Matt Weaver

final project documentation-client

C #

**Purpose:** the purpose of this program is to create user interface that talks to a server and stores information in a database. The information stored it information about DVDs

**Classes:** This program has six classes, five which our support classes, they are DVD, genre, utilities, menus, and responses. Sixth class client is the main program class.

**Note:** the entire project also contains a server program which is mentioned here, but not in great detail for more information about the server program see final project server documentation. Also due to the scope of this sentiment the main methods will be discussed as the main pseudocode instead of class documentation. However, the class documentation will so contain the names of the methods

**Menu Diagrams**

Main menu

Add menu

Update Menu

Delete or Undo menu

Search Menu

Select Menu

**class diagram**

|  |
| --- |
| **Menus** |
| -Static numbervalidation = (^\d +$):Regex  -static choice = 0: byte  -static temp: string |
| + Static Mainmenu (): void  + Static AddMenu (): void  + static UpdateMenu (): void  + static DeleteandUndomenu(): void  + static SelectMenu () : void  + static Search Menu (): void  + static genredescription (): void |

**Menus methods pseudocode**

1. **Main menu**
2. open try
3. clean the screen
4. display the following on the screen ignoring the "
5. "Welcome to DVD collection program. Please select an option from the menu below."
6. Insert a blank line
7. "Add DVD Press 1"
8. "Update DVD Press 2"
9. "Delete or Undo DVD Press 3"
10. "Search for DVD Press 4"
11. "Select DVD Press 5"
12. "Exit Press 6"
13. insert a blank line
14. "choice:"
15. temp = input
16. if number validation matches temp then
17. choice = temp converted to a byte
18. else Main menu method again
19. clear screen
20. use a switch statement for the choices. for the switch statement the cases equal the number option
21. pressing six will exit the application completely
22. default choice is main menu again
23. depending on the number call the appropriate menu option
24. close try
25. catch not implementation exception and a general exception
26. clear the screen
27. tell the user what the exception is. Suggest restarting the program
28. close catches
29. close method
30. be sure to line up the options so that their evenly spaced on the screen
31. **Add menu**
32. declare a new DVD object called d
33. open try
34. clear the screen
35. display the following on-screen ignoring the "
36. "Add Menu."
37. Insert a blank line
38. "Add Full DVD Press 1"
39. "Add Title Press 2"
40. "Back to Main Menu Press 3"
41. insert a blank line
42. "choice:"
43. choice = input
44. temp = input
45. if number validation matches temp then
46. choice = temp converted to a byte
47. else add menu method again
48. clear screen
49. use a switch statement for the choices. For the switch statement the cases equal the number option
50. default choice is add menu again
51. depending on the number call the appropriate menu option
52. close try
53. catch not implementation exception and a general exception
54. clear the screen
55. tell the user what the exception is and tell them to press any key to return to the main menu. Be sure to code this function as well.
56. close catches
57. be sure to line up the options so that their evenly spaced on the screen
58. close method
59. **update menu**
60. declare a new DVD object called d
61. open try
62. clear the screen
63. display the following on-screen ignoring the "
64. "Update Menu."
65. Insert a blank line
66. "Update Full DVD Press 1"
67. "Update Title Press 2"
68. "Update First Actor's Name Press 3"
69. "Update Second Actor's Name Press 4"
70. "Update Production Company Press 5 "
71. "Update Genre Press 6"
72. "Update Running Time Press 7"
73. "Update Type Press 8"
74. Back to Main Menu Press 9"
75. insert a blank line
76. "choice:"
77. temp = input
78. if number validation matches temp then
79. choice = temp converted to a byte
80. else update menu method again
81. clear screen
82. use a switch statement for the choices. For the switch statement the cases equal the number option
83. default choice is update menu's again
84. depending on the number call the appropriate menu option
85. close try
86. catch not implementation exception and a general exception
87. clear the screen
88. tell the user what the exception is and tell them to press any key to return to the main menu. Be sure to code this function as well.
89. be sure to line up the options so that their evenly spaced on the screen
90. close catches
91. **Delete or Undo menu**
92. declare a new DVD object called d
93. open try
94. clear the screen
95. display the following on-screen ignoring the "
96. "Delete & Undo Menu."
97. Insert a blank line
98. "Temporary Delete Press 1"
99. " Permanently delete Press 2"
100. " Undo Deletion Press 3"
101. " Back to Main Menu Press 4"
102. insert a blank line
103. "choice:"
104. temp = input
105. if number validation matches temp then
106. choice = temp converted to a byte
107. else call the delete and undo method
108. clear screen
109. use a switch statement for the choices. For the switch statement the cases equal the number option
110. default choice is the delete and undo menu again
111. depending on the number call the appropriate menu option
112. close try
113. catch not implementation exception and a general exception
114. clear the screen
115. tell the user what the exception is and tell them to press any key to return to the main menu. Be sure to code this function as well.
116. be sure to line up the options so that their evenly spaced on the screen
117. close catches
118. **Search menu**
119. declare a new DVD object called d
120. open try
121. clear the screen
122. display the following on-screen ignoring the "
123. "Search Menu."
124. "Search by Title Press 1"
125. "Search by First Actor Press 2"
126. "Search by Second Actor Press 3"
127. "Search by Genre Press 4"
128. "Search by Production Company Press 5"
129. "Search by Runtime Press 6"
130. "Search by Type Press 7"
131. "Search by Status Press 8
132. "Back to Main Menu Press 9"
133. insert a blank line
134. "choice:"
135. temp = input
136. if number validation matches temp then
137. choice = temp converted to a byte
138. else search menu method again
139. clear screen
140. use a switch statement for the choices. For the switch statement the cases equal the number option
141. default choice is the search menu again
142. depending on the number call the appropriate menu option
143. close try
144. catch not implementation exception and a general exception
145. clear the screen
146. tell the user what the exception is and tell them to press any key to return to the main menu. Be sure to code this function as well.
147. be sure to line up the options so that their evenly spaced on the screen
148. close catches
149. **Select menu**
150. declare a new DVD object called d
151. declare a byte called choice and set it equal to zero
152. open try
153. clear the screen
154. display the following on-screen ignoring the "
155. insert a blank line
156. "Select Menu."
157. "Select DVD by ID Press 1"
158. "Select DVD by Title Press 2"
159. "Back to Main Menu Press 3"
160. insert a blank line
161. "choice:"
162. temp = input
163. if number validation matches temp then
164. choice = temp converted to a byte
165. else call select menu again
166. clear screen
167. use a switch statement for the choices. For the switch statement the cases equal the number option
168. default choice is the select menu again
169. depending on the number call the appropriate menu option
170. close try
171. catch not implementation exception and a general exception
172. clear the screen
173. tell the user what the exception is and tell them to press any key to return to the main menu. Be sure to code this function as well.
174. be sure to line up the options so that their evenly spaced on the screen
175. close catches
176. close method

**class diagram**

|  |
| --- |
| **Genre** |
| **-**DVD\_Genre\_ID: short  -DVD\_Genre\_Descript: string |
| + Genre()-constructor  + getGenreID (): short  + getGenreDescription(): String  + setGenreID (genreID: short): void  + setGenreDescription(description: string): void |

**Method descriptions**

1. Genre-constructor

* DVD\_Genre\_ID = 0
* DVD\_Genre\_Descript = unknown

1. getGenreID

* return DVD\_Genre\_ID

1. setGenreID

* if genre ID is greater than or equal to zero and less than to 30
* DVD\_Genre\_ID = genreID
* else DVD\_Genre\_ID = 0

1. getGenreDescription

* return DVD\_Genre\_Descript

1. setGenreID

* if description is greater than zero and less than or equal to 20
* DVD\_Genre\_Descript = description
* else DVD\_Genre\_Descript = unknown

**class diagram**

|  |
| --- |
| **Utilities** |
| + Static reader: streamreader  + static writer:streamwriter  + static DVDclient: TCPclient =null |
| + Static GetNodeText (xpath: string,xdoc:XMLdocument): string  + static GetNodeText (xpath: string, XML: string): string  + static connect (): void  + static disconnect (): void |

**Method descriptions**

1. GetNodeText

* declare a string called value and set it equal to or empty
* declare an XML node called xnode set it equal to null
* open try
* set xNode equal to the XML document X path to extract a single note
* if xNode not equal to null
* value equals the inner text of the note
* close try
* catch any appropriate exceptions
* clear the screen for each exception
* let the user know what the error is
* set value equal to null
* return value;

1. GetNodeText

* declare a new XML document called XML doc set it equal to a new empty XML document
* declare a string called value and set it equal to or empty
* open try
* load the XML string into the XML document
* set value equal to GetNodeText with the X path and the XML document as parameters.
* Close try
* catch any appropriate exceptions
* clear the screen for each exception
* let the user know what the error is
* set value equal to null
* do the last three steps for each exception
* return value

1. Connect

* set DVD client equal to new TCP client with parameters of local host as a string and 5000 as an integer
* set reader to new stream reader with parameter DVD client get stream
* set writer to new stream writer with parameter DVD client get stream
* set the auto flush property of the writer equal to true

1. disconnect

* writer.write line equal to Request Action disconnect
* format the above statement to XML using Request and Action as the tags
* reader.read line
* close the writer, close the reader, close the client

**class diagram**

|  |
| --- |
| **DVD** |
| -DVD\_action: string  -DVD\_ID: string  -DVD\_title: string  -DVD\_first: string  -DVD\_last: string  -DVD\_first2: string  -DVD\_last2: string  -DVD\_cp: string  -DVD\_genre: short  -DVD\_rt: short  -DVD\_type: char  -DVD\_stat: char  -request: string  -response: string  -option: char  -option2: char  -option3: char  -movie: [ ] string  -temp: string  r: responses  -numbervalidation =^\d +$: Regex  -charactervalidation =^ [A-Za-z] {1} $: Regex |
| + DVD ()-constructor  + DVD (ID: string, title: string, first: string, last: string, first2: string, last2: string, company: string, genre: short, runtime: short, type: char, status: char)-constructor  +getAction (): string  + setAction (action: string): void  + getID ():string  + setID(ID:string):void  + getTitle ():string  + setTitle (title: string): void  +getFirstname():string  +setFirstname(first:string):void  +getLastname (): string  + setLastname (last: string): void  + +getFirstname2():string  +setFirstname2(first2:string):void  +getLastname2 (): string  + setLastname2 (last2: string): void  +getCompany ():string  +setCompany (company:string):void  +getGenre (): short  +setGenre (genre: short):void  +getRuntime (): short  +setRuntime (runtime: short):void  +getType (): char  +setType (type: char):void  +getStatus (): char  +setStatus (status: char):void  + AddAll (): void  + AddTitle (): void  + UpdateAll (): void  + UpdateTitle (): void  + UpdateActor1 (): void  + UpdateActor2 (): void  + UpdateCompany (): void  + UpdateGenre (): void  + UpdateRuntime (): void  + UpdateType (): void  + Delete (): void  + Purge (): void  + Recovery (): void  + GenreList (): void  + SearchTitle (): void  + SearchActor1 (): void  + SearchActor2 (): void  + SearchCompany (): void  + SearchGenre (): void  + SearchRuntime (): void  + SearchType (): void  + SearchStatus (): void  + SelectTitle (): void  + SelectID (): void |

