This project simulates a low scale library system, the program manages basic library functions eg. borrowing, adding and removing books.
It has three main parts: Book, Library, and Borrower.

- **Book:**
  - Represents a book and holds information about its title, author, and genre.
  - When we create a new book, we provide these details using the constructor.
  - We can get the title, author, and genre of a book using separate methods.
  - The toString() method helps display the book information in a user-friendly way.

- **Library:**
  - Represents the library itself, which can hold a collection of books with a specific capacity (the maximum number of books it can hold).
  - There is a genreCount variable, which is like a dictionary that keeps track of how many books are there for each genre.
    - We have methods to:
    - countBooks(): Know the total number of books in the library.
    - isStockFull(): Check if the library is full and cannot take more books.
    - addBook(Book book): Add a book to the library collection and update the genre count.
    - updateGenreCount(String genre, int increment): A private method to change the count of books for a specific genre.
    - removeBook(Book book): Remove a book from the library and update the genre count accordingly.
    - displayGenreCount(): Display the count of books for each genre in the library.

- **Borrower:**
  - Represents a person who can borrow books from the library.
  - It has a collection of books (collection) that the borrower borrows from the library.
    - We have methods to:
    - borrowBook(Book book, Library library): Borrow a book from the library and add it to the borrower's collection.
    - displayCollection(): Display the collection of books the borrower has borrowed.

- **BookGenerator:**
  - A utility class to create random books.
  - It has a method called generateRandomBook() that randomly picks a title, author, and genre from predefined lists and creates a new Book with those details.

- **BorrowerApp:**
  - The main class of the program where everything starts.
  - It creates a Library with a capacity of 5 (maximum 5 books).
  - It uses the BookGenerator to create 5 random books and adds them to the library.
  - The library's genre count is then displayed.
  - A Borrower is created, and they borrow two books from the library.
  - The borrower's collection is displayed.
  - The library's genre count is shown again, updated after the books were borrowed.

In summary, this code simulates a simple library system where books can be added, removed, and borrowed by a borrower. It shows how classes can interact with each other, how objects store data, and how utility classes can generate random data. When you run the program, you'll see the library's status and how the borrower interacts with it. It's a neat example of using Java classes, objects, and collections to build a functional program.