### **Tổng quan project**

**Bạn đã bao giờ mượn sách ở thư viện chưa? Khi bạn mượn một cuốn sách, người thủ thư sẽ phải sử dụng một phần mềm để tìm những cuốn sách đó. Họ cũng sử dụng phần mềm này để nhập một cuốn sách mới vào hệ thống hoặc loại bỏ những cuốn sách đã cũ một cách dễ dàng. Trong project này, bạn sẽ thực hiện một phần mềm nhỏ để có thể giúp người thủ thư làm được những việc nêu trên.**

#### **Tại sao tôi cần project này?**

**Đây là một phần trong hệ thống quản lý thư viện để nâng cao tính hiệu quả. Hệ thống sẽ giúp các thủ thư lưu trữ, tìm kiếm, xóa và cập nhật thông tin của các cuốn sách trong thư viện. Không những thế hệ thống còn lưu thông tin những lượt mượn sách, cung cấp cái nhìn tổng quan về việc sử dụng thư viện. Học sinh cũng có thể mượn sách qua hệ thống này. Theo dõi và quản lý danh sách số lượng lớn các cuốn sách trong thư viện nhờ sử dụng linkedlist (danh sách liên kết) sẽ giúp các thao tác xóa, chèn và rấ nhiều thao tác khác nhanh hơn nhiều so với các mảng truyền thống.**

#### **Tôi sẽ học được gì từ project này?**

* + **Biết cách sử dụng dữ liệu kiểu enum trong java**
  + **Hiểu cách làm việc của LinkedList**
  + **Biết một số thao tác cơ bản với LinkedList**
  + **Biết cách kết hợp LinkedList và đối tượng để giải quyết một vấn đề nhất định – Hệ thống quản lý thư viện**
  + **Biết cách sắp xếp một project bằng nhiều lớp khác nhau**
  + **Hiểu được tầm quan trọng của LinkedList**
  + **Hiểu được lợi ích của việc sử dụng LinkedList so với mảng**

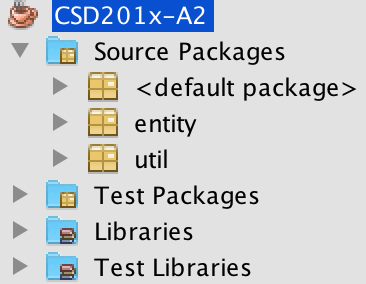
**Bắt đầu:**

1. Khi kiểm tra cấu trúc của project đã được cho, bạn sẽ thấy có một số gói được cho sẵn. Bạn có thể tùy ý sắp xếp lại hoặc tạo một project mới của riêng mình với nhiều lớp hoặc giao diện hơn nếu cần.

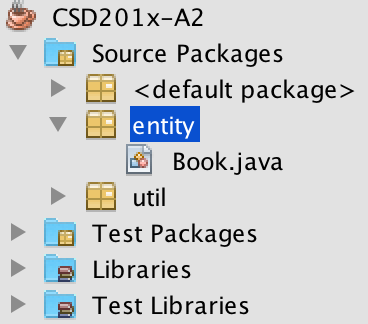
**- entity**: Có chứa các đoạn code có thông tin của đối tượng như: mã sách, tên, số lượng, giá và thông tin cho mượn.

**- util:** chứa đoạn code cho cấu trúc dữ liệu sử dụng trong bài (linked list). Lưu ý rằng chúng ta không sử dụng những linked list có sẵn trong java mà tự tạo cấu trúc riêng để lưu trữ dữ liệu cho các đối tượng.

**- Các lớp khác:** chứa các đoạn code cho hàm để thêm sách mới, tìm kiếm theo mã số sách và xuất thông tin của tất cả các cuốn sách …

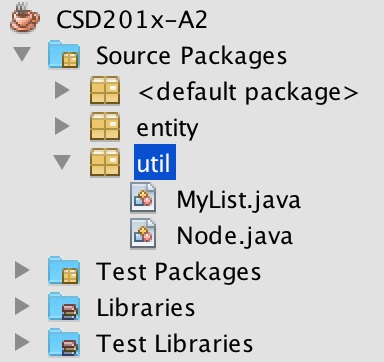


**2. Gói entity**



2.1 Thiết kế và lập trình một lớp Book, trong đó có chứa các thông tin của một cuốn sách như mã số, tiêu đề, số lượng, tình trạng mượn và giá. Lớp này đã được cho sẵn.

**3. Gói util**: chứa các lớp cấu trúc dữ liệu được sử dụng trong bài, ví dụ như LinkedList



**3.1 Lớp Node.java**: chứa thông tin của một Node (nút), bao gồm dữ liệu của nút đó và điểm tham chiếu tới nút tiếp theo. Lớp này được cho sẵn và sử dụng cho LinkedList, Stack and Queue, do đó chúng ta sẽ làm cho nó thành một nút chung.

**3.2 Lớp MyList.java**: chứa đoạn code cho việc sử dụng danh sách và một số thao tác trên danh sách này, dữ liệu trừu tượng (ADT – Abstract Data Type) trong danh sách liên kết là một loại (hoặc một lớp) các đối tượng mà hành vi được xác định bởi một chuỗi giá trị hoặc một chuỗi thao tác như các bảng dưới đây:

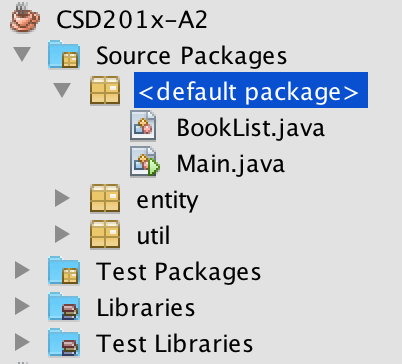
Tìm hiểu về ADT [tại đây](https://www.geeksforgeeks.org/abstract-data-types/)

|  |
| --- |
| **MyList** |
| Node<Book> head, tail |
| +MyList()  +isEmpty():Boolean  +addLast(b:Book):void  +addFirst(b:Book):void  +traversal():void  +size():int  +getNode():Node<Book>  +addAfter(b:Book,k:int):void  +deleteAt(k:int):void  +search(bCode:String):void |

Cách sử dụng ADT này xem [tại đây](https://www.cs.cmu.edu/~adamchik/15-121/lectures/Linked%20Lists/linked%20lists.html).

**4. Gói chính:** chứa các lớp có các thao tác chúng ta cần như thêm thông tin của sách/ người mượn/ lịch sử cho mượn và tìm kiếm thông tin của sách/ người mượn/ thông tin cho mượn …

**5. Lớp BookList.java:** chứa các phương thức/thao tác cho danh sách các cuốn sách như: thêm sách mới, tìm kiếm sách, xóa, hiển thị thông tin sách. Hoàn thành lớp này bằng cách lập trình các thao tác được liệt kê trong biểu đồ các lớp.



|  |
| --- |
| **BookList** |
| MyList: books |
| +BookList()  +getBook(): Book  +addLast():void  +list():void  +search():void  +addFirst():void  +addAfter():void  +deleteAt():void |

*Bạn có thể tùy ý thêm các phương thức/hàm và lớp nếu cần.*