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**.

Problem 6. Selling and Renting

We can now insert items into our shop. It's time we implemented selling and renting logic.

Selling books is done through the command **sell [id] [saleDate]**, which creates a **Sale** with the item with that **[id]** and date **[saleDate]**. Guess who's responsible for keeping sales? That's right – the **SaleManager** we created earlier.

Step 1 – Sale Manager

We're going to have to keep those sales somewhere – in a data structure – such as a **set**. All elements in a set are unique.

Create a **set** that stores all sales and a **AddSale()** method for adding **a sale** to the collection. Make sure that the **SaleManager** operates with the **notion of a sale**, not a concrete type (through the interface **ISale**).

Step 2 – Adding Sales

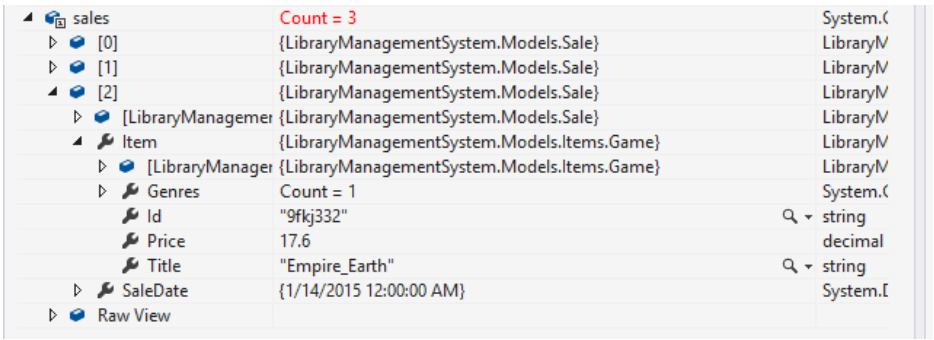
On command, the **ShopEngine** should call the **SaleManager**'s **AddSale()** method and decrement the available supplies by 1. If the item has no supplies, an exception should be thrown.

- Create a custom exception for that purpose – **InsufficientSuppliesException**.

Add 3 sales:

```
supply book 5 id=4fd332&title=Razkazi&price=7.99&author=Elin_Pelin&genre=story
supply game 19 id=9fkj332&title=Empire_Earth&price=17.60&ageRestriction=Minor&genre=strategy
sell 4fd332 24-12-2014
sell 9fkj332 13-01-2015
sell 9fkj332 14-01-2015
```

The result in the debugger should be:



Step 3 – Rents Manager

Do the same for the rent command, but this time in the **RentManager**. Renting an item is done through the command **rent [id] [rentDate] [deadline]**. The **RentManager** should store all rents and add new ones through an **AddRent()** method. Make sure you work through the **IRent** interface.

Step 4 – Adding Rents

The **ShopEngine** should decrement the respective item supplies whenever an item is rented. In case no supplies are available, throw **InsufficientSuppliesException**.

Add 3 rents:

```
supply book 5 id=4fd332&title=Razkazi&price=7.99&author=Elin_Pelin&genre=story
supply game 19 id=9fkj332&title=Empire_Earth&price=17.60&ageRestriction=Minor&genre=strategy
rent 4fd332 24-12-2014 24-01-2015
rent 9fkj332 13-01-2015 23-02-2015
rent 9fkj332 14-01-2015 14-05-2015
```

The result in the debugger should be: