T.C. İSTANBUL MEDİPOL ÜNİVERSİTESİ

MESLEK YÜKSEK OKULU

Bütünleme ÖDEV RAPORU

Aşağıda başlığı belirtilen Bilgisayar Programcılığı 2. sınıf Bahar Dönemi Bilgisayar Programcılığı programındaki İleri Java Uygulamaları Final(Bütünleme) ödev

biçimindeki projesi çalışmamızı 27/03/2025 tarihinde, İleri Java Uygulamaları dersinden

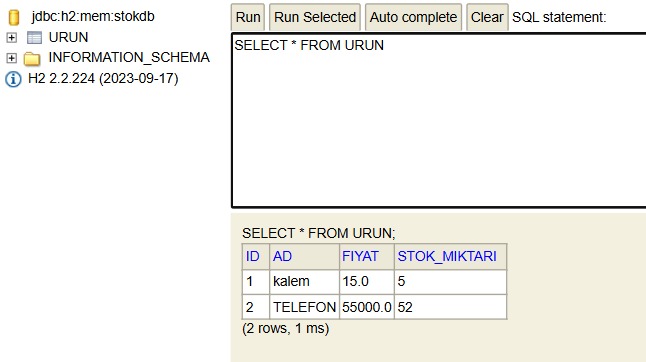
sorumlu

öğretim görevlisi olan Hüseyin Kınay adlı öğretmenimize proje raporunu sunmuş bulunmaktayız.

Genel sınav proje çalışmamın herhangi bir benzerlik içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini saygılarımla arz ederim.

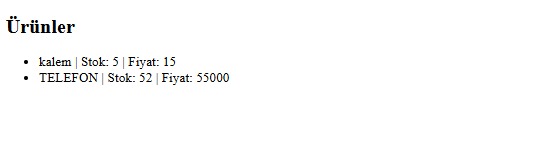
|  |  |
| --- | --- |
| Sorumlu Olunan Ders | İleri Java Uygulamaları |
| Sorumlu Öğretmen | Hüseyin Kınay |
| Öğrenci Adı ve Soyadı | Emirhan Şimşek |
| Öğrenci Numarası | HR230016 |
| Öğrenci Programı | Bilgisayar Programcılığı (İ.Ö.) |
| Sınıf | İkinci (2.) Sınıf |



Uygulamada, H2 bellek içi veritabanı (stokdb) kullanılarak ürün bilgileri URUN tablosunda saklanmaktadır. Yapılan sorgu sonucunda tabloda 2 adet ürün bulunduğu görülmüştür. Bu ürünler:

* **Kalem:** 15 TL, stokta 5 adet
* **Telefon:** 55.000 TL, stokta 52 adet

Bu veriler sayesinde kullanıcı, mevcut ürünlerin fiyatlarını ve stok miktarlarını kolayca görüntüleyebilmektedir. Sistem, ürün takibini sade ve anlaşılır bir arayüz üzerinden sağlamaktadır.



Web arayüzü üzerinde ürün bilgileri sade bir liste formatında sunulmaktadır. Başlık altında, her bir ürün adıyla birlikte stok miktarı ve fiyat bilgisi yer almaktadır. Görselde görüldüğü üzere:

* **kalem** | Stok: 5 | Fiyat: 15
* **TELEFON** | Stok: 52 | Fiyat: 55000

Bu yapı sayesinde kullanıcılar ürünleri hızlıca inceleyebilir, stok ve fiyat durumuna kolayca ulaşabilir. Basit ve kullanıcı dostu bir tasarım tercih edilmiştir.



