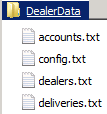
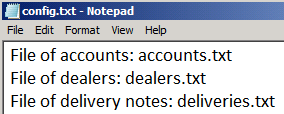
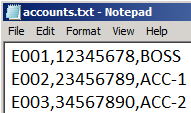
|  |  |  |
| --- | --- | --- |
| **LAB 211 Assignment** | **Type:** | **Long Assignment** |
| **Code:** | **16** |
| **LOC:** | **500** |
| **Slot(s):** | **N/A** |

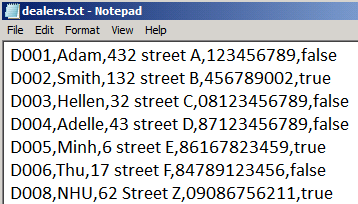
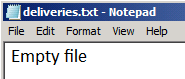
**Title**

Dealers Management Program

**Background**

* AZW, a firm, needs a Java console program for managing it’s product dealers. This program must support a basic security. Roles in the firm include:
* (1) Boss: Managing users
* (2) ACC-1: Managing dealers
* (3) ACC-2: Managing deliveries notes.
* Data files are supported as following:

File related to the program can be setup flexibly through the file ***config.txt***.

All users must be login to system to carry out appropriate management activities based on his/her role.

At the moment, a program for managing dealers is required to develop.

Program model is proposed:

User 🡪 Login 🡪 Managing dealers ( role: ACC-1)

🡪 Managing deliveries (role ACC-2) – developed afterward.

**Program Specifications**

Build a management program. With the following basic functions

1. Show medical examination result
2. Add new patients
3. Record medical examination
4. Real-time update processing

Others- Quit

Each menu choice should invoke an appropriate function to perform the selected menu item. Your program must display the menu after each task and wait for the user to select another option until the user chooses to quit the program.

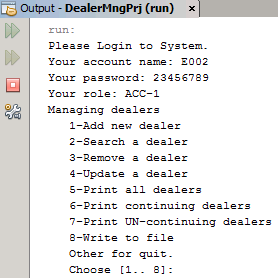
Each department has the following information: departmentID, name, createDate, lastUpdateDate

Each doctor has the properties such that doctorID, name, sex, address, departmentID, createDate, lastUpdateDate

Each patient has the following information: patientID, name, age, address. Patient information is stored in patient.dat file

examination.dat file has stored examination information include: examinationID, doctorID, patientID, result, date.

**Features:**



***A Security in Code view***

* + An object belonging to the LogIn class should be a parameter to create a DealerList object. So, the class DealerList can only be used depending on the LogIn object.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Component/ class | Sum | LOCs | Question Mng |  |
| Structure | 20 | 20 |  |  |
| Account | 10 | 10 |  |  |
| MyTool | 130 | 100 | 30 |  |
| Config | 20 | 20 |  |  |
| AccountChecker | 30 | 30 |  |  |
| Menu | 10 | 10 |  |  |
| DealerList | 200 | 150 | 50 |  |
| LogIn | 80 | 50 | 30 |  |
| Total | 500 |  |  |  |

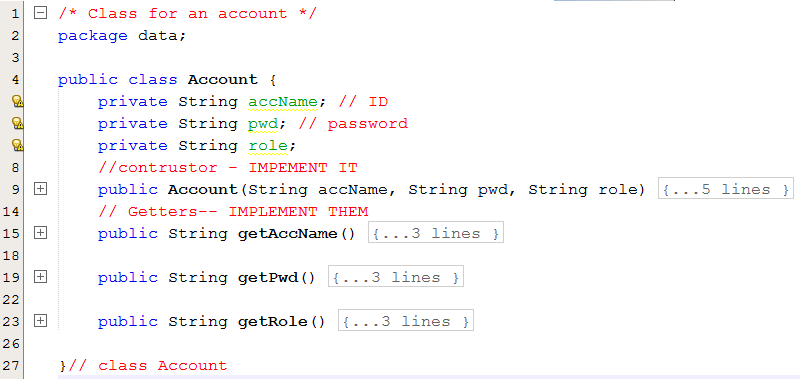
**Design Hint**

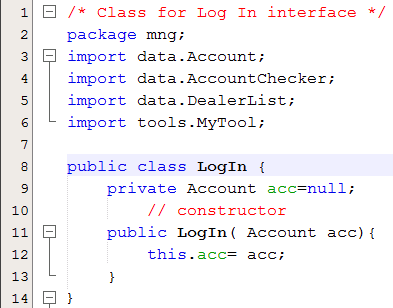
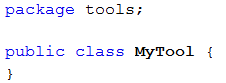
|  |  |
| --- | --- |
|  | Class for an account  Class supporting checking validity of an account  Class for reading config file to get related filenames  Class for a dealer  Class for a list of dealers  Class for Log in interface. The main() method is put in this class  Class for menu  Class for programming common used basic methods |

