

1: Dosya Yapısı ve Paketleme

model/	# Entity sınıfları (Book, User, Category vs)
service/	# İş mantığı (BookService vs)
data/	# Geçici veri yapıları (InMemoryDB gibi)
util/	# Yardımcı sınıflar (InvoiceGenerator vs)
main/	# Çalıştırılabilir sınıf (Main.java)

2. Paketlerin Görevleri

- model: Tüm sınıflar burada olacak (Book, Author, User, Invoice vs).
- service: Kitap ekleme, silme, ödünç alma gibi işlemleri yapan sınıflar.
- data: Geçici verileri tutan sınıflar (Map, List, Set burada kullanılacak).
- util: Tekrarlanan yardımcı işlemleri yapan sınıflar.
- Main: Main sınıfı burada olacak, Scanner ile kullanıcı girişi burada yapılacaktır.

3. Oluşturacağımız Temel Sınıflar (10'dan fazla olacak)

Sınıf Adı	Paket	Açıklama
Book	model	Kitap bilgilerini tutar.
Author	model	Yazar bilgileri.
Category	model	Kitap kategorisi.
User	model	Kütüphane kullanıcısı.
Invoice	model	Ödünç alma iade işlemleri için fatura.
LibraryService	service	Kitap işlemleri.
UserService	service	Kullanıcı işlemleri/ gelen komutlara göre DataStore üzerinden kitap, kullanıcı, fatura gibi işlemleri yapar.
DataStore	data	Kitap, kullanıcı ve fatura bilgilerini geçici tutar.
InvoiceGenerator	util	Fatura oluşturur.
Main	main	Uygulamanın başlangıç noktası.

4. Kullanacağımız OOP Kavramları

- **Encapsulation:** Tüm model sınıflarındaki field'lar private olacak, getter/setter ile erişilecek.
- **Inheritance:** Person adında abstract bir sınıf olacak, User ve belki Author onu kalıtımla alacak.
- **Interface/Abstract Class:** Lendable diye bir interface tanımlayacağız, kitap bu interface'i implement edecek.
- **Composition:** Book içinde Category ve Author obje olarak tutulacak.
- **Polymorphism:** Ödünç alınabilir her şey için ortak bir davranış geliştirilecek.
- **Collections:**
 - List: Kitap listesi.
 - Set: Kullanıcının aldığı kitaplar (duplicate önleme).
 - Map: id'ye göre kitap/fatura eşlemesi.