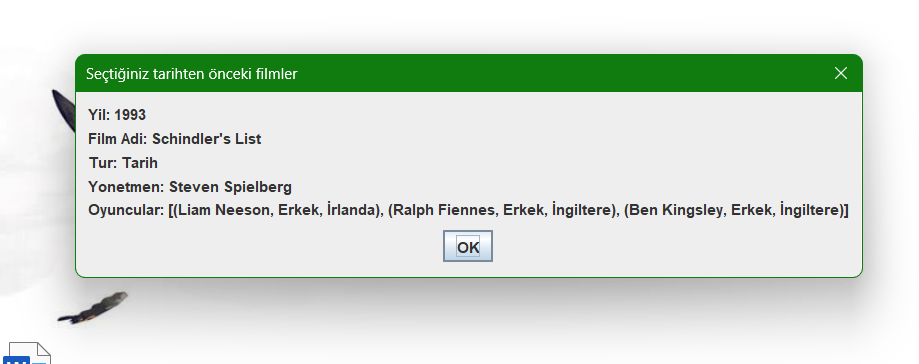


* [Ekran görüntüsü 4: Filmleri listeleme işlemi]:

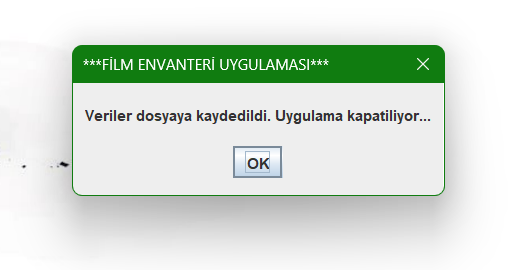
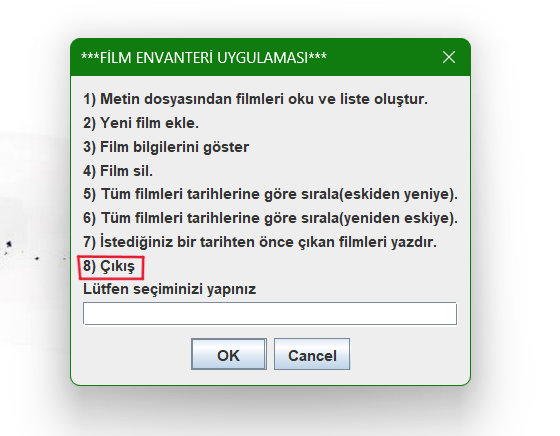


* [Ekran görüntüsü 5: Belirli bir tarihten önce çıkan filmleri listeleme işlemi]:





* [Ekran görüntüsü 6: Çıkış işlemi]:



Programdaki kısıtlamalar:

1. Ana ekrandaki seçimlerinizi 1-8 arasındaki rakamları kullanarak yapmalısınız.

2. Yeni film eklerken filmin yapım yılın değerini yazı ile deği rakamlarla girmelisiniz.

3. Film sildirme işleminde listede var olan filmlerden sildirebilirsiniz aksi takdirde silme işlemi gerçekleşmeyecek ve bununla ilgili bilgilendirme meesajı alacaksınız.