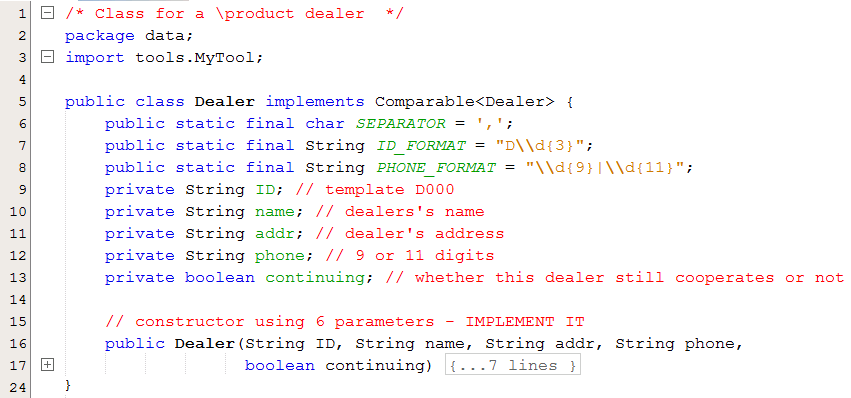
**Implementation – Step 1**

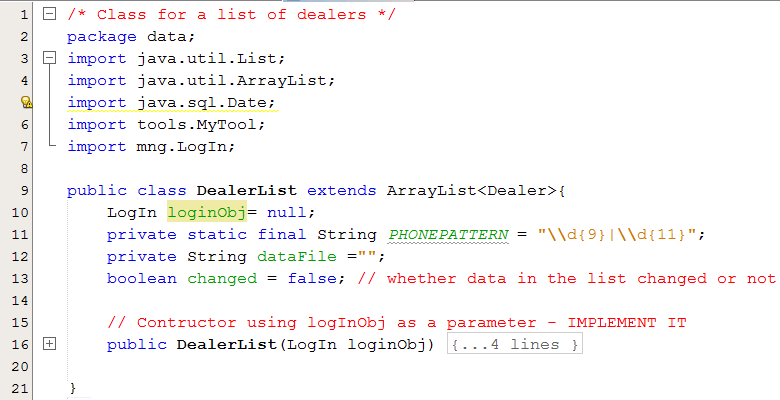
Because there is a relation between classes in different packages (the mng.LogIn class must be used in the data.DealerList class). At this step, we should write basic following code to prevent the Java compiler causing compiled errors. You should implement them in order.

1. ***Class Account***



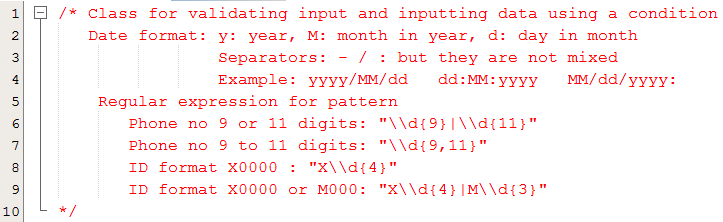
 

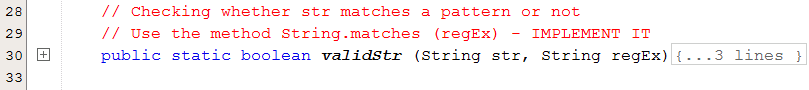
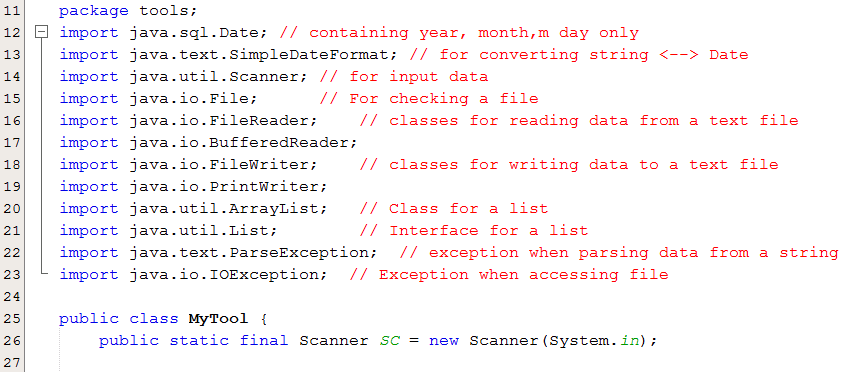