**Method descriptions**

1. DVD-constructor

* set the variables to the following values
* DVD\_action =Empty;
* DVD\_ID = 0
* DVD\_title = Title
* DVD\_first = First
* DVD\_last = Last
* DVD\_first2 = First2
* DVD\_last2 = Last2
* DVD\_cp = Company
* DVD\_genre = 0
* DVD\_rt = 0
* DVD\_type = U
* DVD\_stat = E

1. DVD (DVD (ID: string, title: string, first: string, last: string, first2: string, last2: string, company: string, genre: short, runtime: short, type: char, status: char)-constructor

* set the setters to the appropriate parameter value

1. class getters

* for each getter return the appropriate private variable

1. SetAction

* if action equals one of the following values then
* DVD\_action = action
* else DVD\_action = empty
* values = addall, addtitle, updateall, updatetitle, updateactor1, updateactor2, updatecompany, updategenre, updateruntime, updatetype, searchtitle, searchactor1, searchactor2, searchcompany, searchgenre, searchruntime, searchtype, searchstatus , selectID, selecttitle, delete, temp,disconnect, getgenre

1. SetID

* if number validation matches ID then
* DVD\_ID = ID
* else if DVD\_ID = null
* DVD\_ID = "0"
* else DVD\_ID = "0"

1. SetTitle

* if title length is greater than zero and less than or equal to 50 then
* DVD\_title = title
* else if DVD\_title =null
* DVD\_title = "Title"
* else DVD\_title = "Title"

1. SetFirstname

* if first length is greater than zero and less than or equal to 30 then
* DVD\_first = first
* else if DVD\_first =null
* DVD\_first = "First"
* else DVD\_first = "First"

1. SetLastname

* if last length is greater than zero and less than or equal to 30 then
* DVD\_last = last
* else if DVD\_last =null
* DVD\_last = "last"
* else DVD\_last = "Last"

1. SetFirstname2

* if first2 length is greater than zero and less than or equal to 30 then
* DVD\_first2 = first2
* else if first2 =null
* DVD\_first2 = "First2"
* else DVD\_first2 = "First2"

1. SetLastname2

* if last2 length is greater than zero and less than or equal to 30 then
* DVD\_last2 = last 2
* else if DVD\_last2=null
* DVD\_last2 = "Last2"
* else DVD\_last2 = "Last2

1. SetCompany

* if company length is greater than zero and less than or equal to 40 then
* DVD\_cp= company
* else if DVD\_cp=null
* DVD\_cp= " Company"
* else DVD\_cp= " Company"

1. SetRuntime

* if runtime is greater than zero and less than or equal to 300 then
* DVD\_rt = runtime
* else DVD\_rt = 0

1. SetGenre

* if genre is greater than or equal to zero and less than 30 then
* DVD\_genre = genre
* else DVD\_genre = 0

