

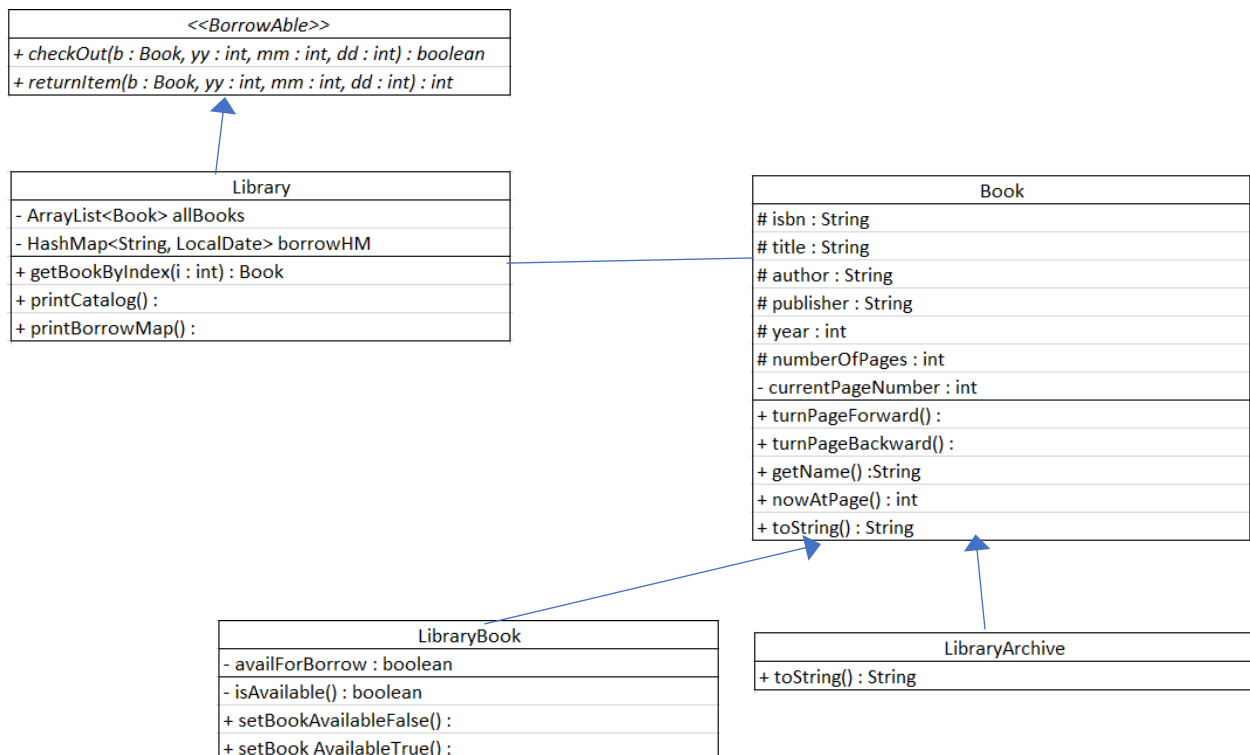
กำหนดให้ห้องสมุดมีทรัพยากร 2 ชนิดคือ หนังสือ (LibraryBook) ซึ่งให้ยืมได้และเอกสารหลักฐาน (LibraryArchive) ซึ่งให้ยืม
ออกไปไม่ได้

รายชื่อหนังสือจะโหลดเข้าโปรแกรมจากจากไฟล์ ClassicBookList.txt และ ArchiveList.txt ไว้ใน allBooks (ArrayList) ข้อมูล
การยืม เก็บไว้ใน borrowHM(HashMap) โดยเก็บเพียงชื่อหนังสือ และ วันที่ยืม (การคืนให้ลบออกจากตารางนี้) อนึ่ง HashMap
คือการใช้ hash(key) เพื่อที่จะทราบว่า value อยู่ที่ไหน (ชื่อหนังสือเป็นคีย์ว่าเก็บวันที่ยืมไว้ที่ไหน) นอกจากนี้ ให้ถือว่าหนังสือใน
ห้องสมุด 1 title มีเพียง copy เดียว ยังไม่ต้องกังวลเรื่องหนังสือมีหลาย copy

การจะสามารถให้บริการยืมได้ ต้อง implement checkOut() และ returnItem() ตามที่ BorrowAble กำหนดไว้ ในความเป็น
จริงการยืม/คืน ใช้เวลาจริงเป็นข้อมูล แต่เพื่อการจำลองเหตุการณ์ ให้ระบุ ปี เดือน วัน เป็นพารามิเตอร์สำหรับการยืม/คืน โดย

- checkOut() จะตอบค่าเป็น true หากเป็น LibraryBook (ไม่ใช่ LibraryArchive) และ availForBorrow ของหนังสือนั้นมีค่า
true และ checkOut() จะ set availForBorrow เป็น false และ ใช้ชื่อหนังสือ เป็น key การเก็บวันที่ยืมหนังสือไว้ใน
borrowHM นั้นเป็นเหตุผลที่ไม่ออกแบบ Borrowable ไว้กับหนังสือ เพราะขอบเขตของปัญหานี้เราต้องการให้ Library เป็น
ผู้จัดการเรื่องวันที่ยืม/วันที่คืน

- returnItem() จะลบรายการออกจาก borrowHM และ set availForBorrow ให้เป็น true โดยตอบเป็นจำนวนวันที่ยืมไป



การนับจำนวนวันใช้ `int d = (int)LocalDateObj1.until(LocalDateObj2, ChronoUnit.DAYS);`

การอ่านข้อมูลที่ละแถวใช้ `Path inFile = Paths.get(".", "ClassicBookList.txt");`

`Charset charset = Charset.forName("UTF-8");`

`try (BufferedReader s = Files.newBufferedReader(inFile, charset)) {`

`while ((line = s.readLine()) != null) { /* your code */ } s.close(); } catch () { }`

`printCatalog ()` ใช้พิมพ์รายละเอียดของหนังสือ โดยที่หากเป็น Archive ให้พิมพ์ว่า Archive Resource : ชื่อเอกสาร

`printBorrowMap()` ใช้พิมพ์ว่ามีหนังสือเล่มใดบ้างที่ยืมไป และยืมไปวันที่เท่าไร

ทำเสร็จแล้วลองพิจารณาดูว่า หากไม่เขียนแบบ OOP จะยากขึ้น-ง่ายขึ้น อย่างไร

หมายเหตุ

pair programming คือเทคนิคพัฒนาโปรแกรมที่มี
คุณภาพ โดยให้คนหนึ่งเป็นคน code อีกคนเป็นคนคอย
observe

code ที่ได้จากการใช้ pair programming จะอ่าน
ง่าย เพราะมีอีกคนคอยตรวจว่าความหมายของ
code ที่กำลังทำอยู่ตรงกับที่ออกแบบไว้หรือไม่ และ
มีความผิดพลาดน้อย การสลับกันเขียนจะช่วยเพิ่ม
ทักษะการพัฒนาฝีมือการเขียนโปรแกรมให้มี
มาตรฐานขององค์กร

Pair programming



Pair programming is an agile software development technique in which two programmers work together at one workstation. One, the driver, writes code while the other, the observer or navigator, reviews each line of code as it is typed in. The two programmers switch roles frequently. [Wikipedia](#)