

## NYP LAB 7

### Dijital İçerik Platformu

#### Senaryo

Bir dijital platform; **Video, Podcast, E-Kitap** içeriklerini tek bir koleksiyonda tutuyor. İçerikler **fiyatlanıyor**, bazıları **indiriliyor**, bazıları **reklam destekli** oynatılıyor ve hepsi **puanlanabiliyor**.

#### 1) Soyut sınıf: Content

##### Alanlar

- String id
- String title
- int durationMin (*video/podcast: dakika, e-kitap: tahmini okuma dk*)
- double basePrice

##### Ortak metotlar

- String getInfo() → "ID | Başlık | Süre"
- boolean isFree() → basePrice == 0
- abstract double finalPrice()
- abstract String type()

Content üzerinden nesne üretilmesin (abstract).

#### 2) Interface'ler

##### Rateable

- void addRating(int stars) (*1–5 değilse puanı **yok say**, ekleme yapma*)
- double averageRating()
- int ratingCount()

##### Downloadable

- double downloadSizeMB()
- boolean isDownloadable() (*kuralları içerik türü belirler*)

##### AdSupported

- int adCount()
- double adDiscount() (*toplam indirim miktarı*)

#### 3) Somut sınıflar (en az 3)

##### A) VideoContent extends Content implements Rateable, Downloadable

Kurallar:

- finalPrice = basePrice + durationMin \* 0.02
- downloadSizeMB = durationMin \* 15
- isDownloadable: **ücretsiz** içerikler indirilebilir, ücretli indirilemez (örnek kural)

Ek alan önerisi:

- int quality (480/720/1080) → 1080 ise boyutu %30 artır

##### B) PodcastContent extends Content implements Rateable, AdSupported

Kurallar:

- adDiscount = adCount \* 0.10
- finalPrice = max(0, basePrice - adDiscount) (*max için exception yok; negatif olmasın*)

Ek alan önerisi:

- boolean live → live ise reklam sayısı 0 kabul edilsin

##### C) EBookContent extends Content implements Rateable, Downloadable

Kurallar:

- finalPrice = basePrice + durationMin \* 0.005
- downloadSizeMB = 25
- isDownloadable: her zaman true

Ek alan önerisi:

- int pageCount → 300+ sayfa ise basePrice'a +1.5 ekle
- 

#### 4) Polimorfizm görevleri

List<Content> library içinde karışık içerikler oluştur ve şu çıktıları üret:

1. Tüm içerikler için:
  - type(), getInfo(), finalPrice()
2. **Interface tabanlı polimorfizm:**
  - List<Rateable> rateables oluştur (içeriklerden rateable olanları ekle)
  - Hepsine puan ekle, ortalamaları yazdır
3. **Davranışa göre filtreleme:**
  - List<Downloadable> downloads oluştur
  - isDownloadable() == true olanları ve downloadSizeMB() değerlerini yazdır
4. **Reklam özelliği olanları yönetme:**
  - List<AdSupported> ads oluştur
  - Toplam reklam sayısını ve toplam indirimi hesapla

ContentManager sınıfı yaz ve aşağıdaki metotları gerçekleştir:

- void add(Content c)
- void printAll()
- double totalRevenueIfAllBought() → tüm içeriklerin finalPrice() toplamı
- void printTopRated(int n) → Rateable olanları averageRating()'e göre sırala, ilk n'i yazdır
- void printDownloadReport() → indirilebilir olanları boyuta göre büyükten küçüğe yazdır

Aynı id eklenirse: **ekleme yapma**, konsola "ID zaten var, eklenmedi" yaz.

#### İndirim stratejisi (interface ile)

Bir interface ekle:

- PricingPolicy
  - double apply(Content c)

İki politika yaz:

- NoDiscountPolicy → fiyatı değiştirme
- WeekendDiscountPolicy → tüm içeriklere %10 indirim uygula

ContentManager içinde aktif politika seçilebilir olsun:

- setPricingPolicy(PricingPolicy p)
- finalPrice() yerine politika üzerinden "ödenecek fiyat" hesaplat