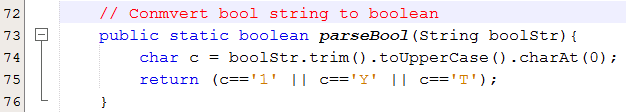
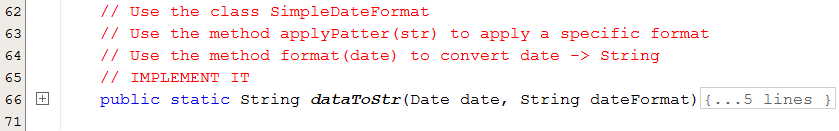
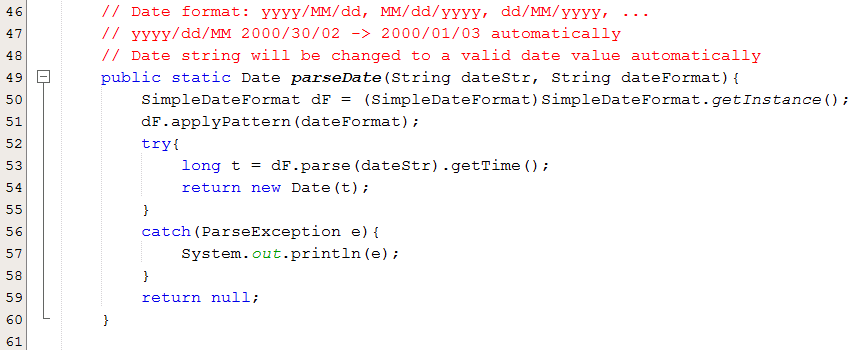
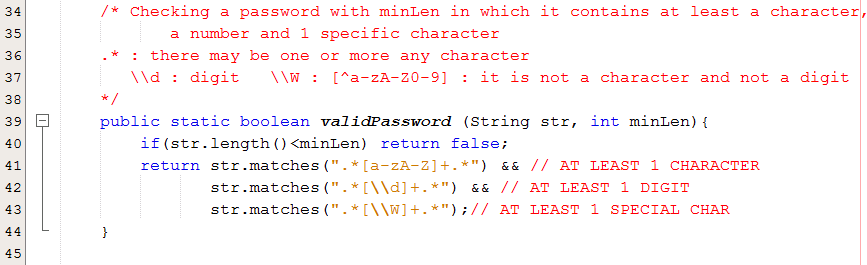


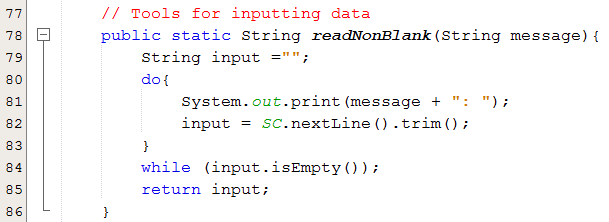
**Implementation – Step 2, Implementations in Details**