1. SetType

* if type =D or B or U then
* DVD\_type = type
* else DVD\_type =U

1. SetStatus

* if status =E or N then
* DVD\_stat = status
* else DVD\_stat = E

**Note: below is the pseudocode for the rest of the methods in DVD**

**DVD methods pseudocode**

1. **AddAll**
2. declare a new string writer sw set it equal to an empty string writer
3. declare a new XML textwriter XML writer set it equal to a new XML textwriter with the string writer as a parameter
4. open try
5. clear the screen
6. display the following: Add All DVD Information
7. insert a blank line
8. open do while loop
9. display the following: View Genre Category List? (Y/N) :
10. insert a blank line
11. temp = input
12. if character validation matches temp then
13. option equals convert temp to character
14. convert option to uppercase
15. if option equals Y'
16. call the genre list method
17. close if statement
18. else if option equals 'N'
19. DVD\_action = addall
20. call the above setter and set the value equal to the variable
21. display the following information when necessary using write instead of write line
22. call the connect method from the utilities class
23. display the following: Title:
24. DVD\_title = input
25. call the above setter and set the value equal to the input
26. display the following: First leading actor first name:
27. DVD\_first = input
28. call the above setter and set the value equal to the input
29. display the following: First leading actor last name:
30. DVD\_last = input
31. call the above setter and set the value equal to the input
32. display the following: Second leading actor first name:
33. DVD\_first2= input
34. call the above setter and set the value equal to the input
35. display the following: Second leading actor last name:
36. DVD\_last2 = input
37. call the above setter and set the value equal to the input
38. display the following: Production Company:
39. DVD\_cp= input
40. call the above setter and set the value equal to the input
41. display the following: Genre Code Number:
42. temp = input
43. if number validation matches temp then
44. DVD\_genre = temp converted to a short
45. else DVD\_genre = 0
46. call the above setter and set the value equal to the input
47. display the following: Runtime in minutes:
48. temp = input
49. if number validation matches temp then
50. DVD\_rt= temp converted to a short
51. else DVD\_rt = 0
52. call the above setter and set the value equal to the input
53. display the following: Type (D for DVD or B for Blu-ray or U for unknown):
54. temp = input
55. if character validation matches temp then
56. DVD\_type = temp converted to a character
57. else DVD\_type = U
58. convert DVD\_type to uppercase
59. call the above setter and set the value equal to the input
60. DVD\_stat =' E'
61. call the above setter and set the value equal to the variable
62. insert a blank line
63. close if statement
64. close while loop. Loop while option not equal to ' Y' and option not equal to 'N'
65. display the following: please verify all information. Use write line
66. display the information as it was entered above make sure both the actors first and last name are on one line. Display the runtime as the following: runtime: value. Where value is the value from the getter. Everything else gets displayed on a separate line
67. use the getters to populate the output
68. insert a blank line
69. open do while
70. display the following: Correct? (Y/N)
71. temp = input
72. if character validation matches temp then
73. option2 equals convert temp to character
74. convert option2 to uppercase
75. if option2 equals 'N'
76. called the disconnect method from utilities class
77. call the add all method
78. close if
79. else if option2 equals 'Y'
80. start an XML document using Request as the start element
81. create XML document using Action, Title, First, Last, First2, Last2, Company, Genre, Runtime, Type, and Status as the tags. Use the getters to populate with the appropriate values that correspond to each tag. Remember to convert the values that are not strings into strings
82. after the XML document has been written close the textwriter
83. set request equal to the string writer variable
84. close the string writer
85. write the request to the server using the writer write line method from utilities class
86. set response equal to the reader read line method from utilities class
87. disconnect from the server using the disconnect method from the utilities class
88. call the add method from the responses class passing in the response as a parameter
89. close if statement
90. close while loop. Loop while option2 not equal to ' Y' and option2 not equal to 'N'
91. insert a blank line
92. open do while loop
93. display the following add another DVD? (Y/N):
94. temp = input
95. if character validation matches temp then
96. option3 equals convert temp to character
97. convert option3 to uppercase
98. if option 3 =' Y'
99. call the add all method
100. close if statement
101. else if option 3 =' N'
102. call the add menu method from the menu class
103. close if statement
104. close while loop. Loop while option3 not equal to ' Y' and option3 not equal to 'N'
105. close try
106. catch any exceptions that are relevant for this method. With each exception clear the screen. Tell the user what the error is and tell them to press any key to return to the add menu. Disconnect from the server using the disconnect method from utilities class close the textwriter and the string writer. Call the add menu method from the menus class. Be sure to code all these items into the program
107. **Add Title**
108. declare a new string writer sw set it equal to an empty string writer
109. declare a new XML textwriter XML writer set it equal to a new XML textwriter with the string writer as a parameter
110. open try
111. clear the screen
112. display the following: Add DVD title
113. insert a blank line
114. DVD\_action = addtitle
115. call the above setter and set the value equal to the variable
116. display the following information when necessary using write instead of write line
117. call the connect method from the utilities class
118. display the following: Title:
119. DVD\_title = input
120. call the above setter and set the value equal to the input
121. DVD\_stat =' E'
122. call the above setter and set the value equal to the variable
123. insert a blank line
124. display the following: please verify all information. Use write line
125. display the information as it was entered. Everything gets displayed on a separate line
126. use the getters to populate the output
127. insert a blank line
128. open do while
129. display the following: Correct? (Y/N)
130. temp = input
131. if character validation matches temp then
132. option equals convert temp to character
133. convert option to uppercase
134. if option equals 'N'
135. called the disconnect method from utilities class
136. call the add title method
137. close if
138. else if option equals 'Y'
139. start an XML document using Request as the start element
140. create XML document using Action, Title, and Status as the tags. Use the getters to populate with the appropriate values that correspond to each tag. Remember to convert the values that are not strings into strings
141. after the XML document has been written close the textwriter
142. set request equal to the string writer variable
143. close the string writer
144. write the request to the server using the writer write line method from utilities class
145. set response equal to the reader read line method from utilities class
146. disconnect from the server using the disconnect method from the utilities class
147. call the add method from the responses class passing in the response as a parameter
148. close if statement
149. close while loop. Loop while option not equal to ' Y' and option not equal to 'N'
150. insert a blank line
151. open do while loop
152. display the following add another DVD Title? (Y/N):
153. temp = input
154. if character validation matches temp then
155. option2 equals convert temp to character
156. convert option2 to uppercase
157. if option 2 =' Y'
158. call the add title method
159. close if statement
160. else if option 2=' N'
161. call the add menu method from the menu class
162. close if statement
163. close while loop. Loop while option2 not equal to ' Y' and option2 not equal to 'N'
164. close try
165. catch any exceptions that are relevant for this method. With each exception clear the screen. Tell the user what the error is and tell them to press any key to return to the add menu. Disconnect from the server using the disconnect method from utilities class close the textwriter and the string writer. Call the add menu method from the menus class. Be sure to code all these items into the program
166. **Delete**
167. declare a new string writer sw set it equal to an empty string writer
168. declare a new XML textwriter XML writer set it equal to a new XML textwriter with the string writer as a parameter
169. open try
170. clear the screen
171. display the following: Temporary Delete DVD
172. insert a blank line
173. DVD\_action = temp (word instead of variable)
174. call the above setter and set the value equal to the variable
175. display the following information when necessary using write instead of write line
176. call the connect method from the utilities class
177. display the following DVD ID:
178. DVD\_ID = input
179. call the above setter and set the value equal to the input
180. insert a blank line
181. display the following: please verify all information. Use write line
182. display the information as it was entered. Everything gets displayed on a separate line
183. use the getters to populate the output
184. insert a blank line
185. open do while
186. display the following: Correct? (Y/N)
187. temp = input
188. if character validation matches temp then
189. option equals convert temp to character
190. convert option to uppercase
191. if option equals 'N'
192. called the disconnect method from utilities class
193. call the Delete method
194. close if
195. else if option equals 'Y'
196. start an XML document using Request as the start element
197. create XML document using Action and ID as the tags. Use the getters to populate with the appropriate values that correspond to each tag. Remember to convert the values that are not strings into strings
198. after the XML document has been written close the textwriter
199. set request equal to the string writer variable
200. close the string writer
201. write the request to the server using the writer write line method from utilities class
202. set response equal to the reader read line method from utilities class
203. disconnect from the server using the disconnect method from the utilities class
204. call the modify method from the responses class passing in the response as a parameter
205. close if statement
206. close while loop. Loop while option not equal to ' Y' and option not equal to 'N'
207. insert a blank line
208. open do while loop
209. display the following Delete another DVD? (Y/N):
210. temp = input
211. if character validation matches temp then
212. option2 equals convert temp to character
213. convert option2 to uppercase
214. if option 2 =' Y'
215. call the Delete method
216. close if statement
217. else if option 2=' N'
218. call the delete and undo menu from the menu class
219. close if statement
220. close while loop. Loop while option2 not equal to ' Y' and option2 not equal to 'N'
221. close try
222. catch any exceptions that are relevant for this method. With each exception clear the screen. Tell the user what the error is and tell them to press any key to return to the delete and undo menu . Disconnect from the server using the disconnect method from utilities class close the textwriter and the string writer. Call the delete and undo menu method from the menus class. Be sure to code all these items into the program
223. **Genre List**
224. declare a new string writer sw set it equal to an empty string writer
225. declare a new XML textwriter XML writer set it equal to a new XML textwriter with the string writer as a parameter
226. declared by called choice and set it equal to zero
227. clear the screen
228. display the following: Genre Category List
229. insert a blank line
230. DVD\_action = getgenre
231. call the above setter and set the value equal to the variable
232. call the connect method from the utilities class
233. start an XML document using Request as the start element
234. create XML document using Action as the tag. Use the getters to populate with the appropriate values that correspond to each tag. Remember to convert the values that are not strings into strings
235. after the XML document has been written close the textwriter
236. set request equal to the string writer variable
237. close the string writer
238. write the request to the server using the writer write line method from utilities class
239. set response equal to the reader read line method from utilities class
240. disconnect from the server using the disconnect method from the utilities class
241. call the Genrelist method from the responses class passing in the response as a parameter
242. set the above step to the movie array
243. display the following: Code Genre. Separate the words with a tab
244. use a for loop to loop through the entire movie array.
245. Display in the following format: {0}\t {1}, I, movie [I]. Make sure each item is on a separate line
246. open do while loop
247. insert a blank line
248. display the following: Please select previous screen from list below
249. Insert a blank line
250. Add All Press 1"
251. " Update All Press 2"
252. "Update Genre Press 3"
253. "Search Genre Press 4"
254. insert a blank line
255. "choice:"
256. temp = input
257. if number validation matches temp then
258. choice = temp converted to a byte
259. use a switch statement for the choices. For the switch statement the cases equal the number option
260. default choice is the Genre List again
261. depending on the number call the appropriate menu option
262. loop while choice is not between 1-4
263. **Purge**
264. declare a new string writer sw set it equal to an empty string writer
265. declare a new XML textwriter XML writer set it equal to a new XML textwriter with the string writer as a parameter
266. open try
267. clear the screen
268. display the following: Purge DVD
269. insert a blank line
270. DVD\_action = delete
271. call the above setter and set the value equal to the variable
272. display the following information when necessary using write instead of write line
273. call the connect method from the utilities class
274. display the following DVD ID:
275. DVD\_ID = input
276. call the above setter and set the value equal to the input
277. insert a blank line
278. display the following: please verify all information. Please remember that this will completely remove all information of the DVD permanently. Use write line
279. display the information as it was entered. Everything gets displayed on a separate line
280. use the getters to populate the output
281. insert a blank line
282. open do while
283. display the following: Correct? (Y/N)
284. temp = input
285. if character validation matches temp then
286. option equals convert temp to character
287. convert option to uppercase
288. if option equals 'N'
289. called the disconnect method from utilities class
290. call the Purge method
291. close if
292. else if option equals 'Y'
293. start an XML document using Request as the start element
294. create XML document using Action and ID as the tags. Use the getters to populate with the appropriate values that correspond to each tag. Remember to convert the values that are not strings into strings
295. after the XML document has been written close the textwriter
296. set request equal to the string writer variable
297. close the string writer
298. write the request to the server using the writer write line method from utilities class
