1 /\*\*  
 2 \* The class Book creates a book object with the ability to change and display  
 3 \* the following attributes:  
 4 \* title, author, genre, edition, pages read, rating  
 5 \*   
 6 \* @author Stephanie Gremillion  
 7 \* @ version 17.0.2  
 8 \* @since 2022/06/24  
 9 \*/  
10   
11 public class Book {  
12   
13 // set variables  
14 private String title, author, genre, edition;  
15 private int pagesRead, rating;  
16   
17 // constructor  
18 public Book(String conTitle, String conAuthor, String conGenre,   
19 String conEdition, int conPages, int conRating) {  
20   
21 title = conTitle;  
22 author = conAuthor;  
23 genre = conGenre;  
24 edition = conEdition;  
25 pagesRead = conPages;  
26 rating = ((conRating \* 100) / 10);  
27 }  
28   
29 // setting the attributes  
30 public void setTitle(String titl) { title = titl; } // setting the title  
31 public void setAuthor(String auth) { author = auth; } // setting the author  
32 public void setGenre(String genr) { genre = genr; } // setting the genre  
33 public void setEdition(String editn) { edition = editn; } // setting the edition  
34 public void setPages(int pages) { pagesRead = pages; } // setting the pagesRead  
35 public void setRating(int rate) { rating = ((rate \* 100) / 10); } // setting the rating  
36   
37 // manipulating pages and rating  
38 public void changePages(int changePage) { pagesRead += changePage; } // changes the pages  
39 public void changeRating(int changeRate) { rating = ((changeRate \* 100) / 10); } // changes the rating  
40   
41 // getting the attributes  
42 public String getTitle() { return title; } // getting the title  
43 public String getAuthor() { return author; } // getting the author  
44 public String getGenre() { return genre; } // getting the genre  
45 public String getEdition() { return edition; } // getting the edition  
46 public int getPages() { return pagesRead; } // getting the pages  
47 public int getRating() { return rating; } // getting the rating  
48   
49 }

1 /\*\*  
 2 \* This tester BookTester reads and writes Book objects.  
 3 \* It has an interactive menu to manipulate Book objects.  
 4 \*   
 5 \* @author Stephanie Gremillion  
 6 \* @ version 17.0.2  
 7 \* @since 2022/06/24  
 8 \*/  
 9   
 10 import java.util.ArrayList;  
 11 import java.util.Scanner;  
 12   
 13 public class BookTester {  
 14   
 15 public static void main(String[] args) {  
 16 // declare variables  
 17 int mainMenuChoice = 0;  
 18 String title = "temp";  
 19 String author = "temp";  
 20 String genre = "temp";  
 21 String edition = "temp";  
 22 int pages = 0;  
 23 int rating = 0;  
 24 int editBook = 0; // whick book user wants to edit  
 25 Scanner input = new Scanner(System.in);  
 26 String temp = "temp"; // for determining if user entered number  
 27 int addSub = 0; // if user wants to add (0) or subtract (1) pages  
 28   
 29 // setting and initializing default books  
 30 ArrayList<Book> books = new ArrayList<Book>();  
 31 books.add(new Book("Madame Bovary", "Gustave Flaubert", "Fiction", "First", 52, 5));  
 32 books.add(new Book("The Great Gatsby", "F. Scott Fitzgerald", "Fiction", "First", 0, 0));  
 33 books.add(new Book("The Adventures of Huckleberry Finn", "Mark Twain", "Fiction", "First", 11, 5));  
 34 books.add(new Book("Hamlet", "William Shakespeare", "Ficition", "Third", 5, 3));  
 35 books.add(new Book("Moby Dick", "Herman Melville", "Fiction", "First", 0, 0));  
 36 books.add(new Book("The Odyssey", "Homer", "Fiction", "First", 42, 4));  
 37   
 38 // loop menu choices until exit  
 39 do { // run main menu  
 40 mainMenuChoice = mainMenu();  
 41   
 42 // run menu choice  
 43 System.out.println(); // empty line between menu and choice display  
 44 switch(mainMenuChoice) {  
 45 case 1:  
 46 for(int i = 0; i < books.size(); i++) {  
 47 System.out.println("Book " + (i+1) + ": ");  
