Lessons Learned from Project 1

Hailun Zhu [hailunz]

Project1 Unit1

1. What is the relationship between containment and encapsulation (as applied in this project), when building components?

   In this project, in the Automobile class we have used the instances of OptionSet class and in OptionSet class we used Option class, this is containment.

   And each of these class implements encapsulation. To utilize encapsulation, I declare the variables of a class as private and I provide public setter and getter methods to modify and view the variables values.

2. What are some ways to analyze data (presented in requirements) to design Objects?

   I firstly transfer the requirements to a class diagram. Then based on the class diagram, I can design the corresponding Objects.

   In the class diagram, I defined the variables and their types, as well as the functions that an Object should have, and also their relationships. It is more clearly shown in the class diagram that what are expected to implements. Thus this is a good way to analyze data.

3. What strategies can be used to design core classes, for future requirements, so that they are reusable, extensible and easily modifiable?

   One way is to create an abstract class and several interfaces. We can design a basic abstract class that will implement all the methods that a class would need to use. And the class will extend the abstract class and the interfaces. Each time we want to reuse the class, we initiate a new instance. And we can create new interface to just focus on certain functionalities. And we could only modify the methods in the abstract class.

4. What are good conventions for making a Java class readable?

To make a java class readable, we can standardize the variable and methods' names. And we can order the variables and the method by their names and types.

We could also add necessary comments into the code to make it more readable.

And also keep a maximum value for the line length and wrap lines when necessary.

5. What are the advantages and disadvantages of reading data from sources such as text files or databases in a single pass and not use intermediary buffering?

   The advantages for reading data from sources are that there is no requirement for the buffer size and the file size. If the file size is large or the buffer size is small, we could not read the file by saving it to the buffer. The disadvantage is that after you read a line you could not go back. You can only read the file line by line or byte by byte, and so on.

   The advantage for storing in a buffer is that we can retrieve whatever data we have read. The disadvantage is that there will be some problems if the buffer size is not enough to hold the file content.

6. What is the advantage of using Serialization? What issues can occur, when using Serialization with Inner classes?

   The advantage of using Serialization is we could store the object and retrieve the object directly to and from a file stored on the disk. And we could send the object and receive the object over the network.

   If we have inner classes, every inner class should also be serializable.

7. Where can following object relationships be used: encapsulation, association, containment, inheritance and polymorphism?

   For each basic class, we could use encapsulation to protect the variables inside the class.

   We could use association when some classes will work together with other classes. Like FileIO will use Automobile classes.

For containment, when we have inner class or we need to initialize another class, we will use containment.

For inheritance, we could use an abstract class like the proxyAutomobile, and extends the abstract class to inherit the variables and methods.

For polymorphism, we implement several interfaces and each interface only utilize a subset of methods of the proxyAutomobile. And different class implements the same interface could behave different based on their implementation.

8. How can you design objects, which are self-contained and independent?

   An object is self-contained and independent if it has its data and also the data manipulation methods in it. It could process the data on its own without other classes' help. And it can do something independently.

Makeup point in Unit1:

(-0.5) : Dead code in FileIO.java

This is deducted because I have some commented out code in the FileIO.java.

I have already corrected this by removing those unused code. Thus this could be given back.

To sum up, I have learned the concepts and implementations of the following:

1. Object Theory

2. Inner Classes

3. File IO, buffered Reader

4. Serialization

5. Class diagram

Project1 Unit2

1. What role(s) does an interface play in building an API?

   Each interface has its own functionalities. It could only have a subset of methods of the proxyAutomobile for other to implement. It allows objects having different internal implementations to share the same external interface.

   For polymorphism, we implement several interfaces and each interface only utilizes a subset of methods of the proxyAutomobile. And different class implements the same interface could behave different based on their implementation.

2. What is the best way to create a framework, for exposing a complex product, in a simple way and at the same time making your implementation extensible.

   We could use an abstract class like the proxyAutomobile, and extends the abstract class to inherit the variables and methods. And the real implementations are in the content of the abstract class and we only expose the BuildAuto class and we use the corresponding interface to process the data instead of directly use the complex class.

