Depo Yönetim Sistemi - Kod Analizi ve Raporu

Bu raporda, Java ile yazılmış bir Depo Yönetim Sistemi uygulaması analiz edilmiştir.  
Uygulama; ürünlerin eklenmesi, listelenmesi, güncellenmesi ve raflara yerleştirilmesi gibi temel depo yönetimi işlevlerini yerine getirir.  
Rapor içerisinde her sınıfın işlevi ve önemli kod parçaları açıklamalı olarak sunulmuştur.

# 1. Product.java - Veri Modeli Sınıfı

`Product` sınıfı, depo sisteminde bir ürünü temsil eder ve ürünle ilgili verileri kapsülleyen özelliklere sahiptir.  
  
**- \*\*Özellikler\*\*:** - `id`: Her ürün için benzersiz bir kimlik.  
 - `counter`: Her ürün için benzersiz `id` oluşturmak için kullanılan sayıcı, 1000'den başlar.  
 - `productName`: Ürünün adı.  
 - `productorName`: Ürünün üreticisi.  
 - `quantity`: Stoktaki ürün miktarı.  
 - `part`: Ürün birimi.  
 - `shelf`: Ürünün depoda bulunduğu raf.

### Örnek Kod:

public class Product {  
 private String id;  
 public static int counter = 1000;  
 private String productName, productorName;  
 private int quantity;  
 private String part, shelf;  
  
 public Product(String id, String productName, String productorName, int quantity, String part, String shelf) {  
 this.id = id;  
 this.productName = productName;  
 this.productorName = productorName;  
 this.quantity = quantity;  
 this.part = part;  
 this.shelf = shelf;  
 }  
 // Getter and Setter methods...  
}

# 2. ProductService.java - Servis Sınıfı

`ProductService` sınıfı, `Product` varlıklarıyla ilgili iş mantığını yönetir ve ürün ekleme, güncelleme, listeleme gibi işlemleri gerçekleştirir.  
  
- \*\*Özellikler\*\*:  
 - `products`: `LinkedHashMap` olarak tanımlanmış ve ürünlerin `id` ile depolanmasını sağlar.  
 - `sc`: Kullanıcıdan girdi almak için kullanılan `Scanner` nesnesi.  
  
- \*\*Metotlar\*\*:  
 - `addProduct`: Kullanıcıdan ürün bilgilerini alarak yeni bir ürün ekler. Aynı ürün zaten mevcutsa, miktarın güncellenmesi önerilir.  
 - `productId`: Ürün `id`’sini üretir; ürün adının ilk iki harfi, yıl ve sayaç ile birleştirilerek benzersiz `id` oluşturur.  
 - `listProduct`: Tüm ürünleri tablo formatında listeler.

### Örnek Kod:

public void addProduct(Map<String, Product> products) {  
 Product pr = new Product(null, null, null, 0, null, null);  
 System.out.print("Enter a product name: ");  
 String productName = sc.nextLine().toUpperCase().trim();  
 System.out.print("Enter a productor name: ");  
 String productorName = sc.nextLine().toUpperCase().trim();  
 productId(pr);  
 products.put(pr.getId(), pr);  
}  
  
public void productId(Product pr) {  
 pr.setId(pr.getProductName().substring(0, 2) + LocalDate.now().getYear() + Product.counter);  
 Product.counter++;  
}

# 3. DepoApp.java - Ana Uygulama Sınıfı

`DepoApp` sınıfı, kullanıcı arayüzünü sağlar ve `ProductService` ile etkileşimde bulunarak işlemleri yürütür.  
  
- \*\*Metotlar\*\*:  
 - `mainMenu`: Ana menü fonksiyonunu sağlar.  
 - `promptReturnToMenu`: Ana menüye dönmeden önce kullanıcıya beklemesini sağlar.

### Örnek Kod:

public class DepoApp {  
 public static void main(String[] args) {  
 mainMenu();  
 }  
 public static void mainMenu() {  
 ProductService ps = new ProductService();  
 System.out.println("========= Depo Management System =========");  
 System.out.println("1- Define a product");  
 System.out.println("2- List products");  
 System.out.print("Select an Option: ");  
 int select = scan.nextInt();  
 switch (select) {  
 case 1 -> ps.addProduct(ps.products);  
 case 2 -> ps.listProduct(ps.products);  
 default -> System.out.println("Invalid selection.");  
 }  
 }  
}

# Program Akışı

1. `DepoApp`'de `mainMenu()` fonksiyonu ile kullanıcıya işlem seçenekleri sunulur.  
2. Kullanıcı ürün ekleme, listeleme, miktar güncelleme, rafa yerleştirme veya ürünü azaltma işlemlerinden birini seçebilir.  
3. `ProductService` sınıfında her bir işlev için uygun metod çağrılır.

# Öneriler

- \*\*Hata Yönetimi\*\*: Daha fazla hata yönetimi eklenebilir.  
- \*\*Gelişmiş Arama Özelliği\*\*: Ürün adına göre arama yapılabilir.  
- \*\*Veri Kalıcılığı\*\*: Program kapanıp açıldığında verilerin saklanması sağlanabilir.  
  
Bu, sistemin genel işlevselliğini artıracak ve daha esnek bir depo yönetimi uygulaması sağlayacaktır.