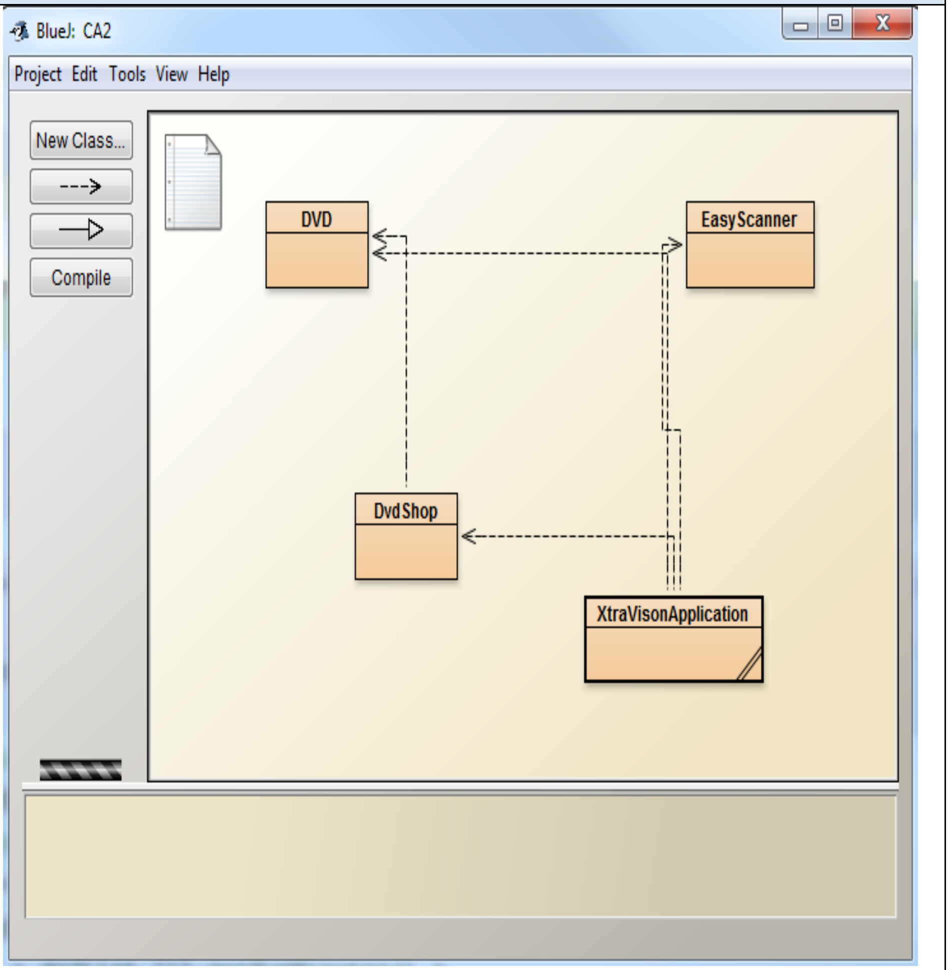
XtraVision Shop Application

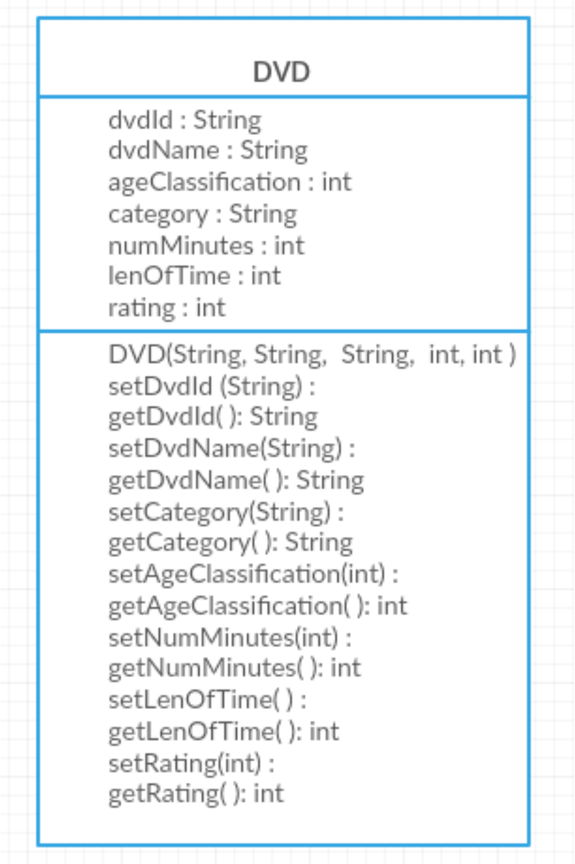
You are required to develop a DVD menu system