3. What is the advantage of exposing methods using different interfaces?

   By using a certain interface, you can only do some of the jobs, it is more simple to use and you would not mess up using the methods for which you should not use.

4. Is there any advantage of creating an abstract class, which contains interface method implementations only?

   In this way, we could make the modification easier, we could only modify the methods inside the abstract class and all the interfaces that will be using this will change at the same time. We do not need change every method in every class.

5. How can you create a software architecture, which addresses the needs of exception handling and recovery?

   We could create an interface named FixAuto, and we create a class which implements this interface and call certain fix method based on the error type. The AutoException is an example. It is an exception management class. And it will call the methods in another class that implements the recovery method to deal with the exceptions.

6. What is the advantage of exposing fix methods for exception management?

   The code is clean. The implementations are in another file and the exception management will only manage the exceptions and call the fix methods to deal with the exception.

7. Why did we have to make the Automobile object static in ProxyAutomotive class?

   Because every instance should have access to the object. Making it static could do so.

8. What is the advantage of adding choice variable in OptionSet class? What measures had to be implemented to expose the choice property in Auto class?

   Adding choice variable could keep track of the user's choice of the OptionSet and could help us to calculate the final price for this model. Set, get and update methods should be implement for the Option choice.

9. When implementing LinkedHashMap for Auto object in proxyAuto class, what was your consideration for managing CRUD operations on this structure? Did you end up doing the implementation of CRUD operation in proxyAuto or did you consider adding another class in Model for encapuslating Auto for the collection and then introducing an instance of this new class in proxyAuto. (Think about this and if this part of your design is not selfcontained, then fix it.)

   Firstly making this static. Secondly adding synchronized to every method of CRUD operations. I decide to do this in the proxyAuto.

To sum up, I have learned the concepts and implementations of the following:

1. Object theory

2. Exception Handling, customer exception handler

3. Abstract classes

4. Interfaces

Project1 Unit3

1. What is the best way to setup multithreading in an Enterprise Class application?

   By creating a class and extends Thread. And after new an instance for this class, we call the start method.

2. What strategy is used for synchronizing, so you end up with a scalable application?

   The basic strategy is to add synchronized key word to every CRUD operation method in the proxyAutomobile. And because the methods in the proxyAutomobile are synchronized, it could work on its own. Thus it is still self-contained. In this way, it could synchronize read and write separately.

   In order to synchronize the read and write operation at the same time, I also add synchronized methods in EditOptions. So that when reading, it could not do writing at the same time.

3. What implementation strategy can be used for creating a race condition for testing multithreading.

   We could use the random wait method to let some method to sleep for sometime, thus the order of the operation might change the outcome and then generate a race condition.

4. How does Synchronization work in JVM? What are the performance consequences of Synchronizing?

   We could add synchronized key word to a method, so each time only one thread could call this method. We could also synchronize the method. Thus each time only one thread could access to read or modify the object.

   And because using synchronization, the thread could be blocked, thus make the program run slower (for waiting for others to release the resource).

To sum up, I have learned the concepts and implementations of the following:

1. Multithread in Java

2. Make the application scalable

Project1 Unit 4

1. In this unit, I learned how to transfer and restore the object through the network.

   Because the Automobile is serializable, it can be sent over the network. And if the server side implementation of the object is the same, it could be successfully reconstruct when we receive the object from the object input stream. Otherwise we could specify a certain serialVersionUID to the object to identify they are the same object.

2. Learned how to use ServerSocket and socket to set up the communication in the network.

   For server, I created a DefaultSocketClientServer, which will maintain a ServerSocket of a given port. It extends the Thread so it is another thread that will always listen to the incoming connection request, by using socket.accept() method. Once it gets a new connection, it will create a new instance of DefaultSocketClientHandler and pass the socket to the handler to deal with the communication issues. The DefaultSocketClientHandler also extends Thread, so it will work in another thread thus when dealing with the request, it will not block the server to open new connections. Each handler deal with one connection, once the job is done, it will close on its own.