 48 System.out.println("Title: " + books.get(i).getTitle());  
 49 System.out.println("Author: " + books.get(i).getAuthor());  
 50 System.out.println("Genre: " + books.get(i).getGenre());  
 51 System.out.println("Edition: " + books.get(i).getEdition());  
 52 System.out.println("Pages Read: " + books.get(i).getPages());  
 53 System.out.println("Rating: " + books.get(i).getRating() + "%");  
 54 System.out.println();  
 55 }  
 56 break;  
 57 case 2:  
 58 // user enters new book information  
 59 System.out.print("Enter Title: ");  
 60 title = input.nextLine();  
 61 System.out.print("Enter Author: ");  
 62 author = input.nextLine();  
 63 System.out.print("Enter Genre: ");  
 64 genre = input.nextLine();  
 65 System.out.print("Enter Edition: ");  
 66 edition = input.nextLine();  
 67   
 68 System.out.print("Enter Pages Read: ");  
 69 temp = input.nextLine();  
 70 while(temp.matches("[0-9]+") == false) { // confirming pages read input is number  
 71 System.out.println("Please enter a number for Pages Read: ");  
 72 temp = input.nextLine();  
 73 }  
 74 pages = Integer.parseInt(temp);  
 75   
 76 System.out.print("Enter Rating (0 - 10): ");  
 77 temp = input.nextLine();  
 78 while((temp.matches("[0-9]+") == false) || (Integer.parseInt(temp) < 0) || (Integer.parseInt(temp) > 10)) { // confirming rating input is number  
 79 System.out.println("Please enter a valid number for Rating: ");  
 80 temp = input.nextLine();  
 81 }  
 82 rating = Integer.parseInt(temp);  
 83 System.out.println(); // empty line  
 84   
 85 // adding new book to list  
 86 books.add(new Book(title, author, genre, edition, pages, rating));  
 87   
 88 // printing out book information for confirmation  
 89 System.out.println("New Book Added: ");  
 90 System.out.println("Title: " + books.get((books.size() - 1)).getTitle());  
 91 System.out.println("Author: " + books.get((books.size() - 1)).getAuthor());  
 92 System.out.println("Genre: " + books.get((books.size() - 1)).getGenre());  
 93 System.out.println("Edition: " + books.get((books.size() - 1)).getEdition());  
 94 System.out.println("Pages Read: " + books.get((books.size() - 1)).getPages());  
 95 System.out.println("Rating: " + books.get((books.size() - 1)).getRating() + "%");  
 96 System.out.println(); // empty line  
 97 break;  
 98 case 3:  
 99 System.out.println("Which number book do you want to edit?");  
100 for(int x = 0; x < books.size(); x++) {  
101 System.out.println("Book " + (x + 1) + ": " + books.get(x).getTitle());  
102 }  
103 temp = input.nextLine();  
104 while((temp.matches("[0-9]+") == false) || (Integer.parseInt(temp) <= 0) || (Integer.parseInt(temp) > books.size())) { // confirming number book input is number and valid  
105 System.out.println("Please enter a valid number: ");  
106 temp = input.nextLine();  
107 }  
108 editBook = Integer.parseInt(temp) - 1;  
109 System.out.println("\nWhat would you like to edit?");  
110 System.out.println("1. Title: " + books.get(editBook).getTitle());  
111 System.out.println("2. Author: " + books.get(editBook).getAuthor());  
112 System.out.println("3. Genre: " + books.get(editBook).getGenre());  
113 System.out.println("4. Edition: " + books.get(editBook).getEdition());  
114 System.out.println("5. Pages Read: " + books.get(editBook).getPages());  
115 System.out.println("6. Rating: " + books.get(editBook).getRating() + "%");  
116 temp = input.nextLine();  
117 while((temp.matches("[0-9]+") == false) || (Integer.parseInt(temp) < 1) || (Integer.parseInt(temp) > 6)) { // confirming selection is number  
118 System.out.println("Please enter a valid number: ");  
119 temp = input.nextLine();  
120 }  
121 System.out.println(); // empty line  
122 switch(Integer.parseInt(temp)) {  
123 case 1:  
124 System.out.println("Enter a new Title: ");  
125 temp = input.nextLine();  
126 books.get(editBook).setTitle(temp);  
