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.  
  
 Sınıfların Görevleri ve Analiz  
Bu uygulama üç ana sınıftan oluşmaktadır:

`Product`,

`ProductService`, ve `DepoApp`.

Her sınıfın görevi aşağıda detaylandırılmıştır.  
  
 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.  
  
- Constructor:  
 - Ürün özelliklerini başlatan bir yapıcı fonksiyon.  
  
- Getters ve Setters:  
 - Her özellik için diğer sınıfların bu özelliklere erişmesine ve değiştirmesine olanak tanıyan getter ve setter metotları tanımlanmıştır.  
  
 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.  
 - **`enterProduct**`: Mevcut ürünün miktarını günceller.  
 - **`putProductOnShelf**`: Ürünü bir raf numarasına yerleştirir.  
 - **`productOutput`**: Ürünün miktarını azaltır.  
  
**3. DepoApp.java - Ana Uygulama Sınıfı**  
`DepoApp` sınıfı, kullanıcı arayüzünü sağlar ve `ProductService` ile etkileşimde bulunarak işlemleri yürütür.  
  
**- Metotlar:**  
 - **`mainMenu**`: Ana menüyü görüntüler ve kullanıcının seçimine göre işlemleri gerçekleştirir.  
 - `**promptReturnToMenu`**: Ana menüye dönmeden önce kullanıcıya beklemesini sağlar.  
  
 **Programın Çalışma 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.  
  
**Kullanılan Metotların Açıklamaları**  
  
**- Benzersiz ID Üretimi (`productId`):** - `productId` metodu, ürün adına, yıla ve sayaç değerine dayalı benzersiz bir `id` üretir.  
  
- **Map Yapısı (`products`):** - `products` için `LinkedHashMap` kullanılarak, ürünlerin eklenme sırasına göre listelenmesi sağlanır.  
  
**- Kullanıcı Girdi Doğrulama:** - Sayısal değerler için geçerli sayı olup olmadığı ve pozitiflik kontrolü yapılır.  
  
 **Öneriler**  
**- Hata Yönetimi** : Daha fazla hata yönetimi eklenebilir.  
- **Gelişmiş Arama Özelliği**: Ürün adına göre arama yapılabilir.  
- **Veri Kalıcılığı**: Program kapanıp açıldığında verilerin saklanması için veri kalıcılığı sağlanabilir.  
  
Bu, sistemin genel işlevselliğini artıracak ve daha esnek bir depo yönetimi uygulaması sağlayacaktır.