Market Uygulaması Proje Dökümanı

Yusuf İslam Yılmaz - 220306019

Bu doküman, bizim Market Uygulaması'nın nasıl çalıştığını ve neler yaptığını anlatıyor. Uygulama C# dilinde yazıldı ve iki bölümden oluşuyor:

1. **MarketConsoleApp**: Kullanıcının konsoldan işlemler yapabildiği ana bölüm.
2. **MarketLibrary**: Uygulamanın arka planda çalışan kısımları, yani iş mantığı ve sınıflar.

# I. Uygulamanın Özellikleri

## Ürün Ekleme

Kullanıcı, markete şu türde ürünler ekleyebiliyor:

* **Temel Gıda**
* **İçecek**
* **Atıştırmalık**

Her ürün için aşağıdaki bilgiler isteniyor:

* Ürün Adı
* Fiyat
* Stok Miktarı
* Barkod Numarası (Benzersiz olmalı)

## Ürünleri Görüntüleme

Marketin içindeki tüm ürünler liste halinde görüntüleniyor. Listeleme sırasında şunlar gösteriliyor:

* Türü (örneğin, Temel Gıda, İçecek)
* Barkod Numarası
* Adı
* Fiyatı
* Stok Miktarı

## Ürün Satışı

* Kullanıcı, satmak istediği ürünün barkod numarasını ve miktarını giriyor.
* Eğer stok yeterliyse satış başarılı oluyor ve stok düşüyor.
* Yetersiz stok veya yanlış barkod numarası girildiğinde hata mesajı veriliyor.

## Çıkış

Kullanıcı istediği zaman uygulamadan çıkış yapabiliyor.

# II. OOP (Nesne Tabanlı Programlama) Prensipleri

## Encapsulation (Kapsüllüme)

* Ürün sınıfındaki özellikler (Ad, Fiyat, Stok, Barkod) private olarak tanımlandı.
* Bu özelliklere sadece getter ve setter metodlarıyla erişiliyor. Böylece yanlış veya geçersiz veri girişi engelleniyor (örneğin, negatif fiyat).

## Abstraction (Soyutlama)

* Urun sınıfı, diğer ürün türlerinin ortak özelliklerini barındıran bir "şablon" gibi çalışıyor.
* Urun sınıfına UrunBilgisi adında bir soyut metot eklendi. Bu metot, türetilen sınıflar tarafından kendi ihtiyaçlarına göre dolduruluyor.

## Polymorphism (Çok Biçimlilik)

* UrunBilgisi metodu sayesinde, her ürün türü (örneğin, Temel Gıda, İçecek) kendine özel bilgileri farklı şekilde gösterebiliyor.

## Inheritance (Kalıtım)

* TemelGida, Icecek, Atistirmalik sınıfları, Urun sınıfından türetilmiştir. Böylece bu sınıflar Urun sınıfındaki özellikleri ve metodları kullanabiliyor.

# III. Dependency Inversion Prensipleri

* **IUrun** adında bir arayüz (interface) eklendi. Tüm ürün sınıfları bu arayüzü kullanıyor.
* Market sınıfı, doğrudan Urun sınıfına bağlı değil. Bunun yerine IUrun arayüzüne bağlı. Bu sayede yeni bir ürün türü eklemek gerektiğinde, Market sınıfını değiştirmemize gerek kalmıyor.

# IV. Uygulama Nasıl Çalışıyor

1. Uygulama başladığında bir menü ekrana geliyor.
2. Kullanıcı menüden şu işlemlerden birini seçebiliyor:
   1. Ürün Ekle
   2. Ürünleri Görüntüle
   3. Ürün Sat
   4. Çıkış
3. Kullanıcı ürün eklemek isterse, ürün türünü ve özelliklerini seçiyor. Bu ürün markete ekleniyor.
4. Kullanıcı ürünleri görüntülemek isterse, marketteki tüm ürünlerin bilgileri listeleniyor.
5. Kullanıcı ürün satışı yapmak isterse, barkod numarasını ve satılacak miktarı giriyor. Eğer her şey uygunsa satış yapılıyor.
6. Kullanıcı çıkış seçeneğini seçtiğinde program kapanıyor.

# V. Örnek Kullanım

### Başlangıç:

Market Uygulamasına Hoş Geldiniz!

Menü:

1. Ürün Ekle

2. Ürünleri Görüntüle

3. Ürün Sat

4. Çıkış

Bir seçenek seçin:

### Ürün Ekleme:

Ürün Türünü Seçin:

1. Temel Gıda

2. İçecek

3. Atıştırmalık

Seçiminiz: 1

Ürün adını girin: Ekmek

Ürün fiyatını girin: 5

Ürün stok miktarını girin: 50

Ürün barkod numarasını girin: 123456

Ürün başarıyla eklendi.

### Ürünleri Görüntüleme:

Mevcut Ürünler:

[Temel Gıda] Barkod: 123456, Ürün: Ekmek, Fiyat: 5 TL, Stok: 50 Adet

### Ürün Satışı:

Mevcut Ürünler:

[Temel Gıda] Barkod: 123456, Ürün: Ekmek, Fiyat: 5 TL, Stok: 50 Adet

Satılacak ürünün barkod numarasını girin: 123456

Satılacak miktarı girin: 10

Ürün başarıyla satıldı.

# VI. Sonuç

Bu program, basit bir market yönetimi uygulaması. İçinde OOP prensipleri ve Dependency Inversiont prensibi var. Uygulama hem düzenli hem de ileride kolayca geliştirilebilecek şekilde yazıldı.