OOP – Multimedia Shop

The goal of this lab is to practice **Object-oriented programming** by building a Multimedia Shop System for managing different items – movies, books and games. The items can be **sold** or **rented**.

A Multimedia Shop is a place where different multimedia items are sold – **books**, **movies** and **games**. Those items can be **sold** or **rented**. Your task is to model the Multimedia Shop using OOP.

Problem 1. Project Structure

First and foremost, we need to structure our project properly.

- Obviously, we are going to have **Items** – **Books**, **Movies** and **Games**. We also need somewhere to keep all the shop's **Rents** and **Sales**.
- So far, we have the classes where to keep the data. However, we also need an **Engine** to receive input, operate with our items, manage sales and rents, etc. **Managing sales** and **rents** is different from reading input, isn't it? It's best we define those two in **separate** classes – **RentManager** and **SaleManager**.
- Our engine will have to work with items, sales and rents, but do we *really* want to couple it with specific classes? What if we decide to add new types of items in the future? This is where interfaces come in – we will define the general behavior of an **item**, **rent** and **sale** in **separate interfaces** – and the specific class will implement that interface.

Create the described classes and interfaces. Think about logically grouping them into **separate namespaces** – the **models** (classes holding data), **interfaces** and **core logic** (shop engine, rent manager and sale manager).

Make sure you name everything properly, following the C# naming conventions.

Note: You may use a class diagram.