299. set response equal to the reader read line method from utilities class
300. disconnect from the server using the disconnect method from the utilities class
301. call the modify method from the responses class passing in the response as a parameter
302. close if statement
303. close while loop. Loop while option not equal to ' Y' and option not equal to 'N'
304. insert a blank line
305. open do while loop
306. display the following Purge another DVD? (Y/N):
307. temp = input
308. if character validation matches temp then
309. option2 equals convert temp to character
310. convert option2 to uppercase
311. if option 2 =' Y'
312. call the Purge method
313. close if statement
314. else if option 2=' N'
315. call the delete and undo menu from the menu class
316. close if statement
317. close while loop. Loop while option2 not equal to ' Y' and option2 not equal to 'N'
318. close try
319. catch any exceptions that are relevant for this method. With each exception clear the screen. Tell the user what the error is and tell them to press any key to return to the delete and undo menu . Disconnect from the server using the disconnect method from utilities class close the textwriter and the string writer. Call the delete and undo menu method from the menus class. Be sure to code all these items into the program
320. **Recovery**
321. declare a new string writer sw set it equal to an empty string writer
322. declare a new XML textwriter XML writer set it equal to a new XML textwriter with the string writer as a parameter
323. open try
324. clear the screen
325. display the following: Undo Deletion
326. insert a blank line
327. DVD\_action = recover
328. call the above setter and set the value equal to the variable
329. display the following information when necessary using write instead of write line
330. call the connect method from the utilities class
331. display the following DVD ID:
332. DVD\_ID = input
333. call the above setter and set the value equal to the input
334. insert a blank line
335. display the following: please verify all information. Use write line
336. display the information as it was entered. Everything gets displayed on a separate line
337. use the getters to populate the output
338. insert a blank line
339. open do while
340. display the following: Correct? (Y/N)
341. temp = input
342. if character validation matches temp then
343. option equals convert temp to character
344. convert option to uppercase
345. if option equals 'N'
346. called the disconnect method from utilities class
347. call the recovery method
348. close if
349. else if option equals 'Y'
350. start an XML document using Request as the start element
351. create XML document using Action and ID as the tags. Use the getters to populate with the appropriate values that correspond to each tag. Remember to convert the values that are not strings into strings
352. after the XML document has been written close the textwriter
353. set request equal to the string writer variable
354. close the string writer
355. write the request to the server using the writer write line method from utilities class
356. set response equal to the reader read line method from utilities class
357. disconnect from the server using the disconnect method from the utilities class
358. call the modify method from the responses class passing in the response as a parameter
359. close if statement
360. close while loop. Loop while option not equal to ' Y' and option not equal to 'N'
361. insert a blank line
362. open do while loop
363. display the following Recover another DVD? (Y/N):
364. temp = input
365. if character validation matches temp then
366. option2equals convert temp to character
367. convert option2 to uppercase
368. if option 2 =' Y'
369. call the Delete method
370. close if statement
371. else if option 2=' N'
372. call the delete and undo menu from the menu class
373. close if statement
374. close while loop. Loop while option2 not equal to ' Y' and option2 not equal to 'N'
375. close try
376. catch any exceptions that are relevant for this method. With each exception clear the screen. Tell the user what the error is and tell them to press any key to return to the delete and undo menu . Disconnect from the server using the disconnect method from utilities class close the textwriter and the string writer. Call the delete and undo menu method from the menus class. Be sure to code all these items into the program
377. **Search Actor1**
378. declare a new string writer sw set it equal to an empty string writer
379. declare a new XML textwriter XML writer set it equal to a new XML textwriter with the string writer as a parameter
380. open try
381. clear the screen
382. display the following: Search by First Actor
383. insert a blank line
384. open do while loop
385. DVD\_action = searchactor1
386. call the above setter and set the value equal to the variable
387. display the following information when necessary using write instead of write line
388. call the connect method from the utilities class
389. display the following: Last Name:
390. DVD\_last = input
391. if DVD\_last = empty then
392. called the disconnect method from utilities class
393. call Search Actor1 method again
394. call the above setter and set the value equal to the input
395. insert a blank line
396. start an XML document using Request as the start element
397. create XML document using Action, Last, as the tags. Use the getters to populate with the appropriate values that correspond to each tag. Remember to convert the values that are not strings into strings
398. after the XML document has been written close the textwriter
399. set request equal to the string writer variable
400. close the string writer
401. write the request to the server using the writer write line method from utilities class
402. set response equal to the reader read line method from utilities class
403. disconnect from the server using the disconnect method from the utilities class
404. call the SearchActor1 method from the responses class passing in the response as a parameter
405. set the above step to the movie array
406. display the following: First Name Last Name Title. Separate the words with a tab
407. use a for loop to loop through the entire movie array. Make sure to count by three when looping
408. Display in the following format: {0}\t \t{1}\t {2}, movie [I], movie [I +1], movie [I +2]. Make sure each item is on a separate line
409. open do while loop
410. display the following Start a new search by First Actor DVD? (Y/N):
411. temp = input
412. if character validation matches temp then
413. option equals convert temp to character
414. convert option to uppercase
415. if option =' Y'
416. call the SearchActor1 method
417. close if statement
418. else if option =' N'
419. call the search menu method from the menu class
420. close if statement
421. close while loop. Loop while option not equal to ' Y' and option not equal to 'N'
422. close try
423. catch any exceptions that are relevant for this method. With each exception clear the screen. Tell the user what the error is and tell them to press any key to return to the search menu. Disconnect from the server using the disconnect method from utilities class close the textwriter and the string writer. Call the search menu method from the menus class. Be sure to code all these items into the program
424. **Search Actor2**
425. declare a new string writer sw set it equal to an empty string writer
426. declare a new XML textwriter XML writer set it equal to a new XML textwriter with the string writer as a parameter
427. open try
428. clear the screen
429. display the following: Search by Second Actor
430. insert a blank line
431. open do while loop
432. DVD\_action = searchactor2
433. call the above setter and set the value equal to the variable
434. display the following information when necessary using write instead of write line
435. call the connect method from the utilities class
436. display the following: Last Name:
437. DVD\_last2 = input
438. if DVD\_last2 = empty then
439. called the disconnect method from utilities class
440. call Search Actor2 method again
441. call the above setter and set the value equal to the input
442. insert a blank line
443. start an XML document using Request as the start element
444. create XML document using Action, Last2, as the tags. Use the getters to populate with the appropriate values that correspond to each tag. Remember to convert the values that are not strings into strings
445. after the XML document has been written close the textwriter
446. set request equal to the string writer variable
447. close the string writer
448. write the request to the server using the writer write line method from utilities class
449. set response equal to the reader read line method from utilities class
450. disconnect from the server using the disconnect method from the utilities class
451. call the SearchActor2 method from the responses class passing in the response as a parameter
452. set the above step to the movie array
453. display the following: First Name Last Name Title. Separate the words with a tab
454. use a for loop to loop through the entire movie array. Make sure to count by three when looping
455. Display in the following format: {0}\t \t{1}\t {2}, movie [I], movie [I +1], movie [I +2]. Make sure each item is on a separate line
456. open do while loop
457. display the following Start a new search by Second Actor? (Y/N):
458. temp = input
459. if character validation matches temp then
460. option equals convert temp to character
461. convert option to uppercase
462. if option =' Y'
463. call the SearchActor2 method
464. close if statement
465. else if option =' N'
466. call the search menu method from the menu class
467. close if statement
468. close while loop. Loop while option not equal to ' Y' and option not equal to 'N'
469. close try
470. catch any exceptions that are relevant for this method. With each exception clear the screen. Tell the user what the error is and tell them to press any key to return to the search menu. Disconnect from the server using the disconnect method from utilities class close the textwriter and the string writer. Call the search menu method from the menus class. Be sure to code all these items into the program
471. **Search Company**
472. declare a new string writer sw set it equal to an empty string writer
473. declare a new XML textwriter XML writer set it equal to a new XML textwriter with the string writer as a parameter
474. open try
475. clear the screen
476. display the following: Search by Production Company
477. insert a blank line
478. open do while loop
479. DVD\_action = searchcompany
480. call the above setter and set the value equal to the variable
481. display the following information when necessary using write instead of write line
482. call the connect method from the utilities class
483. display the following: Production Company
484. DVD\_cp= input
485. if DVD\_cp= empty then
486. called the disconnect method from utilities class
487. call Search company method again
488. call the above setter and set the value equal to the input
489. insert a blank line
490. start an XML document using Request as the start element
491. create XML document using Action, Company, as the tags. Use the getters to populate with the appropriate values that correspond to each tag. Remember to convert the values that are not strings into strings
492. after the XML document has been written close the textwriter
493. set request equal to the string writer variable
494. close the string writer
495. write the request to the server using the writer write line method from utilities class
496. set response equal to the reader read line method from utilities class
497. disconnect from the server using the disconnect method from the utilities class
498. call the SearchCompany method from the responses class passing in the response as a parameter
499. set the above step to the movie array
500. display the following: Production Company Title. Separate the words with a tab
501. use a for loop to loop through the entire movie array. Make sure to count by two when looping
502. Display in the following format: {0}\t \t{1} movie [I], movie [I +1]. Make sure each item is on a separate line
503. open do while loop
504. display the following Start a new search By Production Company? (Y/N):
505. temp = input
506. if character validation matches temp then
507. option equals convert temp to character
508. convert option to uppercase
509. if option =' Y'
510. call the SearchCompany method
511. close if statement
512. else if option =' N'
513. call the search menu method from the menu class
514. close if statement
515. close while loop. Loop while option not equal to ' Y' and option not equal to 'N'
516. close try
517. catch any exceptions that are relevant for this method. With each exception clear the screen. Tell the user what the error is and tell them to press any key to return to the search menu. Disconnect from the server using the disconnect method from utilities class close the textwriter and the string writer. Call the search menu method from the menus class. Be sure to code all these items into the program
518. **search Genre**
519. declare a new string writer sw set it equal to an empty string writer
520. declare a new XML textwriter XML writer set it equal to a new XML textwriter with the string writer as a parameter
521. open try
522. clear the screen
523. display the following: Search by Genre Code Number
524. insert a blank line
525. open do while loop
526. display the following: View Genre Category List? (Y/N) :
527. insert a blank line
528. temp = input
529. if character validation matches temp then
530. option equals convert temp to character
531. convert option to uppercase
532. if option equals Y'
533. call the genre list method
534. close if statement
535. else if option equals 'N'
536. DVD\_action = searchgenre
537. call the above setter and set the value equal to the variable
538. display the following information when necessary using write instead of write line
539. call the connect method from the utilities class
540. display the following: Genre Code Number:
541. temp = input
542. if number validation matches temp then
543. DVD\_genre = temp converted to a short integer
544. else call the search genre method again
545. call the above setter and set the value equal to the input
546. close if statement
547. close while loop. Loop while option not equal to ' Y' and option not equal to 'N'
548. insert a blank line
549. start an XML document using Request as the start element
550. create XML document using Action, Genre, as the tags. Use the getters to populate with the appropriate values that correspond to each tag. Remember to convert the values that are not strings into strings
551. after the XML document has been written close the textwriter
552. set request equal to the string writer variable
553. close the string writer
554. write the request to the server using the writer write line method from utilities class
555. set response equal to the reader read line method from utilities class
556. disconnect from the server using the disconnect method from the utilities class
557. call the SearchGenre method from the responses class passing in the response as a parameter
558. set the above step to the movie array
559. display the following: Genre Category Title. Separate the words with a tab
560. use a for loop to loop through the entire movie array. Make sure to count by two when looping
561. Display in the following format: {0}\t \t{1} movie [I], movie [I +1]. Make sure each item is on a separate line
562. open do while loop
563. display the following Start a new search By Genre Code Number? (Y/N):
564. temp = input
565. if character validation matches temp then
566. option2 equals convert temp to character
567. convert option2 to uppercase
568. if option2 =' Y'
569. call the SearchGenre method
570. close if statement
571. else if option =' N'
572. call the search menu method from the menu class
573. close if statement
574. close while loop. Loop while option2 not equal to ' Y' and option2 not equal to 'N'
575. close try
576. catch any exceptions that are relevant for this method. With each exception clear the screen. Tell the user what the error is and tell them to press any key to return to the search menu. Disconnect from the server using the disconnect method from utilities class close the textwriter and the string writer. Call the search menu method from the menus class. Be sure to code all these items into the program
577. **Search Runtime**
578. declare a new string writer sw set it equal to an empty string writer
579. declare a new XML textwriter XML writer set it equal to a new XML textwriter with the string writer as a parameter
580. open try
581. clear the screen
582. display the following: Search by Runtime
583. insert a blank line
584. open do while loop
585. DVD\_action = searchruntime
586. call the above setter and set the value equal to the variable
587. display the following information when necessary using write instead of write line
588. call the connect method from the utilities class
589. display the following: Runtime in minutes
590. temp = input
591. if number validation matches temp then
592. DVD\_rt = temp converted to a short integer
593. else call search runtime method again
594. call the above setter and set the value equal to the input
595. insert a blank line
596. start an XML document using Request as the start element
597. create XML document using Action Runtime, as the tags. Use the getters to populate with the appropriate values that correspond to each tag. Remember to convert the values that are not strings into strings
598. after the XML document has been written close the textwriter
599. set request equal to the string writer variable
600. close the string writer
601. write the request to the server using the writer write line method from utilities class
602. set response equal to the reader read line method from utilities class
603. disconnect from the server using the disconnect method from the utilities class
604. call the SearchRuntime method from the responses class passing in the response as a parameter
605. set the above step to the movie array
606. display the following: Runtime in Minutes Title. Separate the words with a tab
607. use a for loop to loop through the entire movie array. Make sure to count by two when looping
608. Display in the following format: {0}\t \t{1} movie [I], movie [I +1]. Make sure each item is on a separate line
609. open do while loop
610. display the following Start a new search By Runtime? (Y/N):
611. temp = input
612. if character validation matches temp then
613. option equals convert temp to character
614. convert option to uppercase
615. if option =' Y'
616. call the SearchRuntime method
617. close if statement
618. else if option =' N'
619. call the search menu method from the menu class
620. close if statement
621. close while loop. Loop while option not equal to ' Y' and option not equal to 'N'
622. close try
623. catch any exceptions that are relevant for this method. With each exception clear the screen. Tell the user what the error is and tell them to press any key to return to the search menu. Disconnect from the server using the disconnect method from utilities class close the textwriter and the string writer. Call the search menu method from the menus class. Be sure to code all these items into the program
624. **Search Status**
625. declare a new string writer sw set it equal to an empty string writer
626. declare a new XML textwriter XML writer set it equal to a new XML textwriter with the string writer as a parameter
627. open try
628. clear the screen
629. display the following: Search by Status
630. insert a blank line
631. open do while loop
632. DVD\_action = searchstatus
633. call the above setter and set the value equal to the variable
634. display the following information when necessary using write instead of write line
635. call the connect method from the utilities class
636. display the following: Status (E for exist or N for not exists):
637. temp = input
638. if character validation matches temp then
639. DVD\_stat = temp converted to a character
640. else call search status method again
641. convert DVD\_stat to uppercase
642. call the above setter and set the value equal to the input
643. insert a blank line
644. start an XML document using Request as the start element
645. create XML document using Action, Status, as the tags. Use the getters to populate with the appropriate values that correspond to each tag. Remember to convert the values that are not strings into strings
646. after the XML document has been written close the textwriter
647. set request equal to the string writer variable
648. close the string writer
649. write the request to the server using the writer write line method from utilities class
650. set response equal to the reader read line method from utilities class
651. disconnect from the server using the disconnect method from the utilities class
652. call the SearchStatus method from the responses class passing in the response as a parameter
653. set the above step to the movie array
654. display the following: Status Title DVD ID. Separate the words with a tab
655. use a for loop to loop through the entire movie array. Make sure to count by three when looping
656. Display in the following format: {0}\t \t{1}\t {2} movie [I], movie [I +1], movie [I +2]. Make sure each item is on a separate line
657. open do while loop
658. display the following Start a new search By Production Company? (Y/N):
659. temp = input
660. if character validation matches temp then
661. option equals convert temp to character
662. convert option to uppercase
663. if option =' Y'
664. call the SearchStatus method
665. close if statement
666. else if option =' N'
667. call the search menu method from the menu class
668. close if statement
669. close while loop. Loop while option not equal to ' Y' and option not equal to 'N'
670. close try
671. catch any exceptions that are relevant for this method. With each exception clear the screen. Tell the user what the error is and tell them to press any key to return to the search menu. Disconnect from the server using the disconnect method from utilities class close the textwriter and the string writer. Call the search menu method from the menus class. Be sure to code all these items into the program
672. **Search Type**
673. declare a new string writer sw set it equal to an empty string writer
674. declare a new XML textwriter XML writer set it equal to a new XML textwriter with the string writer as a parameter
675. open try
676. clear the screen
677. display the following: Search by Type
678. insert a blank line
679. open do while loop
680. DVD\_action = searchtype
681. call the above setter and set the value equal to the variable
682. display the following information when necessary using write instead of write line
683. call the connect method from the utilities class
684. display the following: Type (D for DVD or B for Blu-ray or U For Unknown)
685. DVD\_type = input
686. convert DVD\_type to uppercase
687. call the above setter and set the value equal to the input
688. insert a blank line
689. start an XML document using Request as the start element
690. create XML document using Action, Type, as the tags. Use the getters to populate with the appropriate values that correspond to each tag. Remember to convert the values that are not strings into strings
691. after the XML document has been written close the textwriter
692. set request equal to the string writer variable
693. close the string writer
694. write the request to the server using the writer write line method from utilities class
695. set response equal to the reader read line method from utilities class
696. disconnect from the server using the disconnect method from the utilities class
697. call the SearchType method from the responses class passing in the response as a parameter
698. set the above step to the movie array
699. display the following: Type Title. Separate the words with a tab
700. use a for loop to loop through the entire movie array. Make sure to count by two when looping
701. Display in the following format: {0}\t \t{1} movie [I], movie [I +1]. Make sure each item is on a separate line
702. open do while loop
703. display the following Start a new search By Type? (Y/N):
704. temp = input
705. if character validation matches temp then
706. option equals convert temp to character
707. convert option to uppercase
708. if option =' Y'
709. call the SearchType method
710. close if statement
711. else if option =' N'
712. call the search menu method from the menu class
713. close if statement
714. close while loop. Loop while option not equal to ' Y' and option not equal to 'N'
715. close try
716. catch any exceptions that are relevant for this method. With each exception clear the screen. Tell the user what the error is and tell them to press any key to return to the search menu. Disconnect from the server using the disconnect method from utilities class close the textwriter and the string writer. Call the search menu method from the menus class. Be sure to code all these items into the program
717. **Search Title**
718. declare a new string writer sw set it equal to an empty string writer
719. declare a new XML textwriter XML writer set it equal to a new XML textwriter with the string writer as a parameter
720. open try
721. clear the screen
722. display the following: Search by Title
723. insert a blank line
724. open do while loop
725. DVD\_action = searchtitle
726. call the above setter and set the value equal to the variable
727. display the following information when necessary using write instead of write line
728. call the connect method from the utilities class
729. display the following: Title:
730. DVD\_title = input
731. if DVD\_title = empty then called search title method again
732. call the above setter and set the value equal to the input
733. insert a blank line
734. start an XML document using Request as the start element
735. create XML document using Action, Title as the tags. Use the getters to populate with the appropriate values that correspond to each tag. Remember to convert the values that are not strings into strings
736. after the XML document has been written close the textwriter
737. set request equal to the string writer variable
738. close the string writer
739. write the request to the server using the writer write line method from utilities class
740. set response equal to the reader read line method from utilities class
741. disconnect from the server using the disconnect method from the utilities class
742. call the SearchTitle method from the responses class passing in the response as a parameter
743. set the above step to the movie array
744. display the following: DVD ID Title. Separate the words with a tab
745. use a for loop to loop through the entire movie array. Make sure to count by two when looping
746. Display in the following format: {0}\t \t{1} movie [I], movie [I +1]. Make sure each item is on a separate line
747. open do while loop
748. display the following Start a new search By Title? (Y/N):
749. temp = input
750. if character validation matches temp then
751. option equals convert temp to character
752. convert option to uppercase
753. if option =' Y'
754. call the Search Title method
755. close if statement
756. else if option =' N'
757. call the search menu method from the menu class
758. close if statement
759. close while loop. Loop while option not equal to ' Y' and option not equal to 'N'
760. close try
761. catch any exceptions that are relevant for this method. With each exception clear the screen. Tell the user what the error is and tell them to press any key to return to the search menu. Disconnect from the server using the disconnect method from utilities class close the textwriter and the string writer. Call the search menu method from the menus class. Be sure to code all these items into the program
762. **Select ID**
763. declare a new string writer sw set it equal to an empty string writer
764. declare a new XML textwriter XML writer set it equal to a new XML textwriter with the string writer as a parameter
765. open try
766. clear the screen
767. display the following: Select DVD ID
768. insert a blank line
769. DVD\_action = select ID
770. call the above setter and set the value equal to the variable
771. display the following information when necessary using write instead of write line
772. call the connect method from the utilities class
773. display the following DVD ID:
774. DVD\_ID = input
775. call the above setter and set the value equal to the input
776. insert a blank line
777. display the following: please verify all information. Use write line
778. display the information as it was entered. Everything gets displayed on a separate line
779. use the getters to populate the output
780. insert a blank line
781. open do while
782. display the following: Correct? (Y/N)
783. temp = input
784. if character validation matches temp then
785. option equals convert temp to character
786. convert option to uppercase
787. if option equals 'N'
788. call the select ID method
789. close if
790. else if option equals 'Y'
791. start an XML document using Request as the start element
792. create XML document using Action and ID as the tags. Use the getters to populate with the appropriate values that correspond to each tag. Remember to convert the values that are not strings into strings
793. after the XML document has been written close the textwriter
794. set request equal to the string writer variable
795. close the string writer
796. write the request to the server using the writer write line method from utilities class
797. set response equal to the reader read line method from utilities class
798. disconnect from the server using the disconnect method from the utilities class
799. call the select method from the responses class passing in the response as a parameter
800. close if statement
801. close while loop. Loop while option not equal to ' Y' and option not equal to 'N'
802. insert a blank line
803. open do while loop
804. display the following: select another DVD by ID? (Y/N):
805. temp = input
806. if character validation matches temp then
807. option2 equals convert temp to character
808. convert option2 to uppercase
809. if option 2 =' Y'
810. call the select ID method
811. close if statement
812. else if option 2=' N'
813. call the select menu from the menu class
814. close if statement
815. close while loop. Loop while option2 not equal to ' Y' and option2 not equal to 'N'
816. close try
817. catch any exceptions that are relevant for this method. With each exception clear the screen. Tell the user what the error is and tell them to press any key to return to the select menu. Disconnect from the server using the disconnect method from utilities class close the textwriter and the string writer. Call the select menu method from the menus class. Be sure to code all these items into the program
818. **Select Title**
819. declare a new string writer sw set it equal to an empty string writer
820. declare a new XML textwriter XML writer set it equal to a new XML textwriter with the string writer as a parameter
821. open try
822. clear the screen
823. display the following: Select DVD Title
824. insert a blank line
825. DVD\_action = selecttitle
826. call the above setter and set the value equal to the variable
827. display the following information when necessary using write instead of write line
828. call the connect method from the utilities class
829. display the following Title:
830. DVD\_title = input
831. call the above setter and set the value equal to the input
832. insert a blank line
833. display the following: please verify all information. Use write line
834. display the information as it was entered. Everything gets displayed on a separate line
835. use the getters to populate the output
836. insert a blank line
837. open do while
838. display the following: Correct? (Y/N)
839. temp = input
840. if character validation matches temp then
841. option equals convert temp to character
842. convert option to uppercase
843. if option equals 'N'
844. call the select Title method
845. close if
846. else if option equals 'Y'
847. start an XML document using Request as the start element
848. create XML document using Action and Title as the tags. Use the getters to populate with the appropriate values that correspond to each tag. Remember to convert the values that are not strings into strings
849. after the XML document has been written close the textwriter
850. set request equal to the string writer variable
851. close the string writer
852. write the request to the server using the writer write line method from utilities class
853. set response equal to the reader read line method from utilities class
854. disconnect from the server using the disconnect method from the utilities class
855. call the select method from the responses class passing in the response as a parameter
856. close if statement
857. close while loop. Loop while option not equal to ' Y' and option not equal to 'N'
858. insert a blank line
859. open do while loop
860. display the following: select another DVD by Title? (Y/N):
861. temp = input
862. if character validation matches temp then
863. option2 equals convert temp to character
864. convert option2 to uppercase
865. if option 2 =' Y'
866. call the select Title method
867. close if statement
868. else if option 2=' N'
869. call the select menu from the menu class
870. close if statement
871. close while loop. Loop while option2 not equal to ' Y' and option2 not equal to 'N'
872. close try
873. catch any exceptions that are relevant for this method. With each exception clear the screen. Tell the user what the error is and tell them to press any key to return to the select menu. Disconnect from the server using the disconnect method from utilities class close the textwriter and the string writer. Call the select menu method from the menus class. Be sure to code all these items into the program
874. **UpdateAll**
875. declare a new string writer sw set it equal to an empty string writer
876. declare a new XML textwriter XML writer set it equal to a new XML textwriter with the string writer as a parameter
877. open try
878. clear the screen
879. display the following: Update All DVD Information
880. insert a blank line
881. open do while loop
882. display the following: View Genre Category List? (Y/N) :
883. insert a blank line
884. temp = input
885. if character validation matches temp then
886. option equals convert temp to character
887. convert option to uppercase
888. if option equals Y'
889. call the genre list method
890. close if statement
891. else if option equals 'N'
892. DVD\_action = updateall
893. call the above setter and set the value equal to the variable
894. display the following information when necessary using write instead of write line
895. call the connect method from the utilities class
896. display the following: DVD ID
897. DVD\_ID = input
898. call the above setter and set the value equal to the input
899. display the following: Title:
900. DVD\_title = input
901. call the above setter and set the value equal to the input
902. display the following: First leading actor first name:
903. DVD\_first = input
904. call the above setter and set the value equal to the input
905. display the following: First leading actor last name:
906. DVD\_last = input
907. call the above setter and set the value equal to the input
908. display the following: Second leading actor first name:
909. DVD\_first2= input
910. call the above setter and set the value equal to the input
911. display the following: Second leading actor last name:
912. DVD\_last2 = input
913. call the above setter and set the value equal to the input
914. display the following: Production Company:
915. DVD\_cp= input
916. call the above setter and set the value equal to the input
917. display the following: Genre Code Number:
918. temp = input
919. if number validation matches temp then
920. DVD\_genre = temp converted to a short
921. else DVD\_genre = 0
922. call the above setter and set the value equal to the input
923. display the following: Runtime in minutes:
924. temp = input
925. if number validation matches temp then
926. DVD\_rt= temp converted to a short
927. else DVD\_rt = 0
928. call the above setter and set the value equal to the input
929. display the following: Type (D for DVD or B for Blu-ray or U for unknown):
930. temp = input
931. if character validation matches temp then
932. DVD\_type = temp converted to a character
933. else DVD\_type = U
934. convert DVD\_type to uppercase
935. call the above setter and set the value equal to the input
936. insert a blank line
937. close if statement
938. close while loop. Loop while option not equal to ' Y' and option not equal to 'N'
939. display the following: please verify all information. Use write line
940. display the information as it was entered above make sure both the actors first and last name are on one line. Display the runtime as the following: runtime: value. Where value is the value from the getter. Everything else gets displayed on a separate line
941. use the getters to populate the output
942. insert a blank line
943. open do while
944. display the following: Correct? (Y/N)
945. temp = input
946. if character validation matches temp then
947. option2 equals convert temp to character convert
948. option2 to uppercase
949. if option2 equals 'N'
950. call the updateall method
951. close if
952. else if option2 equals 'Y'
953. start an XML document using Request as the start element
954. create XML document using Action, ID, Title, First, Last, First2, Last2, Company, Genre, Runtime, Type, and Status as the tags. Use the getters to populate with the appropriate values that correspond to each tag. Remember to convert the values that are not strings into strings
955. after the XML document has been written close the textwriter
956. set request equal to the string writer variable
957. close the string writer
958. write the request to the server using the writer write line method from utilities class
959. set response equal to the reader read line method from utilities class
960. disconnect from the server using the disconnect method from the utilities class
961. call the modify method from the responses class passing in the response as a parameter
962. close if statement
963. close while loop. Loop while option2 not equal to ' Y' and option2 not equal to 'N'
964. insert a blank line
965. open do while loop
966. display the following Update another DVD? (Y/N):
967. temp = input
968. if character validation matches temp then
969. option3 equals convert temp to character
970. convert option3 to uppercase
971. if option 3 =' Y'
972. call the update all method
973. close if statement
974. else if option 3 =' N'
975. call the update menu method from the menu class
976. close if statement
977. close while loop. Loop while option3 not equal to ' Y' and option3 not equal to 'N'
978. close try
979. catch any exceptions that are relevant for this method. With each exception clear the screen. Tell the user what the error is and tell them to press any key to return to the update menu. Disconnect from the server using the disconnect method from utilities class close the textwriter and the string writer. Call the update menu method from the menus class. Be sure to code all these items into the program
980. **UpdateActor1**
981. declare a new string writer sw set it equal to an empty string writer
982. declare a new XML textwriter XML writer set it equal to a new XML textwriter with the string writer as a parameter
983. open try
984. clear the screen
985. display the following: Update DVD First Actor
986. insert a blank line
987. DVD\_action = updateactor1
988. call the above setter and set the value equal to the variable
989. display the following information when necessary using write instead of write line
990. call the connect method from the utilities class
991. display the following: DVD ID
992. DVD\_ID = input
993. call the above setter and set the value equal to the input
994. display the following: First Name:
995. DVD\_first = input
996. call the above setter and set the value equal to the input
997. display the following: Last Name:
998. DVD\_last = input
999. call the above setter and set the value equal to the input
1000. display the following: please verify all information. Use write line
1001. display the information as it was entered above make sure both the actors first and last name are on one line.. Everything else gets displayed on a separate line
1002. use the getters to populate the output
1003. insert a blank line
1004. open do while
1005. display the following: Correct? (Y/N)
1006. temp = input
1007. if character validation matches temp then
1008. option equals convert temp to character
1009. convert option to uppercase
1010. if option equals 'N'
1011. call the updateactor1 method
1012. close if
1013. else if option equals 'Y'
1014. start an XML document using Request as the start element
1015. create XML document using Action, ID, First, Last, as the tags. Use the getters to populate with the appropriate values that correspond to each tag. Remember to convert the values that are not strings into strings
1016. after the XML document has been written close the textwriter
1017. set request equal to the string writer variable
1018. close the string writer
1019. write the request to the server using the writer write line method from utilities class
1020. set response equal to the reader read line method from utilities class
1021. disconnect from the server using the disconnect method from the utilities class
1022. call the modify method from the responses class passing in the response as a parameter
1023. close if statement
1024. close while loop. Loop while option not equal to ' Y' and option not equal to 'N'
1025. insert a blank line
1026. open do while loop
1027. display the following Update another DVD First Actor? (Y/N):
1028. temp = input
1029. if character validation matches temp then
1030. option2 equals convert temp to character
1031. convert option2 to uppercase
1032. if option2=' Y'
1033. call the updateactor1method
1034. close if statement
1035. else if option2=' N'
1036. call the update menu method from the menu class
1037. close if statement
1038. close while loop. Loop while option2 not equal to ' Y' and option2 not equal to 'N'
1039. close try
1040. catch any exceptions that are relevant for this method. With each exception clear the screen. Tell the user what the error is and tell them to press any key to return to the update menu. Disconnect from the server using the disconnect method from utilities class close the textwriter and the string writer. Call the update menu method from the menus class. Be sure to code all these items into the program
1041. **UpdateActor2**
1042. declare a new string writer sw set it equal to an empty string writer
1043. declare a new XML textwriter XML writer set it equal to a new XML textwriter with the string writer as a parameter
1044. open try
1045. clear the screen
1046. display the following: Update DVD Second Actor
1047. insert a blank line
1048. DVD\_action = updateactor2
1049. call the above setter and set the value equal to the variable
1050. display the following information when necessary using write instead of write line
1051. call the connect method from the utilities class
1052. display the following: DVD ID
1053. DVD\_ID = input
1054. call the above setter and set the value equal to the input
1055. display the following: First Name
1056. DVD\_first2 = input
1057. call the above setter and set the value equal to the input
1058. display the following: Last Name:
1059. DVD\_last2 = input
1060. call the above setter and set the value equal to the input
1061. display the following: please verify all information. Use write line
1062. display the information as it was entered above make sure both the actors first and last name are on one line.. Everything else gets displayed on a separate line
1063. use the getters to populate the output
1064. insert a blank line
1065. open do while
1066. display the following: Correct? (Y/N)
1067. temp = input
1068. if character validation matches temp then
1069. option equals convert temp to character
1070. convert option to uppercase
1071. if option equals 'N'
1072. call the updateactor2 method
1073. close if
1074. else if option equals 'Y'
1075. start an XML document using Request as the start element
1076. create XML document using Action, ID, First2, Last2, as the tags. Use the getters to populate with the appropriate values that correspond to each tag. Remember to convert the values that are not strings into strings
1077. after the XML document has been written close the textwriter
1078. set request equal to the string writer variable
1079. close the string writer
1080. write the request to the server using the writer write line method from utilities class
1081. set response equal to the reader read line method from utilities class
1082. disconnect from the server using the disconnect method from the utilities class
1083. call the modify method from the responses class passing in the response as a parameter
1084. close if statement
1085. close while loop. Loop while option not equal to ' Y' and option not equal to 'N'
1086. insert a blank line
1087. open do while loop
1088. display the following Update another DVD Second Actor? (Y/N):
1089. temp = input
1090. if character validation matches temp then
1091. option2 equals convert temp to character
1092. convert option2 to uppercase
1093. if option2=' Y'
1094. call the UpdateActor 2method
1095. close if statement
1096. else if option2=' N'
1097. call the update menu method from the menu class
1098. close if statement
1099. close while loop. Loop while option2 not equal to ' Y' and option2 not equal to 'N'
1100. close try
1101. catch any exceptions that are relevant for this method. With each exception clear the screen. Tell the user what the error is and tell them to press any key to return to the update menu. Disconnect from the server using the disconnect method from utilities class close the textwriter and the string writer. Call the update menu method from the menus class. Be sure to code all these items into the program
1102. **UpdateCompany**
1103. declare a new string writer sw set it equal to an empty string writer
1104. declare a new XML textwriter XML writer set it equal to a new XML textwriter with the string writer as a parameter
1105. open try
1106. clear the screen
1107. display the following: Update DVD Company
1108. insert a blank line
1109. DVD\_action = updatecompany
1110. call the above setter and set the value equal to the variable
1111. display the following information when necessary using write instead of write line
1112. call the connect method from the utilities class
1113. display the following: DVD ID
1114. DVD\_ID = input
1115. call the above setter and set the value equal to the input
1116. display the following: Production Company:
1117. DVD\_cp= input
1118. call the above setter and set the value equal to the input
1119. display the following: please verify all information. Use write line
1120. display the information as it was entered above. Everything gets displayed on a separate line
1121. use the getters to populate the output
1122. insert a blank line
1123. open do while
1124. display the following: Correct? (Y/N)
1125. temp = input
1126. if character validation matches temp then
1127. option equals convert temp to character
1128. convert option to uppercase
1129. if option equals 'N'
1130. call the update company method
1131. close if
1132. else if option equals 'Y'
1133. start an XML document using Request as the start element
1134. create XML document using Action, ID, Company as the tags. Use the getters to populate with the appropriate values that correspond to each tag. Remember to convert the values that are not strings into strings
1135. after the XML document has been written close the textwriter
1136. set request equal to the string writer variable
1137. close the string writer
1138. write the request to the server using the writer write line method from utilities class
1139. set response equal to the reader read line method from utilities class
1140. disconnect from the server using the disconnect method from the utilities class
1141. call the modify method from the responses class passing in the response as a parameter
1142. close if statement
1143. close while loop. Loop while option not equal to ' Y' and option not equal to 'N'
1144. insert a blank line
1145. open do while loop
1146. display the following Update DVD company? (Y/N):
1147. temp = input
1148. if character validation matches temp then
1149. option2 equals convert temp to character
1150. convert option2 to uppercase
1151. if option 2 =' Y'
1152. call the update company method
1153. close if statement
1154. else if option2=' N'
1155. call the update menu method from the menu class
1156. close if statement
1157. close while loop. Loop while option2 not equal to ' Y' and option2 not equal to 'N'
1158. close try
1159. catch any exceptions that are relevant for this method. With each exception clear the screen. Tell the user what the error is and tell them to press any key to return to the update menu. Disconnect from the server using the disconnect method from utilities class close the textwriter and the string writer. Call the update menu method from the menus class. Be sure to code all these items into the program
1160. **Update genre**
1161. declare a new string writer sw set it equal to an empty string writer
1162. declare a new XML textwriter XML writer set it equal to a new XML textwriter with the string writer as a parameter
1163. open try
1164. clear the screen
1165. display the following: Update DVD Genre Code Number is
1166. insert a blank line
1167. open do while loop
1168. display the following: View Genre Category List? (Y/N) :
1169. insert a blank line
1170. temp = input
1171. if character validation matches temp then
1172. option equals convert temp to character
1173. convert option to uppercase
1174. if option equals Y'
1175. call the genre list method
1176. close if statement
1177. else if option equals 'N'
1178. DVD\_action = updategenre
1179. call the above setter and set the value equal to the variable
1180. display the following information when necessary using write instead of write line
1181. call the connect method from the utilities class
1182. display the following: DVD ID
1183. DVD\_ID = input
1184. call the above setter and set the value equal to the input
1185. display the following: Genre Code Number:
1186. temp = input
1187. if number validation matches temp then
1188. DVD\_genre = temp converted to a short
1189. else DVD\_genre = 0
1190. call the above setter and set the value equal to the input
1191. insert a blank line
1192. close if statement
1193. close while loop. Loop while option not equal to ' Y' and option not equal to 'N'
1194. display the following: please verify all information. Use write line
1195. display the information as it was entered above . Everything else gets displayed on a separate line
1196. use the getters to populate the output
1197. insert a blank line
1198. open do while
1199. display the following: Correct? (Y/N)
1200. temp = input
1201. if character validation matches temp then
1202. option equals convert temp to character
1203. convert option2 to uppercase
1204. if option2 equals 'N'
1205. call the updategenre method
1206. close if
1207. else if option2 equals 'Y'
1208. start an XML document using Request as the start element
1209. create XML document using Action, ID, Title, Genre, as the tags. Use the getters to populate with the appropriate values that correspond to each tag. Remember to convert the values that are not strings into strings
1210. after the XML document has been written close the textwriter
1211. set request equal to the string writer variable
1212. close the string writer
1213. write the request to the server using the writer write line method from utilities class
1214. set response equal to the reader read line method from utilities class
1215. disconnect from the server using the disconnect method from the utilities class
1216. call the modify method from the responses class passing in the response as a parameter
1217. close if statement
1218. close while loop. Loop while option2 not equal to ' Y' and option2 not equal to 'N'
1219. insert a blank line
1220. open do while loop
1221. display the following Update another DVD Genre? (Y/N):
1222. temp = input
1223. if character validation matches temp then
1224. option equals convert temp to character
1225. convert option3 to uppercase
1226. if option 3 =' Y'
1227. call the updategenre method
1228. close if statement
1229. else if option 3 =' N'
1230. call the update menu method from the menu class
1231. close if statement
1232. close while loop. Loop while option3 not equal to ' Y' and option3 not equal to 'N'
1233. close try
1234. catch any exceptions that are relevant for this method. With each exception clear the screen. Tell the user what the error is and tell them to press any key to return to the update menu. Disconnect from the server using the disconnect method from utilities class close the textwriter and the string writer. Call the update menu method from the menus class. Be sure to code all these items into the program
1235. **Update Runtime**
1236. declare a new string writer sw set it equal to an empty string writer
1237. declare a new XML textwriter XML writer set it equal to a new XML textwriter with the string writer as a parameter
1238. open try
1239. clear the screen
1240. display the following: Update DVD Runtime
1241. insert a blank line
1242. DVD\_action = updateruntime
1243. call the above setter and set the value equal to the variable
1244. display the following information when necessary using write instead of write line
1245. call the connect method from the utilities class
1246. display the following: DVD ID
1247. DVD\_ID = input
1248. call the above setter and set the value equal to the input
1249. display the following: Runtime in minutes:
1250. temp = input
1251. if number validation matches temp then
1252. DVD\_rt= temp converted to a short
1253. else DVD\_rt = 0
1254. call the above setter and set the value equal to the input
1255. display the following: please verify all information. Use write line
1256. display the information as it was entered above Display the runtime as the following: runtime: value. Where value is the value from the getter. Everything else gets displayed on a separate line
1257. use the getters to populate the output
1258. insert a blank line
1259. open do while
1260. display the following: Correct? (Y/N)
1261. temp = input
1262. if character validation matches temp then
1263. option equals convert temp to character
1264. convert option to uppercase
1265. if option equals 'N'
1266. call the update runtime method
1267. close if
1268. else if option equals 'Y'
1269. start an XML document using Request as the start element
1270. create XML document using Action, ID, Runtime as the tags. Use the getters to populate with the appropriate values that correspond to each tag. Remember to convert the values that are not strings into strings
1271. after the XML document has been written close the textwriter
1272. set request equal to the string writer variable
1273. close the string writer
1274. write the request to the server using the writer write line method from utilities class
1275. set response equal to the reader read line method from utilities class
1276. disconnect from the server using the disconnect method from the utilities class
1277. call the modify method from the responses class passing in the response as a parameter
1278. close if statement
1279. close while loop. Loop while option not equal to ' Y' and option not equal to 'N'
1280. insert a blank line
1281. open do while loop
1282. display the following Update another DVD runtime? (Y/N):
1283. temp = input
1284. if character validation matches temp then
1285. option2 equals convert temp to character
1286. convert option2to uppercase
1287. if option 2 =' Y'
1288. call the update runtime method
1289. close if statement
1290. else if option2=' N'
1291. call the update menu method from the menu class
1292. close if statement
1293. close while loop. Loop while option2 not equal to ' Y' and option2 not equal to 'N'
1294. close try
1295. catch any exceptions that are relevant for this method. With each exception clear the screen. Tell the user what the error is and tell them to press any key to return to the update menu. Disconnect from the server using the disconnect method from utilities class close the textwriter and the string writer. Call the update menu method from the menus class. Be sure to code all these items into the program
1296. **Update Title**
1297. declare a new string writer sw set it equal to an empty string writer
1298. declare a new XML textwriter XML writer set it equal to a new XML textwriter with the string writer as a parameter
1299. open try
1300. clear the screen
1301. display the following: Update DVD Title
1302. insert a blank line
1303. DVD\_action = updatetitle
1304. call the above setter and set the value equal to the variable
1305. display the following information when necessary using write instead of write line
1306. call the connect method from the utilities class
1307. display the following: DVD ID
1308. DVD\_ID = input
1309. call the above setter and set the value equal to the input
1310. display the following: Title:
1311. DVD\_title = input
1312. call the above setter and set the value equal to the input
1313. display the following: please verify all information. Use write line
1314. display the information as it was entered above. Everything gets displayed on a separate line
1315. use the getters to populate the output
1316. insert a blank line
1317. open do while
1318. display the following: Correct? (Y/N)
1319. temp = input
1320. if character validation matches temp then
1321. option equals convert temp to character
1322. convert option to uppercase
1323. if option equals 'N'
1324. call the update title method
1325. close if
1326. else if option equals 'Y'
1327. start an XML document using Request as the start element
1328. create XML document using Action, ID, Title as the tags. Use the getters to populate with the appropriate values that correspond to each tag. Remember to convert the values that are not strings into strings
1329. after the XML document has been written close the textwriter
1330. set request equal to the string writer variable
1331. close the string writer
1332. write the request to the server using the writer write line method from utilities class
1333. set response equal to the reader read line method from utilities class
1334. disconnect from the server using the disconnect method from the utilities class
1335. call the modify method from the responses class passing in the response as a parameter
1336. close if statement
1337. close while loop. Loop while option not equal to ' Y' and option not equal to 'N'
1338. insert a blank line
1339. open do while loop
1340. display the following Update another DVD Title? (Y/N):
1341. temp = input
1342. if character validation matches temp then
1343. option 2equals convert temp to character
1344. convert option2to uppercase
1345. if option 2 =' Y'
1346. call the update title method
1347. close if statement
1348. else if option2=' N'
1349. call the update menu method from the menu class
1350. close if statement
1351. close while loop. Loop while option2 not equal to ' Y' and option2 not equal to 'N'
1352. close try
1353. catch any exceptions that are relevant for this method. With each exception clear the screen. Tell the user what the error is and tell them to press any key to return to the update menu. Disconnect from the server using the disconnect method from utilities class close the textwriter and the string writer. Call the update menu method from the menus class. Be sure to code all these items into the program
1354. **Update Type**
1355. declare a new string writer sw set it equal to an empty string writer
1356. declare a new XML textwriter XML writer set it equal to a new XML textwriter with the string writer as a parameter
1357. open try
1358. clear the screen
1359. display the following: Update DVD Type
1360. insert a blank line
1361. DVD\_action = updatetype
1362. call the above setter and set the value equal to the variable
1363. display the following information when necessary using write instead of write line
1364. call the connect method from the utilities class
1365. display the following: DVD ID
1366. DVD\_ID = input
1367. call the above setter and set the value equal to the input
1368. display the following: Type (D for DVD or B for Blu-ray or U for unknown):
1369. temp = input
1370. if character validation matches temp then
1371. DVD\_type = temp converted to a character
1372. else DVD\_type = U
1373. convert DVD\_type to uppercase
1374. call the above setter and set the value equal to the input
1375. display the following: please verify all information. Use write line
1376. display the information as it was entered above. Everything gets displayed on a separate line
1377. use the getters to populate the output
1378. insert a blank line
1379. open do while
1380. display the following: Correct? (Y/N)
1381. temp = input
1382. if character validation matches temp then
1383. option equals convert temp to character
1384. convert option to uppercase
1385. if option equals 'N'
1386. call the update type method
1387. close if
1388. else if option equals 'Y'
1389. start an XML document using Request as the start element
1390. create XML document using Action, ID, Type as the tags. Use the getters to populate with the appropriate values that correspond to each tag. Remember to convert the values that are not strings into strings
1391. after the XML document has been written close the textwriter
1392. set request equal to the string writer variable
1393. close the string writer
1394. write the request to the server using the writer write line method from utilities class
1395. set response equal to the reader read line method from utilities class
1396. disconnect from the server using the disconnect method from the utilities class
1397. call the modify method from the responses class passing in the response as a parameter
1398. close if statement
1399. close while loop. Loop while option not equal to ' Y' and option not equal to 'N'
1400. insert a blank line
1401. open do while loop
1402. display the following Update DVD runtime? (Y/N):
1403. temp = input
1404. if character validation matches temp then
1405. option2 equals convert temp to character
1406. convert option2to uppercase
1407. if option 2 =' Y'
1408. call the update type method
1409. close if statement
1410. else if option2=' N'
1411. call the update menu method from the menu class
1412. close if statement
1413. close while loop. Loop while option2 not equal to ' Y' and option2 not equal to 'N'
1414. close try
1415. catch any exceptions that are relevant for this method. With each exception clear the screen. Tell the user what the error is and tell them to press any key to return to the update menu. Disconnect from the server using the disconnect method from utilities class close the textwriter and the string writer. Call the update menu method from the menus class. Be sure to code all these items into the program.