   The handler will deal with four types of request. One is to accept a stream from a file and build auto model from the file and store it in the LinkedHashMap. One is to receive a property object and construct the model based on the object. One is to return the available model list to the client. The last one is to send the selected model object over the network to the client.

   For the client side, I create a new package called client. I created a DefaultSocketClient class, which creates a new thread and will maintain a socket to communicate with the server.

   It could send out 4 types of request. One is to send a byte array stream to the server, one is to send the property object, one is to request to get available model list, the other is to get a certain model. And there is a class called SelectCarOption, which implements UpdateAuto and CreateAuto interface to update options in the client side and display the model.

Makeup points:

(-1) : Please elaborate your decision decisions properly in the document

In this document I have illustrate my decisions. Thus I think this point could be given back.

To sum up, I have learned the concepts and implementations of the following:

1. Multithreaded server and client

2. Java Server socket and socket.

Project1 Unit5

1. I learned how to use Servlet.

   In this project I do not change the server side code.

   I have created a new package servlet in the client.

   The getAvailableModels class extends HttpServlet and in the doGet method, I created a DefaultSocketClient to send out the request and wait to get the available model list and put the information into a JSP to show the result.

   The getModel class also extends HttpServlet. In the doGet method, it gets the name of the selected model and use a DefaultSocketClient to request a model object from server and then output the information into a JSP. In the doPost method, it gets the user's option of the current model and display the final result and the price for the choices in a JSP.

2. I have learned how to use a JSP.

   I have three jsp files in my project. The getModels.jsp will show the result of the available model list for user to choose. The showModel.jsp will show the options for a selected model. And the showChoice.jsp will show the final result and the price for the choices.

3. I have learned how to create form and get form data from the html, get/ post requests.

To sum up, I have learned the concepts and implementations of the following:

1. Web programming
2. HTTP protocols
3. Building forms
4. Using Servlet
5. Using JSP
6. MVC model

Project1 Unit6

1. The table schema that I designed.

```
CREATE TABLE Automobile(

      id INTEGER not NULL AUTO_INCREMENT,

      name VARCHAR(255) not null Unique,

      make VARCHAR(255),

      model VARCHAR(255),

      base_price FLOAT not NULL,

      PRIMARY KEY (id)
```

```
    );

    CREATE TABLE OptionSet(
        id INTEGER not NULL AUTO_INCREMENT,
        name VARCHAR(255) not NULL,
        auto_id INTEGER not NULL,
        PRIMARY KEY (id),
        FOREIGN KEY (auto_id)
        REFERENCES Automobile(id)
        ON DELETE CASCADE
    );
    CREATE TABLE Options(
        id INTEGER not NULL AUTO_INCREMENT,
        name VARCHAR(255) not NULL,
        price FLOAT,
        optset_id INTEGER,
        PRIMARY KEY (id),
        FOREIGN KEY(optset_id)
        REFERENCES OptionSet(id)
        ON DELETE CASCADE
);
```

2. In this unit I learned how to interact with the JDBC and MySQL database.

   I have created a database package on the server side.

   I created a JDBCAdapter abstract class that help set up the connection with the database.

   I created 4 classes that extends the JDBC Adapter to do different jobs. The CreateTable class will help to create certain table if not exited. The AddModel will help to load the information into three tables correspondingly. The DeleteModel will delete entry in the Automobile table and it will cascade delete the corresponding entries in the OptionSet and Option table. The UpdateTable will use the information in Automobile table to find the certain entry in OptionSet table or Option table and update the entry that meet the requirement.

3. I learned how to set up MySQL on MAC and manipulate data in the database by using select, create, delete operations.

4. In order to add, update, delete, I have implemented or modified methods in the proxyAutomobile class to create a class to connect to the database and do the operations.

To sum up, I have learned the concepts and implementations of the following:

1. Working with Relational Databases.

2. Schema design for relational databases.

3. SQL functions

4. Using JDBC

5. Setup and usage of MySQL