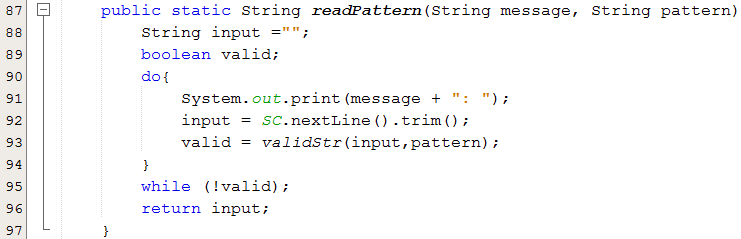
1. ***Class MyTool***

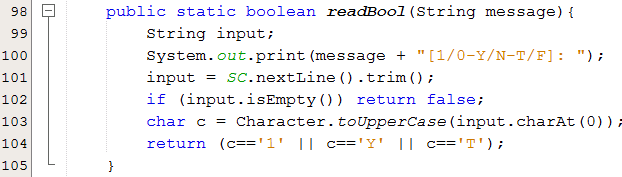


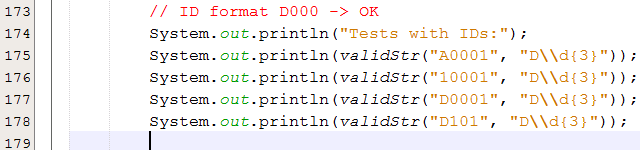
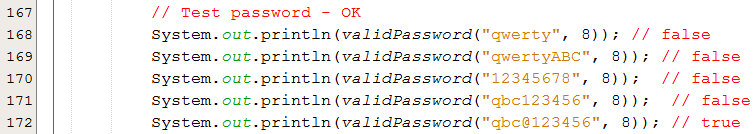
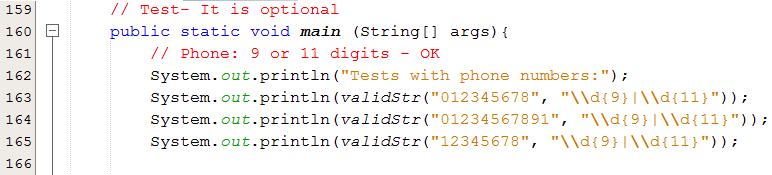
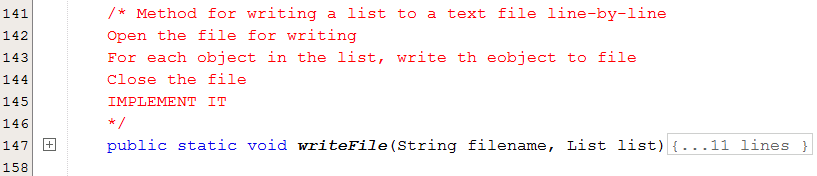
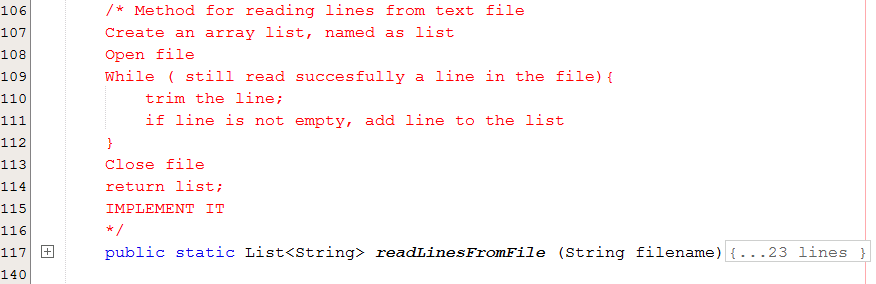