**class diagram**

|  |
| --- |
| **Responses** |
| **-**ID: string  -title: string  -first: string  -last: string  -first2: string  -last2: string  -company: string  -genre: string  -description: string  -runtime: string  -type: string  -status: string  -code: string  -message: string  -Value1 = 0: int: const  -Value2 =1: int: const  -Value3 = 2: int: const |
| + Add (Response: String): void  + Modify (Response: String): void  + Select (Response: String): void  + GenreList (Response: String): [] string  + Search actor1 (Response: String): [] string  + Search actor2 (Response: String): [] string  + Search company (Response: String): [] string  + Search genre (Response: String): [] string  + Search runtime (Response: String): [] string  + Search status (Response: String): [] string  + Search type (Response: String): [] string  + SearchTitle (Response: String): [] string |

**Responses methods pseudocode**

1. **Add**
2. ID = GetNodeText from the utilities class with Response/ID, Response as parameters
3. if ID not equal to empty them
4. display the following: DVD was added with the idea of ID. Where ID is the ID gotten by the GetNodeText method
5. close if statement
6. else message = GetNodeText from the utilities class with Response/Message, Response as parameters
7. display the message on the screen
8. close else statement
9. **Modify**
10. code = GetNodeText from the utilities class with Response/ErrorCode, Response as parameters
11. message = GetNodeText from the utilities class with Response/Message as parameters
12. if code = 0 then
13. insert a blank line
14. display the message on the screen
15. insert a blank line
16. close if statement
17. else insert a blank line
18. display the message on the screen
19. insert a blank line
20. close else statement
21. **Select**
22. code = GetNodeText from the utilities class with Response/ErrorCode, Response as parameters
23. if code = 0 then
24. set all the remaining private variables are equal to GetNodeText from the utilities class with Response/variable, response as parameters where valuable is the current private variable name
25. display the information on screen in the same order that was taken out of the XML document. Both first and last names of the actors on the same line. Each actor gets put on a separate line. Display runtime as follows runtime: value minutes. Where value is the value assigned to the runtime variable. Everything else is put on a separate line
26. close if statement
27. else insert a blank line
28. message = GetNodeText from the utilities class with Response/Message as parameters
29. display the message on the screen
30. insert a blank line
31. close else statement
32. **GenreList**
33. create a new link list of strings called list and set it equal to an empty link list
34. create a new XML document called XML doc and set it equal to an empty XML document
35. load the XML document with response
36. code = GetNodeText from the utilities class with Response/ErrorCode, Response as parameters
37. if code = 0 then
38. create a new XML node list called node list and set it equal to XML doc select nodes method with Response/Genres/Genre as a parameter
39. set ID and description to empty
40. use a for each loop to loop through the node list
41. for each node in the node list ID equals the first item at the first index in the node. The index value is stored in the variable value1
42. description equals the second item at the second index in the node the index value is stored in the variable value2
43. add description to the end of list
44. close if statement
45. else message = GetNodeText from the utilities class with Response/Message as parameters
46. add message to the end of list
47. close else statement
48. return list as an array
49. **search actor1**
50. create a new link list of strings called actor1 results and set it equal to an empty link list
51. create a new XML document called XML doc and set it equal to an empty XML document
52. load the XML document with response
53. code = GetNodeText from the utilities class with Response/ErrorCode, Response as parameters
54. if code = 0 then
55. create a new XML node list called node list and set it equal to XML doc select nodes method with Response/DVDS /DVD as a parameter
56. set title, first and last to empty
57. use a for each loop to loop through the node list
58. for each node in the node list first equals the first item at the first index in the node. The index value is stored in the variable value1
59. last equals the second item at the second index in the node, the index value is stored in the variable value2
60. title equals the third item at the third index in the node, the index value is stored in the variable value3
61. add first to the end of actor 1 results
62. add last to the end of actor 1 results
63. add title to the end of actor1 results
64. close if statement
65. else message = GetNodeText from the utilities class with Response/Message as parameters
66. add message to the end of actor1 results
67. close else statement
68. return actor 1results as an array
69. **search actor2**
70. create a new link list of strings called actor 2results and set it equal to an empty link list
71. create a new XML document called XML doc and set it equal to an empty XML document
72. load the XML document with response
73. code = GetNodeText from the utilities class with Response/ErrorCode, Response as parameters
74. if code = 0 then
75. create a new XML node list called node list and set it equal to XML doc select nodes method with Response/DVDS /DVD as a parameter
76. set title, first2 and last2 to empty
77. use a for each loop to loop through the node list
78. for each node in the node list first2 equals the first item at the first index in the node. The index value is stored in the variable value1
79. last2 equals the second item at the second index in the node, the index value is stored in the variable value2
80. title equals the third item at the third index in the node, the index value is stored in the variable value3
81. add first 2to the end of actor 2 results
82. add last2 to the end of actor 2 results
83. add title to the end of actor 2results
84. close if statement
85. else message = GetNodeText from the utilities class with Response/Message as parameters
86. add message to the end of actor2 results
87. close else statement
88. return last name2 results as an array
89. **search status**
90. create a new link list of strings called status results and set it equal to an empty link list
91. create a new XML document called XML doc and set it equal to an empty XML document
92. load the XML document with response
93. code = GetNodeText from the utilities class with Response/ErrorCode, Response as parameters
94. if code = 0 then
95. create a new XML node list called node list and set it equal to XML doc select nodes method with Response/DVDS /DVD as a parameter
96. set title, ID and status to empty
97. use a for each loop to loop through the node list
98. for each node in the node list status equals the first item at the first index in the node. The index value is stored in the variable value1
99. title equals the second item at the second index in the node, the index value is stored in the variable value2
100. ID equals the third item at the third index in the node, the index value is stored in the variable value3
101. add status to the end of status results
102. add ID to the end of status results
103. add title to the end of status results
104. close if statement
105. else message = GetNodeText from the utilities class with Response/Message as parameters
106. add message to the end of status results
107. close else statement
108. return status results as an array
109. **search company**
110. create a new link list of strings called company results and set it equal to an empty link list
111. create a new XML document called XML doc and set it equal to an empty XML document
112. load the XML document with response
113. code = GetNodeText from the utilities class with Response/ErrorCode, Response as parameters
114. if code = 0 then
115. create a new XML node list called node list and set it equal to XML doc select nodes method with Response/DVDS /DVD as a parameter
116. set title, and company to empty
117. use a for each loop to loop through the node list
118. for each node in the node list company equals the first item at the first index in the node. The index value is stored in the variable value1
119. title equals the second item at the second index in the node, the index value is stored in the variable value2
120. add company to the end of company results
121. add title to the end of company results
122. close if statement
123. else message = GetNodeText from the utilities class with Response/Message as parameters
124. add message to the end of company results
125. close else statement
126. return company results as an array
127. **search genre**
128. create a new link list of strings called genre results and set it equal to an empty link list
129. create a new XML document called XML doc and set it equal to an empty XML document
130. load the XML document with response
131. code = GetNodeText from the utilities class with Response/ErrorCode, Response as parameters
132. if code = 0 then
133. create a new XML node list called node list and set it equal to XML doc select nodes method with Response/DVDS /DVD as a parameter
134. set title, and description to empty
135. use a for each loop to loop through the node list
136. for each node in the node list description equals the first item at the first index in the node. The index value is stored in the variable value1
137. title equals the second item at the second index in the node, the index value is stored in the variable value2
138. add description to the end of genre results
139. add title to the end of genre results
140. close if statement
141. else message = GetNodeText from the utilities class with Response/Message as parameters
142. add message to the end of genre results
143. close else statement
144. return last genre results as an array
145. **search runtime**
146. create a new link list of strings called Time results and set it equal to an empty link list
147. create a new XML document called XML doc and set it equal to an empty XML document
148. load the XML document with response
149. code = GetNodeText from the utilities class with Response/ErrorCode, Response as parameters
150. if code = 0 then
151. create a new XML node list called node list and set it equal to XML doc select nodes method with Response/DVDS /DVD as a parameter
152. set title, and runtime to empty
153. use a for each loop to loop through the node list
154. for each node in the node list runtime equals the first item at the first index in the node. The index value is stored in the variable value1
155. title equals the second item at the second index in the node, the index value is stored in the variable value2
156. add runtime to the end of time results
157. add title to the end of time results
158. close if statement
159. else message = GetNodeText from the utilities class with Response/Message as parameters
160. add message to the end of time results
161. close else statement
162. return time results as an array
163. **search type**
164. create a new link list of strings called type results and set it equal to an empty link list
165. create a new XML document called XML doc and set it equal to an empty XML document
166. load the XML document with response
167. code = GetNodeText from the utilities class with Response/ErrorCode, Response as parameters
168. if code = 0 then
169. create a new XML node list called node list and set it equal to XML doc select nodes method with Response/DVDS /DVD as a parameter
170. set title, and type to empty
171. use a for each loop to loop through the node list
172. for each node in the node list type equals the first item at the first index in the node. The index value is stored in the variable value1
173. title equals the second item at the second index in the node, the index value is stored in the variable value2
174. add type to the end of type results
175. add title to the end of type results
176. close if statement
177. else message = GetNodeText from the utilities class with Response/Message as parameters
178. add message to the end of type results
179. close else statement
180. return type results as an array
181. **search title**
182. create a new link list of strings called title results and set it equal to an empty link list
183. create a new XML document called XML doc and set it equal to an empty XML document
184. load the XML document with response
185. code = GetNodeText from the utilities class with Response/ErrorCode, Response as parameters
186. if code = 0 then
187. create a new XML node list called node list and set it equal to XML doc select nodes method with Response/DVDS /DVD as a parameter
188. set title, and ID to empty
189. use a for each loop to loop through the node list
190. for each node in the node list ID equals the first item at the first index in the node. The index value is stored in the variable value1
191. title equals the second item at the second index in the node, the index value is stored in the variable value2
192. add ID to the end of title results
193. add title to the end of title results
194. close if statement
195. else message = GetNodeText from the utilities class with Response/Message as parameters
196. add message to the end of title results
197. close else statement
198. return title results as an array

**class diagram**

|  |
| --- |
| **Client** |
|  |
| Static Main (args: string []): void |

**Main method pseudocode**

1. **Main**
2. Call the main menu method of the menus class