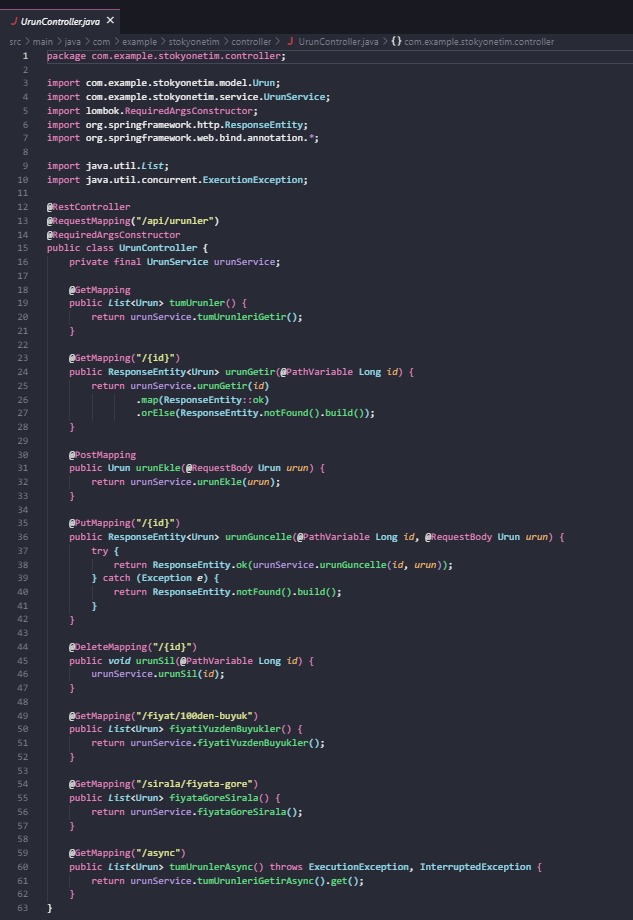
Bu sınıf, Spring Boot uygulamasında başlangıçta otomatik olarak çalışan bir bileşendir. Uygulama ayağa kalktığında bazı örnek işlemleri çalıştırarak hem temel Java kavramlarını hem de modern programlama tekniklerini kullanıcıya göstermektedir.

İlk olarak, kullanıcıya bir döngü örneği sunularak 1’den 5’e kadar olan sayıların toplamı hesaplanmaktadır. Bu, değişken kullanımı ve döngü mantığını kavramak açısından oldukça öğreticidir.

Ardından, generic yapının örneği verilmiştir. Kullanıcı, burada tip bağımsız bir şekilde hem metin hem de sayısal verilerin ekrana nasıl yazdırılabileceğini gözlemleyebilir. Bu durum, yazılan fonksiyonların esnekliğini ve tekrar kullanılabilirliğini artırmaktadır.

Bir sonraki bölümde, modern Java’nın sunduğu lambda ifadeleri ve Stream API kullanılarak ürün listesinden belirli koşullara uyan verilerin nasıl filtrelenip yazdırılabileceği gösterilmektedir. Örnekte, fiyatı belli bir değerin üzerinde olan ürünler süzülerek kullanıcıya sunulmuştur. Bu yaklaşım, gerçek dünyadaki ürün filtreleme sistemleriyle benzer mantıktadır ve işlevsel programlamanın gücünü ortaya koyar.

Son olarak, bir arama algoritması örneğiyle kullanıcıya dizilerde veri arama işleminin nasıl yapılacağı anlatılmıştır. Basit bir doğrusal arama algoritmasıyla belirli bir karakterin listedeki konumu bulunmuş ve sonuç olarak ekrana yazdırılmıştır.



Haklısınız, özür dilerim. Bazen detaylara çok giriyorum.

Kısa ve öz bir şekilde, o kod görselindeki UrunController.java dosyası bana şunları anlatıyor:

**Kullanıcı Gözünden, Kısaca:**

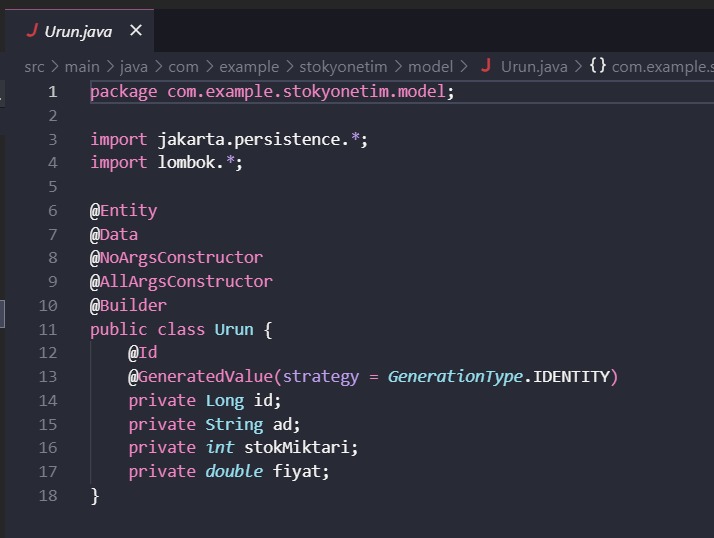
Bu kod, benim gibi bir kullanıcının bu stok uygulamasındaki ürünlerle neler yapabileceğini belirleyen bir "komut merkezi".

* **Tüm Ürünleri Göster:** Bir komut vererek tüm ürün listemi görebilirim.
* **Belirli Bir Ürünü Bul:** Ürünün kimlik numarasını vererek sadece o ürüne bakabilirim.
* **Yeni Ürün Ekle:** Stoğuma yeni bir ürün geldiğinde, bilgilerini vererek ekleyebilirim.
* **Ürünü Güncelle:** Mevcut bir ürünün bilgilerini (stok, fiyat vb.) değiştirebilirim.
* **Ürünü Sil:** İstemediğim ürünleri listeden çıkarabilirim.

Ayrıca, bazı akıllı kısa yollar da var:

* **Pahalı Ürünleri Göster:** Fiyatı belli bir miktardan yüksek olanları anında listeleyebilirim.
* **Fiyata Göre Sırala:** Ürünleri ucuzdan pahalıya sıralayabilirim.
* **Hızlı Listeleme:** Çok ürün varsa, listeyi daha çabuk görmemi sağlayan bir seçenek de var.

Kısacası, bu kod benim ürün stokumu yönetmek için bir dizi basit ve kullanışlı "talimat" sunuyor.



* Sorgu başarılı

Elbette, sadece bu görseldeki Urun.java kodunu bir kullanıcının ağzından anlatayım, yine kısa ve öz bir şekilde:

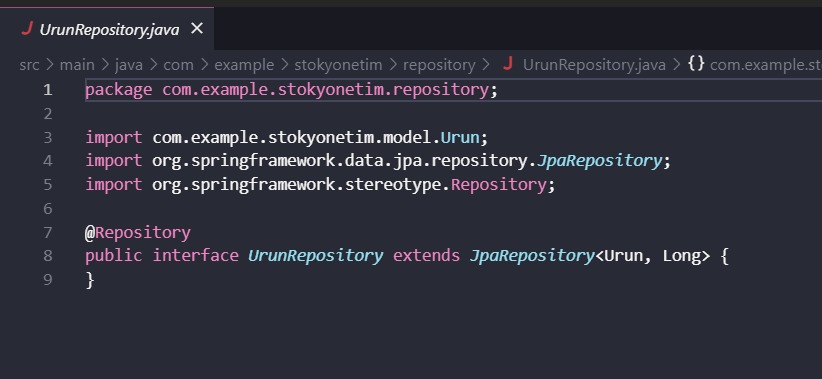
**Bu Kod Bana Ne Anlatıyor? (Bir Kullanıcının Gözünden, Ürün Bilgisi)**

Önümdeki bu kod parçası (Urun.java dosyası), benim için çok temel bir şeyi tanımlıyor: **Bir ürünün tam olarak ne olduğunu ve hangi bilgilere sahip olabileceğini.**

Sanki bir ürün bilgi kartının şablonu gibi:

* **Benzersiz Bir Kimliği Var (ID):** Her ürünün kendine ait bir numarası olacak. Bu numara, sistemin ürünleri karıştırmamasını sağlıyor ve ben de o numarayla ürünü kolayca bulabilirim.
* **Bir Adı Var:** Her ürünün bir ismi olacak, mesela "Kalem", "Defter" veya "Telefon".
* **Stok Miktarı Var:** Kaç tane üründen stokta bulunduğumun bilgisi tutulacak.
* **Bir Fiyatı Var:** Her ürünün bir satış fiyatı olacak.