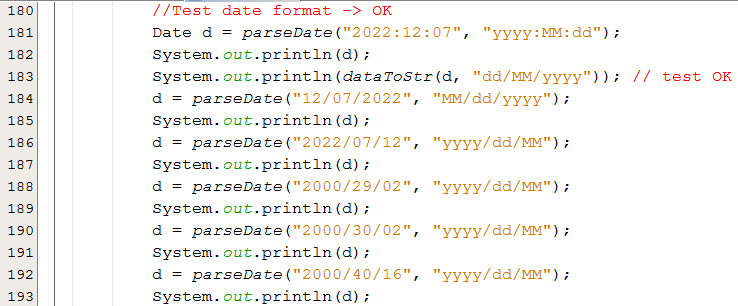


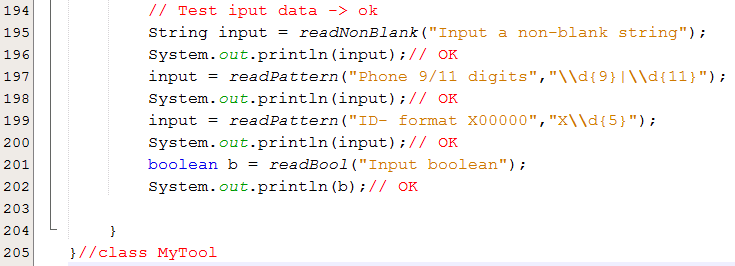


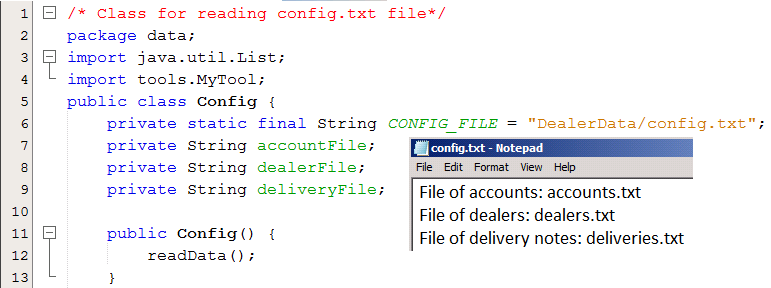


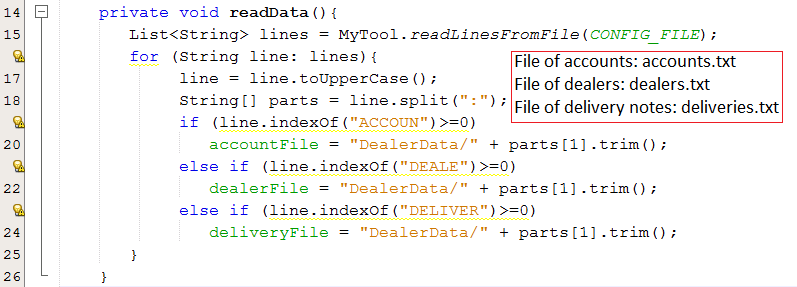


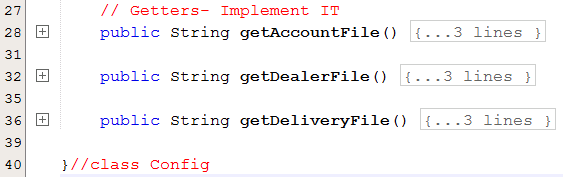




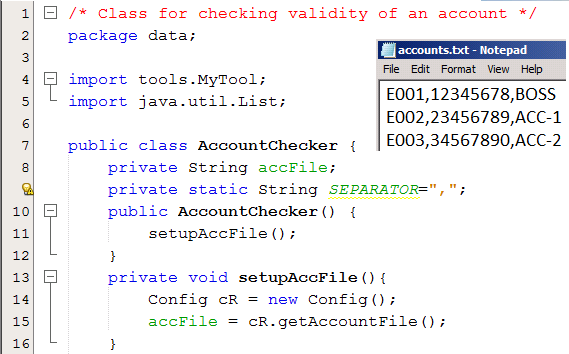
  
***3- Class Config***

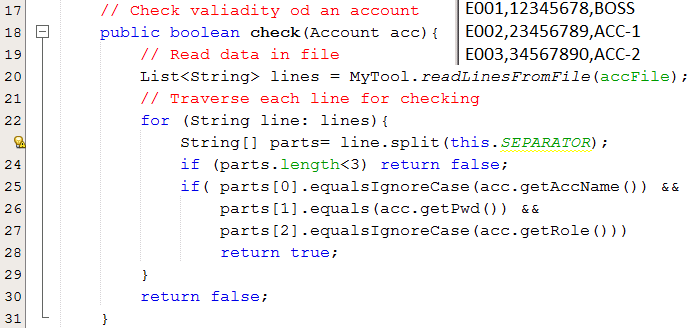


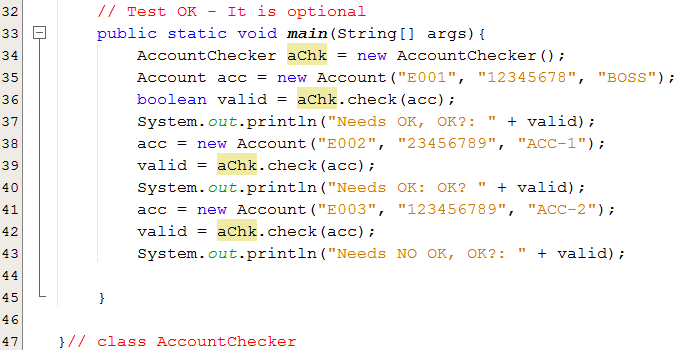




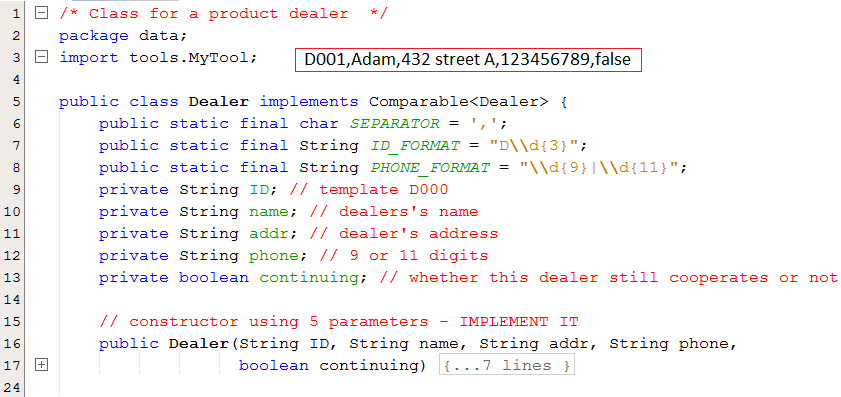
***4- Class AccountChecker***

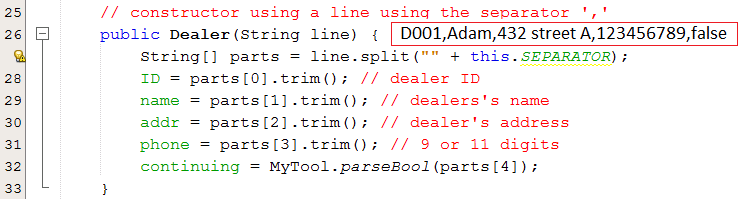


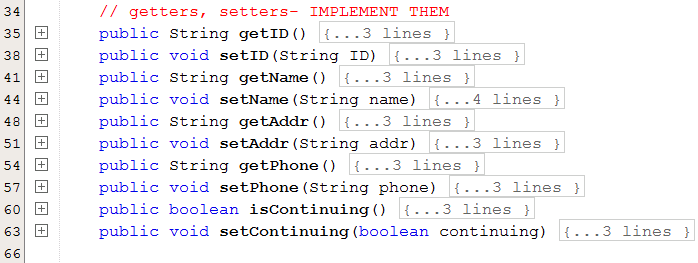


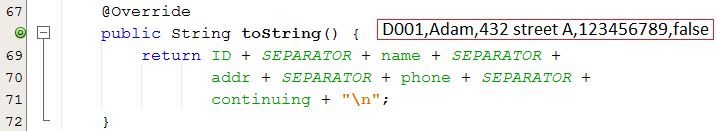


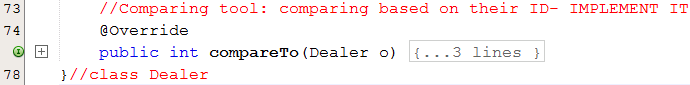
***5- Class Dealer***



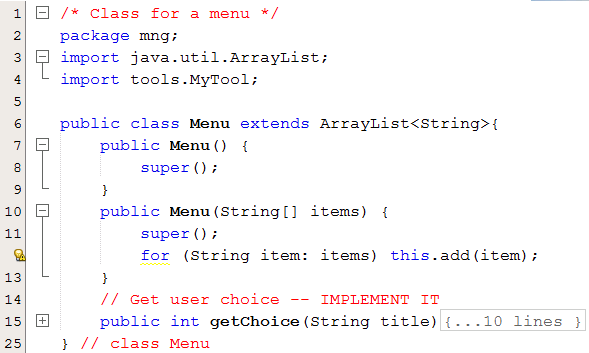




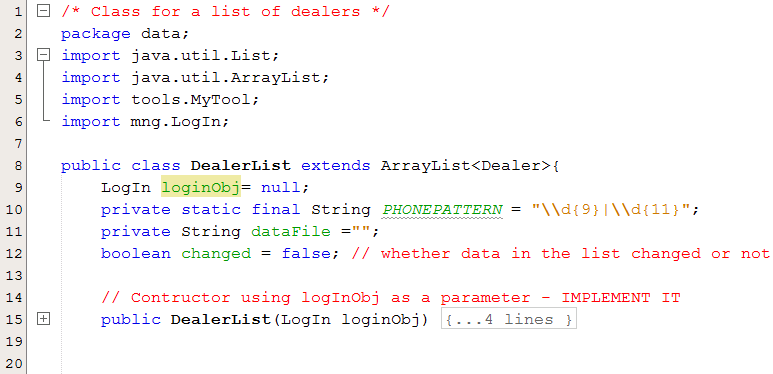


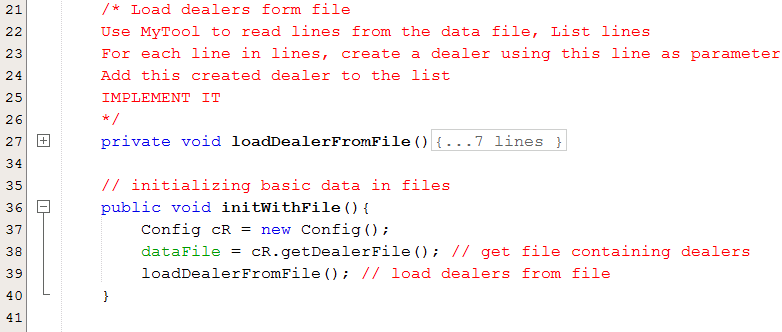


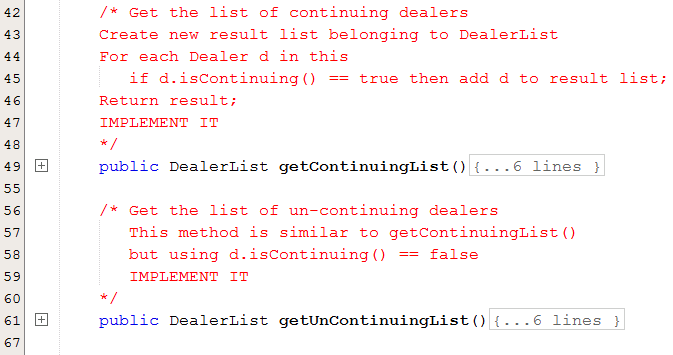
***6- Class Menu***

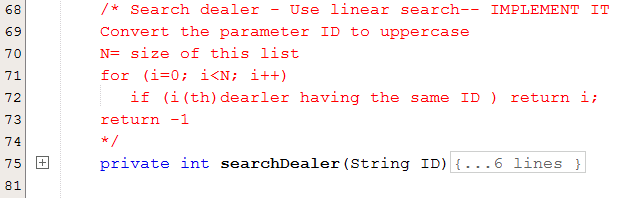


