# 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**.

## Rents and Sales

We have our items, now we need to implement our **Rent** and **Sale** logic.

#### Step 1 – Sales

An item from the shop can be sold. The information about the sale should be stored in the **Sale** class (implementing the **ISale** interface).

* Holds an **Item** and **SaleDate.**

Constructors:

* Sale(item, saleDate)
* Sale(item) – sets as **SaleDate** the **current date**

#### Step 2 – Rents

Items from the shop may be **rented** for a certain **period of time**. Each rent has a **RentStatus** - either **Pending** (the deadline of the rent has not passed), **Overdue** (past deadline, the person has not returned the item) and **Returned** (the person has returned the item).

Create a **Rent** class, implementing the **IRent** interface.

* Holds an **Item**, **RentStatus,** and **RentFine**. The rent fine is calculated as follows: **1%** of the **item price** for **each day** after the deadline. Use the **current date** when checking the **RentStatus** and **RentFine**.

Constructors:

* Rent(item, rentDate, deadline) – creates a rent with **item**, rented on **rentDate** with a **deadline**
* Rent(item, rentDate) – sets as **Deadline** 30 days after **rentDate**
* Rent(item) – sets as **RentDate** the current date and as **Deadline** – 30 days after the current date

Methods:

* ReturnItem() – returns the item and sets the state to Returned.

#### Step 3 – Test Your Classes

|  |
| --- |
| DateTime today = DateTime.Now;  DateTime fiveYearsAgo = today.AddYears(-5);  Sale dieHardSale = new Sale(dieHardMovie, fiveYearsAgo);  Console.WriteLine(dieHardSale.SaleDate); // 1/30/2015 2:31:55 PM (today)  Sale acSale = new Sale(acGame);  Console.WriteLine(acSale.SaleDate); // 1/30/2010 2:31:55 PM  DateTime afterOneWeek = today.AddDays(30);  Rent bookRent = new Rent(sallingerBook, today, afterOneWeek);  Console.WriteLine(bookRent.RentState); // Pending  DateTime lastMonth = today.AddDays(-34);  DateTime lastWeek = today.AddDays(-8);  Rent movieRent = new Rent(godfatherMovie, lastMonth, lastWeek);  Console.WriteLine(movieRent.RentState); // Overdue  movieRent.ReturnItem();  Console.WriteLine(movieRent.RentState); // Returned  Console.WriteLine(movieRent.ReturnDate); // 1/30/2015 2:41:53 PM  Console.WriteLine(movieRent.RentFine); // 7.9200 |