Coverage for **library.py**: 94%

200 statements | 192 run | 8 missing | 0 excluded | 4 partial

« prev   ^ index   » next   coverage.py v7.6.8, created at 2024-12-01 17:34 +0100

```
 1  """Library management system, a database of books
 2  each Book, in  Library.txt:
 3  - Title (string)
 4  - Author (string)
 5  - Availability (string that is "true" or "false")
 6  - ISBN (string)
 7  - Edition (string)
 8
 9  Representation we will use after reading Library.txt
10  Book (ISBN as key in dict):
11  - title (string)
12  - author (string)
13  - numAvailable (int)
14  - numUnavailable (int)
15  - edition (string)
16  """
17
18  class Library:
19      def __init__(self, libraryFile):
20          self.libraryFile = libraryFile
21          self.books = self.readLibrary()
22
23      def readLibrary(self):
24          library = {}
25          with open(self.libraryFile, 'r') as file:
26              for line in file:
27                  bookTitle, bookAuthor, availability, isbn, edition = line.strip().split(',')
28                  bookIsAvailable = availability == "true"
29                  if isbn in library:
30                      if bookIsAvailable:                                          30 ↛ 31
31                          library[isbn]["numAvailable"] += 1
32                      else:
33                          library[isbn]["numUnavailable"] += 1
34                  else:
35                      library[isbn] = {
36                          "title": bookTitle,
37                          "author": bookAuthor,
38                          "numAvailable": 1 if bookIsAvailable else 0,
39                          "numUnavailable": 0 if bookIsAvailable else 1,
40                          "edition": edition
41                      }
42          return library
43
44      def updateLibrary(self):
45          with open(self.libraryFile, 'w') as file:
46              for isbn, book in self.books.items():
47                  title = book["title"]
48                  author = book["author"]
49                  edition = book["edition"]
50                  numAvail = book["numAvailable"]
51                  numUnavail = book["numUnavailable"]
```

```python
                bookAvailable = f"{title},{author},true,{isbn},{edition}\n"
                bookUnavailable = f"{title},{author},false,{isbn},{edition}\n"
                books = [bookAvailable] * numAvail + [bookUnavailable] * numUnavail
                file.writelines(books)

    def addBook(self, author, title, isbn, edition, amount):
        responses = []
        amount = int(amount)
        if isbn in self.books:
            self.books[isbn]['numAvailable'] += amount
        else:
            self.books[isbn] = {
                "title": title,
                "author": author,
                "numAvailable": amount,
                "numUnavailable": 0,
                "edition": edition
            }
        responses.append(f"Added new book {isbn}")
        return responses

    def removeBook(self, isbn, amount):
        responses = []
        amount = int(amount)
        if isbn in self.books:
            if amount <= self.books[isbn]['numAvailable']:
                self.books[isbn]['numAvailable'] -= amount
                responses.append(f'Removed {amount} of book {isbn}')
            else:
                responses.append('WARNING: Not enough books to remove')
        else:
            responses.append('WARNING: Book not found')
        return responses

    def borrowBook(self, isbn, amount):
        responses = []
        amount = int(amount)
        if isbn in self.books:
            if amount <= self.books[isbn]['numAvailable']:
                self.books[isbn]['numAvailable'] -= amount
                self.books[isbn]['numUnavailable'] += amount
                responses.append(f'{amount} of book {isbn} borrowed')
            else:
                responses.append('WARNING: No available books')
        else:
            responses.append("WARNING: No books found")
        return responses

    def returnBook(self, isbn, amount):
        responses = []
        amount = int(amount)
        if isbn in self.books:
            if amount <= self.books[isbn]['numUnavailable']:
                self.books[isbn]['numUnavailable'] -= amount
                self.books[isbn]['numAvailable'] += amount
                responses.append(f'{amount} of book {isbn} returned')
            else:
                responses.append('WARNING: Books already available')
        else:
            responses.append("WARNING: No books found")
        return responses

    def searchBook(self, author, title):
        responses = []
        foundBooks = []
        for isbn in self.books:
            book = self.books[isbn]
            bookTitle = book['title']
            bookAuthor = book['author']
            if title.lower() == bookTitle.lower() and author.lower() == bookAuthor.lower():
                edition = book["edition"]
                numAvail = book["numAvailable"]
                numUnavail = book["numUnavailable"]
                bookAvailable = f"{bookTitle}, {bookAuthor}, Available, {isbn}, {edition}"
                bookUnavailable = f"{bookTitle}, {bookAuthor}, Unavailable, {isbn}, {edition}
                books = [bookAvailable] * numAvail + [bookUnavailable] * numUnavail
                foundBooks += books
        if not foundBooks:
            responses.append("WARNING: No books found")
        else:
            responses = foundBooks
        return responses

    def searchBookTitle(self, title):
        responses = []
        foundBooks = []
        for isbn in self.books:
            book = self.books[isbn]
            bookTitle = book['title']
            numAvail = book["numAvailable"]

            if title.lower() in bookTitle.lower() and numAvail > 0:
                author = book['author']
                edition = book['edition']
                bookAvailable = f"{bookTitle}, {author}, Available, {isbn}, {edition}"
                books = [bookAvailable] * numAvail
                foundBooks += books
        if not foundBooks:
            responses.append("WARNING: No books found")
        else:
            responses = foundBooks
        return responses

    def searchBookAuthor(self, author):
        responses = []
        foundBooks = []
        for isbn in self.books:
            book = self.books[isbn]
            bookAuthor = book['author']
            numAvail = book['numAvailable']

            if author.lower() == bookAuthor.lower() and numAvail > 0:
                title = book['title']
                edition = book['edition']
```

```
166                 bookAvailable = f"{title}, {bookAuthor}, Available, {isbn}, {edition}"
167                 books = [bookAvailable] * numAvail
168                 foundBooks += books
169         if not foundBooks:
170             responses.append("WARNING: No books found")
171         else:
172             responses = foundBooks
173         return responses
174 
175     def searchBookISBN(self, isbn):
176         responses = []
177         foundBooks = []
178         if isbn in self.books:
179             book = self.books[isbn]
180             title = book['title']
181             author = book['author']
182             edition = book['edition']
183             numAvail = book['numAvailable']
184             bookAvailable = f"{title}, {author}, Available, {isbn}, {edition}"
185             books = [bookAvailable] * numAvail
186             foundBooks += books
187         if not foundBooks:
188             responses.append("WARNING: No books found")
189         else:
190             responses = foundBooks
191         return responses
192 
193     def listAvailableBooks(self):
194         responses = []
195         availableBooks = []
196         for isbn in self.books:
197             book = self.books[isbn]
198             numAvail = book['numAvailable']
199             if numAvail > 0:                                              199 ↛ 196
200                 title = book['title']
201                 author = book['author']
202                 edition = book['edition']
203                 bookAvailable = f"{title}, {author}, Available, {isbn}, {edition}"
204                 books = [bookAvailable] * numAvail
205                 availableBooks += books
206         if not availableBooks:
207             responses.append("WARNING: No books found")
208         else:
209             responses = availableBooks
210         return responses
211 
212     def listAvailableAuthors(self):
213         responses = []
214         availableAuthors = set()
215         for isbn in self.books:
216             book = self.books[isbn]
217             numAvail = book['numAvailable']
218             if numAvail > 0:                                              218 ↛ 215
219                 availableAuthors.add(book['author'])
220         if not availableAuthors:
221             responses.append("WARNING: No available authors found")
222         else:
```

```
223             responses = list(availableAuthors)
224         return responses
225 
226 def processCommands(library, inputFile):
227     with open(inputFile, 'r') as file:
228         for line in file:
229             command = line.strip()
230             responses = runCommand(library, command)
231             for response in responses:
232                 print(response)
233 
234 def runCommand(library, inputCommand):
235     command, *options = inputCommand.split('*')
236     if command == 'AddBook':
237         return library.addBook(options[0], options[1], options[2], options[3], options[4])
238     elif command == 'RemoveBook':
239         return library.removeBook(options[0], options[1])
240     elif command == 'BorrowBook':
241         return library.borrowBook(options[0], options[1])
242     elif command == 'ReturnBook':
243         return library.returnBook(options[0], options[1])
244     elif command == 'SearchBook':
245         return library.searchBook(options[0], options[1])
246     elif command == 'SearchBookTitle':
247         return library.searchBookTitle(options[0])
248     elif command == 'SearchBookAuthor':
249         return library.searchBookAuthor(options[0])
250     elif command == 'SearchBookISBN':
251         return library.searchBookISBN(options[0])
252     elif command == 'ListAvailableBooks':
253         return library.listAvailableBooks()
254     elif command == 'ListAvailableAuthors':                                    254 →/256
255         return library.listAvailableAuthors()
256     return []
257 
258 def main():
259     library = Library('Library.txt')
260     processCommands(library, 'input.txt')
261     library.updateLibrary()
262 
```

« prev ^ index » next coverage.py v7.6.8, created at 2024-12-01 17:34 +0100