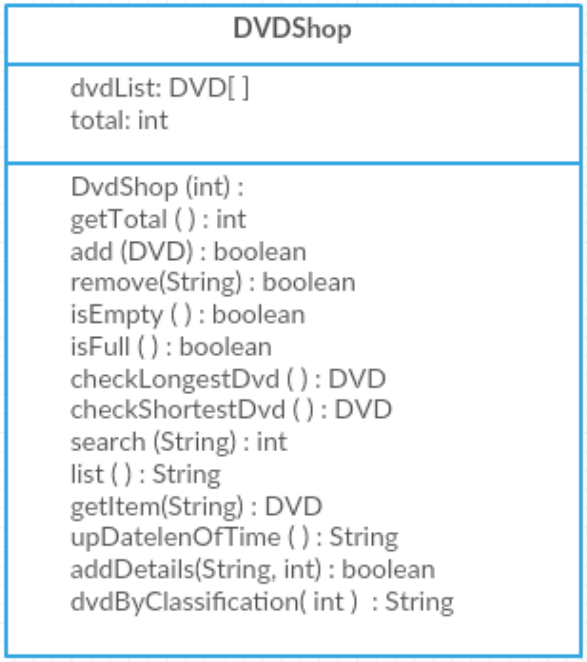
This will involve the development of three new classes:

- DVD
- DvdShop
- XtraVisionApplication

You will use the EasyScanner for user input.



 DVD dvdId : String dvdName : String ageClassification : int category : String numMinutes : int lenOfTime : int rating : int DVD(String, String, String, int, int) setDvdId (String) : getDvdId( ): String setDvdName(String) : getDvdName( ): String setCategory(String) : getCategory( ): String setAgeClassification(int) : getAgeClassification( ): int setNumMinutes(int) : getNumMinutes( ): int setLenOfTime( ) : getLenOfTime( ): int setRating(int) : getRating( ): int	The **DVD** class. This is a template / blueprint class for a DVD. It stores data on the id, name, age classification, category, rating, running time and length of time in shop (assume it starts at 0 until changed). The constructor updates **some** of these attributes (instance variables) with the information passed as a parameter. See runtime version. You need an accessor and mutator methods for each attribute.



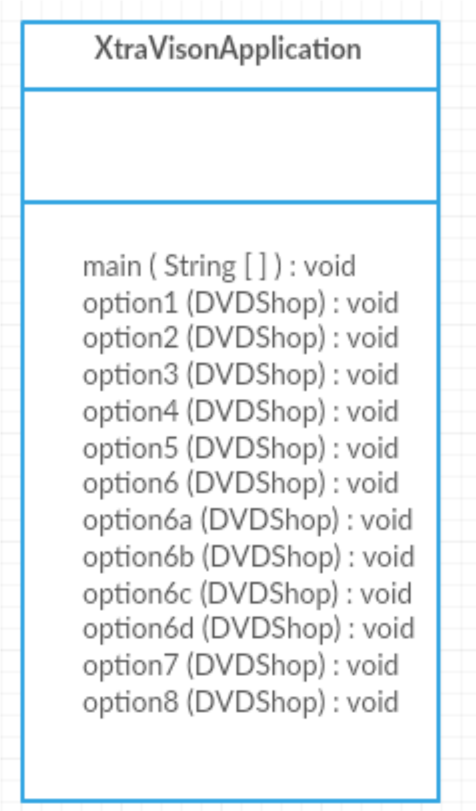
```
DVDShop
-----------------------------
dvdList: DVD[ ]
total: int
-----------------------------
DvdShop (int) :
getTotal ( ) : int
add (DVD) : boolean
remove(String) : boolean
isEmpty ( ) : boolean
isFull ( ) : boolean
checkLongestDvd ( ) : DVD
checkShortestDvd ( ) : DVD
search (String) : int
list ( ) : String
getItem(String) : DVD
upDatelenOfTime ( ) : String
addDetails(String, int) : boolean
dvdByClassification( int )  : String
```

The **DVDShop** class.

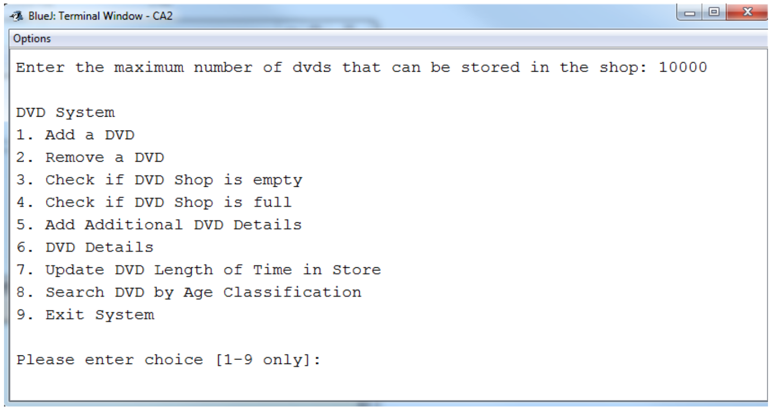
This class creates an array of dvd and handles it. You will need to maintain a total attribute that will hold the number of dvds stored in the system.