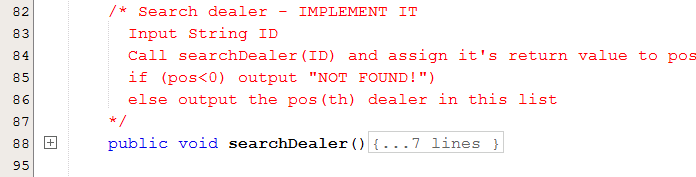
***7- Class DealerList***

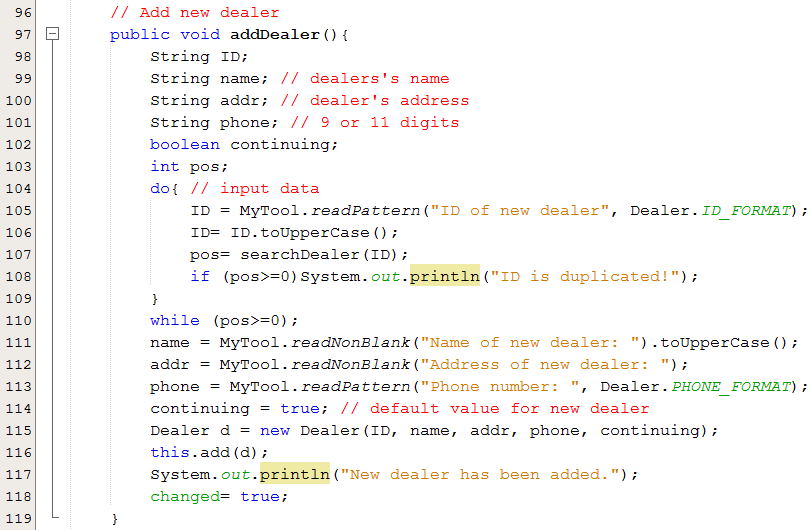


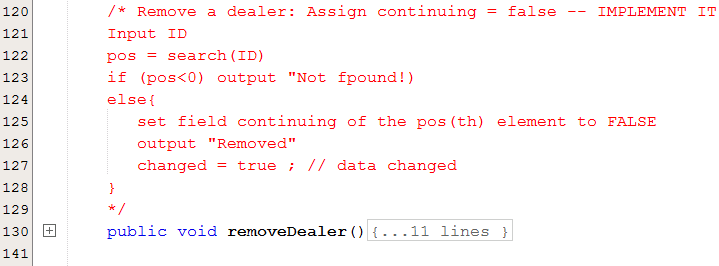


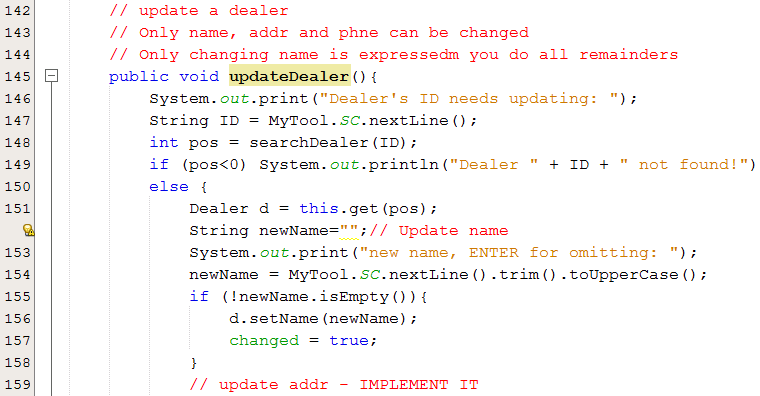


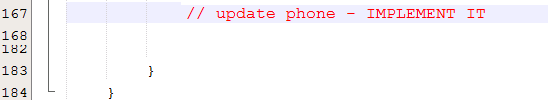


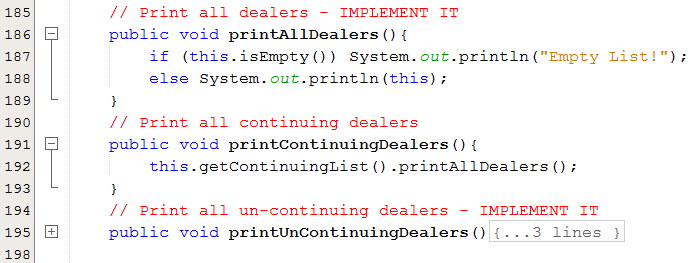


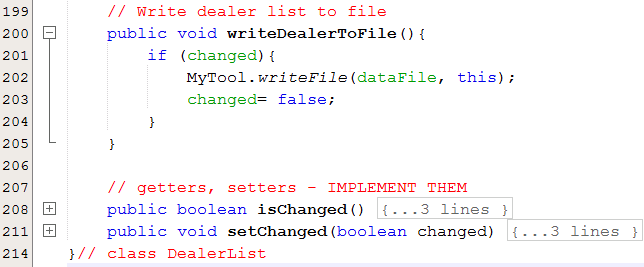




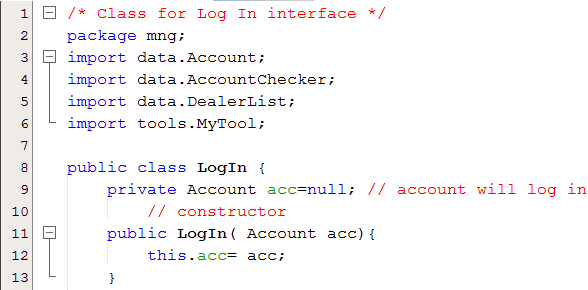


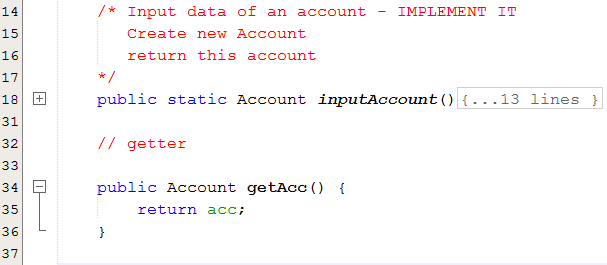


