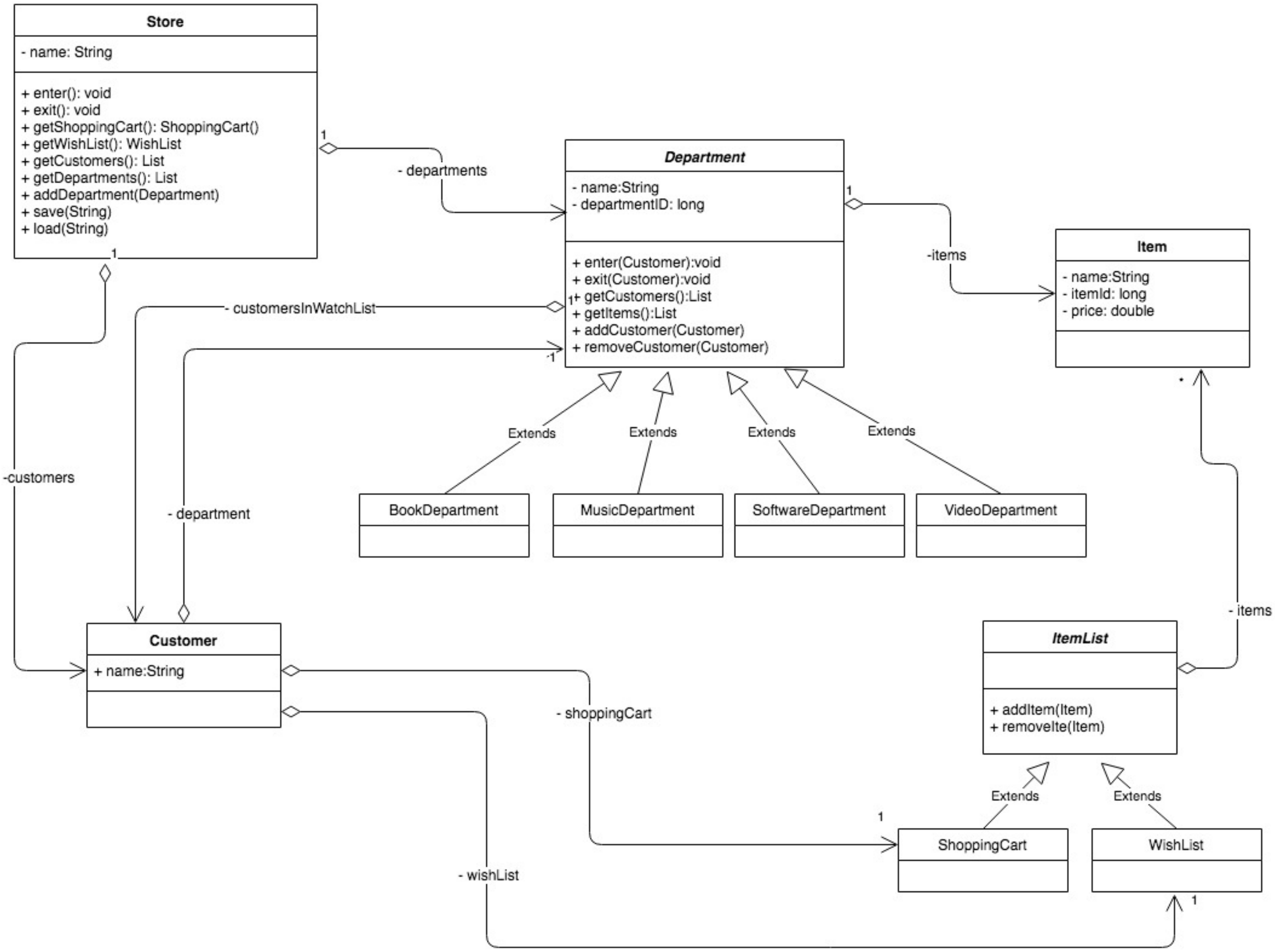
Homework 5 – System Design Patterns

In this assignment, you will apply design patterns to the design and implementation of an on-line shopping site, using the framework diagrammed below.