Bu kod bana diyor ki: "Bir ürün kaydettiğinde, bu dört temel bilgi (ID, Ad, Stok Miktarı, Fiyat) mutlaka olacak." Sanki bir form doldurur gibi, her yeni ürün için bu alanları doldurmam gerektiğini anlıyorum. Bu, tüm ürün bilgilerimin düzenli ve standart olmasını sağlıyor, ki bu da benim için önemli.

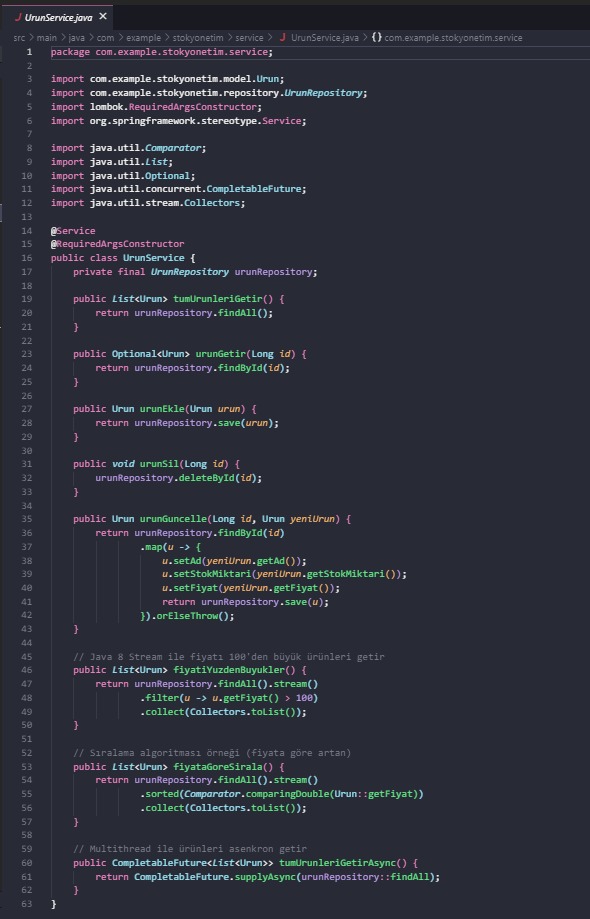


Bu kod parçası (UrunRepository.java), benim için aslında bir "depo müdürü" gibi.

Bana diyor ki:

* **"Ürün bilgilerini nereye kaydedeceğimi ve nereden alacağımı ben bilirim."**
* **"Ürünleri kaydetme, bulma, güncelleme veya silme gibi tüm işlemleri ben hallederim, sen bana ne istediğini söylemen yeterli."**

Yani, ben bir ürünü eklemek, bulmak veya silmek istediğimde, uygulamanın diğer kısımları doğrudan veritabanıyla uğraşmak zorunda kalmıyor. Sadece bu "depo müdürüyle" konuşuyorlar, o da tüm zor işleri benim adıma yapıyor. Bu sayede, hem işler düzenli ilerliyor hem de verilerimin güvenli bir şekilde saklandığından emin olabiliyorum.



Bu kod parçası (UrunService.java), benim için aslında bir "ürün danışmanı" veya "işlem yöneticisi" gibi.

Bana diyor ki:

* **"Benden ne istersen, ben senin için o işi hallederim."**
* **"Tüm ürünleri listelemek mi istiyorsun? Ben getiririm."**
* **"Belirli bir ürünü bulmak mı istiyorsun? Ben bulurum."**
* **"Yeni ürün mü ekleyeceksin? Bana söyle, ben kaydederim."**
* **"Bir ürünün bilgilerini mi güncelleyeceksin? Yeni bilgileri bana ver, ben düzenlerim."**
* **"Bir ürünü silmek mi istiyorsun? Bana söyle, ben silerim."**

Ayrıca, daha akıllı işler de yapıyor:

* **"Fiyatı yüksek olanları mı görmek istiyorsun? Ben sana sadece onları listelerim."**
* **"Ürünleri fiyata göre sıralamak mı istiyorsun? Ben senin için sıraya dizerim."**
* **"Çok ürün olduğunda bile listeyi hızlıca getirmek için farklı yollar bilirim."**

Kısacası, bu kod benim ürünlerimle ilgili tüm karmaşık işlemleri yöneten, bana sadece ne yapmak istediğimi söylemem yeterli olan bir arka plan uzmanı. Tüm detaylarla o ilgileniyor, benim işim kolaylaşıyor.

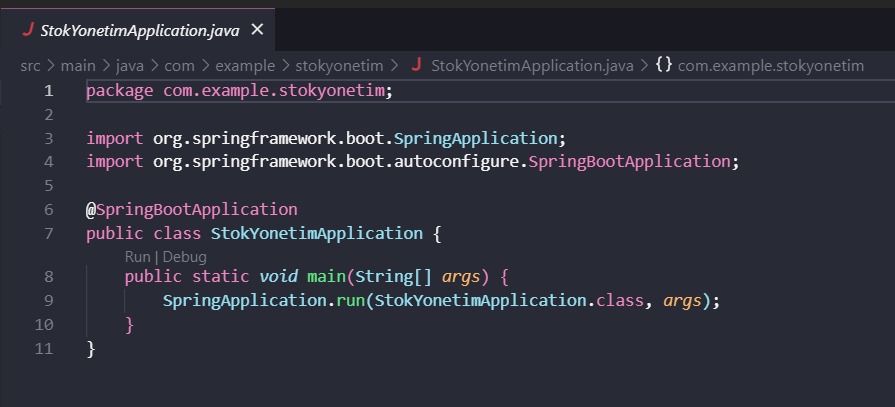
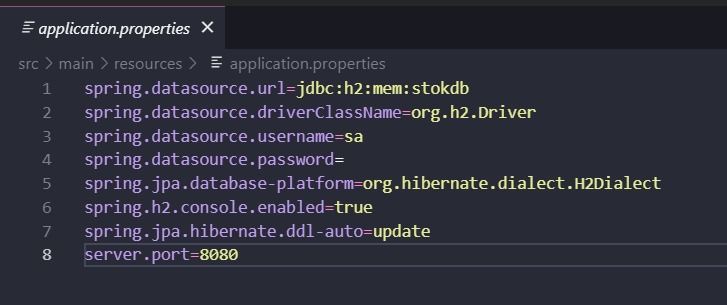


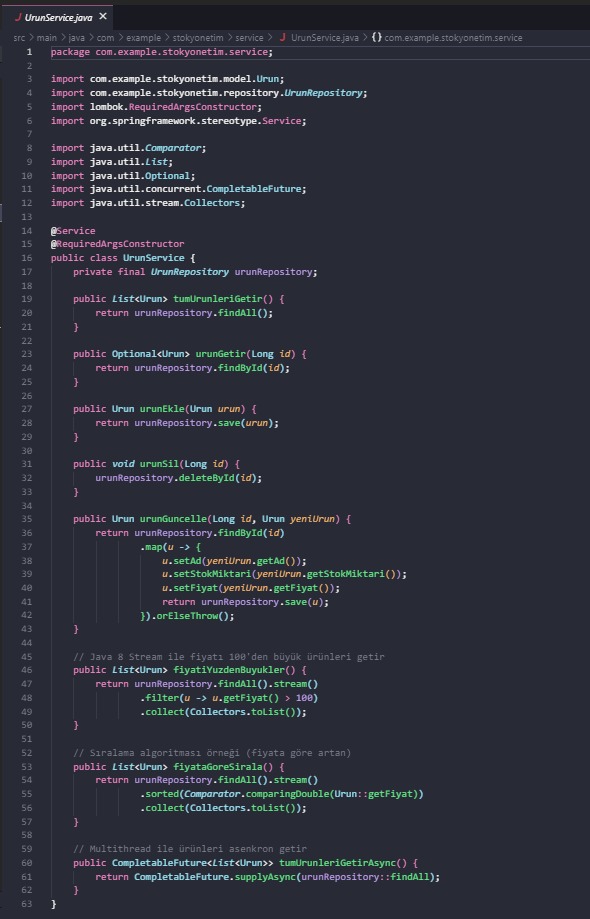
Bu kod parçası (pom.xml dosyası), benim için adeta bu ürün yönetim uygulamasının "tarif defteri" gibi.

