

RENTAL STORE- ADVANCED DESIGN PATTERNS

1. Design a Rental store that supports different types of items including movies, video games, music CDs, xbox, ps4, and books. Also, supports different categories of movies, music CDs, DVDs, and books
2. Support not only renting, but also selling those items
3. Basic functions to support renting and selling transactions for those items in the store

Design Classes Overview

The modified code is attached in the zip.

Below is the structure of the files and design decisions

- Runner.java- Main class with data set to run the program
- Item.java- The superclass from which all the below entities extend
 - Book.java
 - Categories: Thriller, Fiction
 - DVD.java
 - Categories: 256MB, 512MB, 1024MB
 - Movie.java
 - Categories: NewRelease, Childrens, Regular
 - PS4.java
 - Categories: Version1, Version2
 - VideoGame.java
 - Categories: Race, Kids
 - Xbox.java
 - Categories: Model2004, Model2006
 - Each of the above categories have a different selling, rental price and different frequent points
 - The above design is extensible to add any new kind of item in the future
- Transaction.java- The class responsible for maintaining the cart with items that are rented and sold
 - It has an array list of
 - Items chosen for Rental

- Items chosen for Sale
 - Transaction is a singleton class
- Rental.java and Sale.java- Individual classes holding the particular item in the cart, that is either rented or to be sold
 - Both extend from CartItem.java superclass as they share some similar properties
- Each of the subclasses in Item have a strategy class for calculating the frequent renter points, selling and rental prices
 - The strategy to be applied is chosen via a Factory based on the category of item rented/sold
- Once the appropriate strategy is applied, the functional flow moves on to applying coupons based on the frequent renter points earned by the customer
 - 3 decorators were created for handling the three kinds of coupons
 - Deduct10PriceDecorator.java
 - HalfPriceOffDecorator.java
 - Buy1Get1FreeDecorator.java
 - All the three extend from the Decorator.java class
 - The choice of which decorator to apply in sequence is done through the factory class based on the frequent points for the customer