The classes are described below:

Store – the on-line shopping site itself

Attributes / relationships:

- name – the site's name (e.g., "Amazon.com")
- departments – the various sales departments
- customers – the clients currently using the store

Methods:

- enter(Customer c) - Customer c enters the store
- exit(Customer c) – Customer c exits the store
- getShoppingCart() – returns an empty ShoppingCart

- getWishList() – returns an empty WishList
- getCustomers() – returns a list of the customers in the store
- getDepartments() – returns a list of the store's departments
- addDepartment() – adds a new department to the store
- save(String filename) – persist all the system to a file
- load(String filename) – loads all the system from a file

Department – abstract superclass for a department

Attributes / relationships:

- name – the department name (e.g., "Computer Software")
- items – items available for sale in the department
- customers – the clients currently using the department
- observers – the clients to be notified of sales or new items
- departmentID – a unique ID for the department

Methods:

- enter(Customer c) - Customer c enters the department
- exit(Customer c) – Customer c exits the department
- getCustomers() – returns a list of the customers in the department
- getItems() – returns a list of the items for sale in the department
- addCustomer() – adds a customer to the department sales watchlist
- removeCustomer() – removes a customer from the department sales watchlist

BookDepartment, MusicDepartment, SoftwareDepartment, VideoDepartment – possible subclasses of Department

Item – an item for sale in a department

Attributes / relationships:

- name – the item name (e.g., "Adobe Photoshop")
- itemId – a unique ID for the item
- departmentID – the ID of the department selling the item
- price – the price of the item

Customer – a client of the on-line shopping site

Attributes / relationships:

- name – the customer name (e.g., "John Doe")
- shoppingCart – the cart being used by the customer
- wishList – the wish list generated by the customer
- department – the department the customer is currently watching

ItemList – abstract superclass for a list of items

Attributes / relationships:

- items – items currently in the item list

Methods:

- addItem(item) – add an item to the list
- removeItem(item) – remove an item from the list

ShoppingCart, WishList – concrete implementations of *ItemList*

Using the framework described above, design and implement using Java, a working version of the on-line shopping site (a desktop application). Your assignment must have at least three customers, at least four departments, and at least six items for sale in each department. *You should provide a file to load the scenario described before in your submission.* The user interface should be a simple console application.

Your program must implement the following design patterns:

- **Iterator**
  - Your *ItemList* implementations should provide an iterator.
  - For simplicity, you can use an array of up to 100 elements for your list. You cannot use any Collection implementation, however.
- **Singleton**
  - Only one instance of any department type should be allowed to be created.
- **Observer**
  - Customers can subscribe to the sales watch list for each department.
  - In other words, a customer should be able to request notification whenever a new item is added for sale in a department or a specified old item's price is reduced. The customer should print on screen a "Wow!" message.
  - Demonstrate this pattern by having at least two customers register for such notifications with at least one department (see below).

Provide a Main class to demonstrate the functionality:

Load the Data File and Print a message. Use that data to demonstrate the following.

1. Watchlist
   a. Add two customers to a department watchlist
   b. Change the price on an item on that department.
   c. Remove one customer from that department watchlist.
   d. Change the price again.
2. Shopping Cart
   a. Add three items to a customer shopping cart
   b. Print the items on a customer shopping cart (using for each loop and the customized iterator)
3. Department Management
   a. Add a software department to the store

b. Add it again
c. Print the list of departments of the store

A short report (no more than five pages) should be submitted with your fully implemented program. This report should describe your use of design patterns in the assignment, including:

- Class diagrams (UML notation)
- Identification of other viable patterns for this program (or reasonable extensions of it)
- A screenshot of the output of running your Main Class program
- A paragraph describing which pattern can you use to implement a feature that offer customers the opportunity to buy combinations of items within a department. For example, buy all Harry Potter movies on Blu-ray, and all the "Harry Clifton Chronicles" books, or all three volumes of Donald Knuth's "The Art of Computer Programming". *Include a UML class diagram to show How will you apply the pattern to implement that feature* (do not include the entire system diagram)

Important Note: the classes presented in this assignment describe a framework for you to work with. As you start implementing the assignment, and in particular to apply patterns, your classes will eventually change. Those changes should be among the descriptions you should include in the report.

What to submit…

You should submit on Canvas:

- **Source code** (*.java* files only, do not submit a zip file with your project). *[60 points]*
    - *Include your name as a comment in the first line of each java file.*

- **The report** (a single *.pdf* file that includes all the items described before). *Note that the presentation of that report (i.e., how readable and descriptive it is) will be considering when grading*. *[40 points]*

Submit a *printed copy* of your report at the beginning of the class on Tuesday November 13th.

(failing to submit this, your report will not be graded)

You have 3 weeks for this assignment. However, **start working on it the sooner the better**.