127 System.out.println("New Title: " + books.get(editBook).getTitle());  
128 System.out.println(); // empty line  
129 break;  
130 case 2:  
131 System.out.println("Enter a new Author: ");  
132 temp = input.nextLine();  
133 books.get(editBook).setAuthor(temp);  
134 System.out.println("New Author: " + books.get(editBook).getAuthor());  
135 System.out.println(); // empty line  
136 break;  
137 case 3:  
138 System.out.println("Enter a new Genre: ");  
139 temp = input.nextLine();  
140 books.get(editBook).setGenre(temp);  
141 System.out.println("New Genre: " + books.get(editBook).getGenre());  
142 System.out.println(); // empty line  
143 break;  
144 case 4:  
145 System.out.println("Enter a new Edition: ");  
146 edition = input.nextLine();  
147 books.get(editBook).setEdition(edition);  
148 System.out.println("New Edition: " + books.get(editBook).getEdition());  
149 System.out.println(); // empty line  
150 break;  
151 case 5:  
152 System.out.println("What do you want to do?");  
153 System.out.println("1. Add pages");  
154 System.out.println("2. Subtract pages");  
155 temp = input.nextLine();  
156 while((temp.matches("[0-9]+") == false) || (Integer.parseInt(temp) > 2) || (Integer.parseInt(temp) < 1)) {  
157 System.out.println("Please enter a valid selection.");  
158 temp = input.nextLine();  
159 }  
160 addSub = Integer.parseInt(temp);  
161 if(addSub == 1) {  
162 System.out.println("How many pages would you like to add?");  
163 temp = input.nextLine();  
164 while(temp.matches("[0-9]+") == false) {  
165 System.out.println("Please enter a valid number.");  
166 temp = input.nextLine();  
167 }  
168 books.get(editBook).changePages(Integer.parseInt(temp));  
169 System.out.println("New amount of Pages Read: " + books.get(editBook).getPages());  
170 System.out.println(); // empty line  
171 }  
172 else {  
173 System.out.println("How many pages would you like to subtract?");  
174 temp = input.nextLine();  
175 while(temp.matches("[0-9]+") == false) {  
176 System.out.println("Please enter a valid number.");  
177 temp = input.nextLine();  
178 }  
179 while((books.get(editBook).getPages() - Integer.parseInt(temp)) < 0) {  
180 System.out.println(books.get(editBook).getTitle() + " has " + books.get(editBook).getPages() + " pages read.");  
181 System.out.println("Please enter a number less than " + books.get(editBook).getPages() + ".");  
182 temp = input.nextLine();  
183 }  
184 books.get(editBook).changePages((Integer.parseInt(temp)) \* -1);  
185 System.out.println("New amount of Pages Read: " + books.get(editBook).getPages());  
186 System.out.println(); // empty line   
187 }  
188 break;  
189 case 6:  
190 System.out.println("Enter a new Rating (0-10): ");  
191 temp = input.nextLine();  
192 while((temp.matches("[0-9]+") == false) || (Integer.parseInt(temp) < 0) || (Integer.parseInt(temp) > 10)) { // confirming rating is number  
193 System.out.println("Please enter a valid number (0-10).");  
194 temp = input.nextLine();  
195 }  
196 books.get(editBook).changeRating(Integer.parseInt(temp));  
197 System.out.println("New Rating: " + books.get(editBook).getRating() + "%");  
198 System.out.println(); // empty line  
199 break;  
200 default:  
201 break;  
202 }  
203 break;  
204 default:  
205 break;  
206 }  
207 } while(mainMenuChoice != 4); // loop ends when user chooses 4 (exit)  
208 }  
209   
210 public static int mainMenu() {  
211 // declare variables  
212 Scanner input = new Scanner(System.in);  
213 int choice;  
214   
215 // show main menu options  
216 System.out.println("Please choose an option: ");  
217 System.out.println("1. Show Books");  
218 System.out.println("2. Add Book");  
219 System.out.println("3. Edit Book");  
220 System.out.println("4. Exit");  
221   
222 // accept selection and return result to main  
223 choice = Integer.parseInt(input.nextLine());  
224 return choice;  
225 }  
226 }