Typical methods would include (**but not limited to see UML diagram for full list**):

- a constructor that will set the size of the array and set the total number of DVDs to 0.
- add a DVD to the system (to the array)
- remove a DVD from the system (from the array)
- check if the system is empty
- check if the system is full
- return the total number of DVDs in the system
- search the system for a DVD
- show all of the DVDs stored in the system
- a method to update **ALL** the DVDs length of years in the store (this will update all the dvd length of time in store by 1).



The XtraVisionApplication class contains the main method. This class displays the menu repeatedly to the user. Note that there is a private method behind each option on the menu. i.e. if the user enters 1 it will call another method to execute option 1, if the user enters 2 it will call another method to execute option 2 etc.



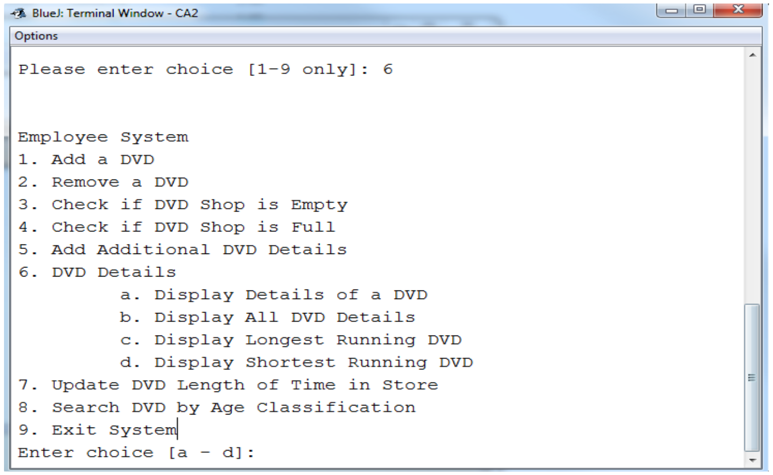
```
Enter the maximum number of dvds that can be stored in the shop: 10000

DVD System
1. Add a DVD
2. Remove a DVD
3. Check if DVD Shop is empty
4. Check if DVD Shop is full
5. Add Additional DVD Details
6. DVD Details
7. Update DVD Length of Time in Store
8. Search DVD by Age Classification
9. Exit System

Please enter choice [1-9 only]:
```

When the user selects option 6, a sub-menu is displayed to the user.

Please see the runtime version for the operation of these menus.



```
Please enter choice [1-9 only]: 6


Employee System
1. Add a DVD
2. Remove a DVD
3. Check if DVD Shop is Empty
4. Check if DVD Shop is Full
5. Add Additional DVD Details
6. DVD Details
          a. Display Details of a DVD
          b. Display All DVD Details
          c. Display Longest Running DVD
          d. Display Shortest Running DVD
7. Update DVD Length of Time in Store
8. Search DVD by Age Classification
9. Exit System
Enter choice [a - d]:
```

Information for Project

1. You **MAY** need additional methods in the blueprint classes.
2. Remember no printing out data in the blueprint classes.
3. When entering an age classification for the dvd you must ensure they user can only enter 12 or 15 or 18 (nothing else). You can put in error checking wherever it is needed.

Important Points

1. This is an individual project. Absolutely **NO** team efforts allowed.
2. Code **MUST** be commented.
3. Your program **MUST** run as sample program runs.
4. Any sign of cheating/copying will result in a **ZERO GRADE** for all students involved regardless of who wrote the original code.
5. By uploading your assignment, you are electronically signing the WIT anti-plagiarism declaration. Please see the WIT website for more details on this policy.
6. You will be interviewed on your programs when they are submitted to determine authorship and understanding. These interviews will take place on Thursday and Friday 2nd and 3rd of May 2019. Make a copy of your assignment for interview purposes and remove **ALL** comments prior to interview. You will only be interviewed on un-commented code.
7. You must demo your code or it will result in a **ZERO GRADE.**

8. You will receive a grade for your interview. This is a multiplier for your assignment grade.
9. Please submit by Thursday 2th May 2019 by 10am (however you can submit it early).
10. Late submissions will not be accepted.
11. You must submit the fully commented project via moodle (zip and upload, folder called YourName).
12. You must ensure you have uploaded the correct file as after the deadline no submission will be accepted.