Bana diyor ki:

* **"Ben bir proje yapacağım, adı 'stok-yonetim'."**
* **"Bu proje Java ile yazılacak."**
* **"Bu, Java'nın ileri konularını kapsayan basit bir stok yönetim uygulaması olacak."**
* **"Bu projeyi yaparken bazı 'malzemelere' ihtiyacım var. Bunlar benim için gerekli araçlar, kütüphaneler ve eklentiler."**

Yani, bir ürün yaparken hangi malzemelerin (örneğin Spring Boot, veritabanı, Lombok gibi) kullanılacağını ve projenin nasıl birleştirileceğini (derleneceğini) açıklayan bir kullanım kılavuzu gibi. Ben bir kullanıcı olarak, bu dosya sayesinde uygulamanın ne kadar modern araçlarla yapıldığını ve hangi temel bileşenlere sahip olduğunu anlıyorum. Benim için arka planda her şeyin düzenli ve doğru bir şekilde kurulduğunu gösteriyor.





Şu an elimdeki tüm bu kod ve ekran görüntülerine baktığımda, anladığım şey şu: Ben bir **stok yönetimi sistemiyle** karşı karşıyayım. Bu sistem, benim ürünlerimi düzenli ve kolayca takip etmemi sağlıyor.

**Genel olarak bana şunları sunuyor:**

1. **Ürün Bilgilerini Tutuyor:** Her bir ürünün (mesela kalem veya telefon) kendine ait bir ID'si, adı, stok miktarı ve fiyatı var. Yani, her ürünün bir "kimlik kartı" mevcut.
2. **Ürünlerle İstediğim İşlemleri Yapabiliyorum:**
   * **Listeleme:** Tüm ürünlerimi görebiliyorum. Hatta sadece fiyatı belli bir miktarın üstünde olanları veya fiyata göre sıralanmış bir listeyi bile alabiliyorum. (Ekran görüntüsündeki "Ürünler" listesi bunu kanıtlıyor!)
   * **Bulma:** Belirli bir ürünü hızlıca bulabiliyorum.
   * **Ekleme:** Stoğuma yeni ürünler kaydedebiliyorum.
   * **Güncelleme:** Ürünlerin stok miktarını veya fiyatını değiştirebiliyorum.
   * **Silme:** Artık ihtiyacım olmayan ürünleri sistemden kaldırabiliyorum.
3. **Arka Planda Ne Oluyor?**
   * Bu işlemlerin hepsi, benim görmediğim bir "veri deposu"nda (veritabanı) saklanıyor. (H2 konsolundaki "URUN" tablosu bunu gösteriyor!)
   * Sistem, bu depoyla benim adıma konuşan "uzman servisler" ve "kontrol panelleri" sayesinde çalışıyor.
   * Uygulama, kendi ayarları (8080 portu gibi) ve kullandığı "malzemeler" (Spring Boot gibi teknolojiler) ile sorunsuz çalışıyor.

Kısacası, bu uygulama benim **stoklarımı dijital ortamda yönetmem için tasarlanmış, basit, düzenli ve işlevsel bir araç.** Ürünlerimi takip etmek ve düzenlemek artık çok daha kolay!