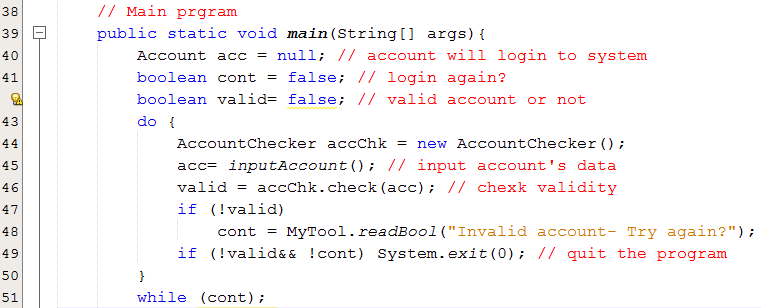


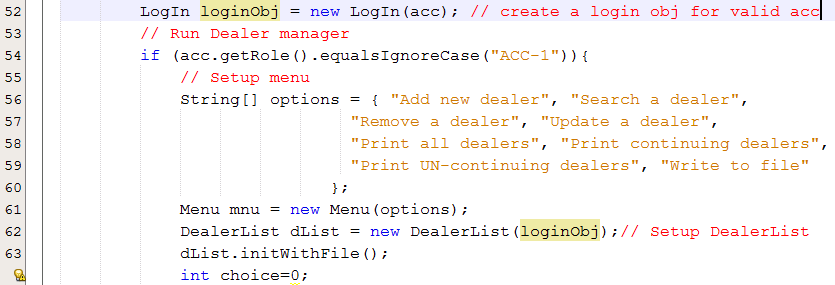


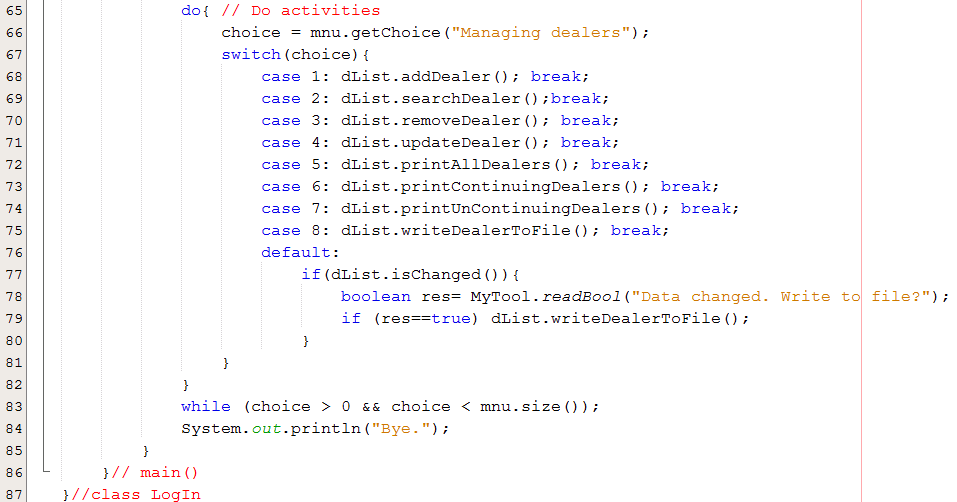
***8- Class LogIn***











* The above specifications are only basic information; you must perform a requirements analysis step and build the application according to real requirements.
* The lecturer will explain the requirement only once on the first slot of the assignment.