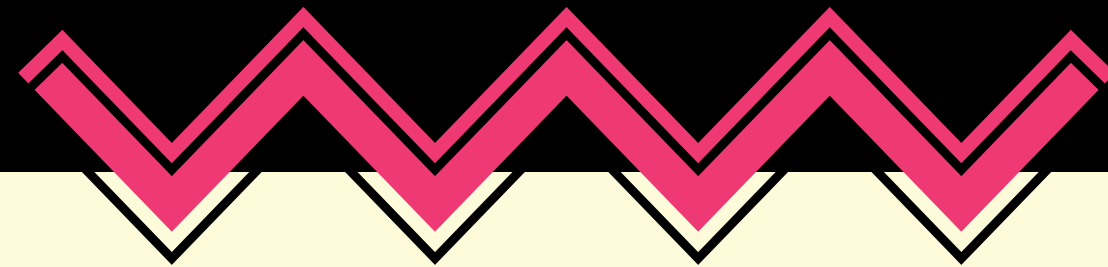


# LIBRARY MANAGEMENT SYSTEM

Start

# LIBRARY MANAGEMENT SYSTEM



- ผู้พัฒนา: กลุ่ม เอาฉันทออกไป
- 67162110374-2 สหณัฐ ติ 16
- 65162110011-2 พนรกร ก้อง 2
- คำอธิบายสั้น ๆ:
- “ระบบจัดการห้องสมุดที่รองรับการยืม – คืนสิ่งของและจัดเก็บข้อมูลสมาชิก โดยใช้หลักการ OOP”

```
class Book extends LibraryItem {
    private author: string;

    constructor(title: string, itemId: string, author: string) {
        super(title, itemId);
        this.author = author;
    }
    getDetails(): string {
        return `Book: ${this.getTitle()} by ${this.author} (ID: ${this.itemId})`;
    }
}
```

```
class Magazine extends LibraryItem {
    private issueDate: string;

    constructor(title: string, itemId: string, issueDate: string) {
        super(title, itemId);
        this.issueDate = issueDate;
    }
    getDetails(): string {
        return `Magazine: ${this.getTitle()} (ID: ${this.itemId}), Issue Date: ${this.issueDate}`;
    }
}
```

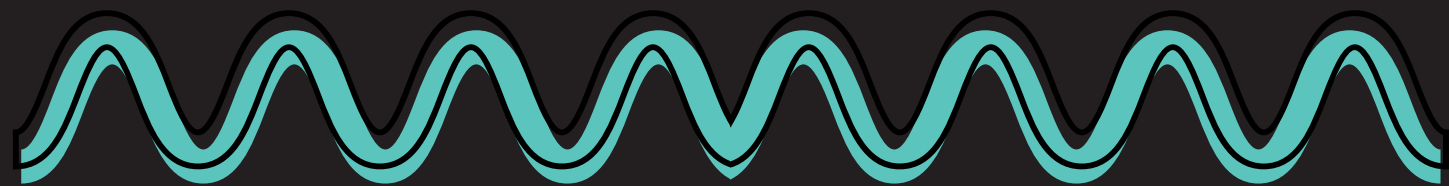
```
class DVD extends LibraryItem {
    private issueDate: string;
    private director: string;

    constructor(title: string, itemId: string, issueDate: string, director: string) {
        super(title, itemId);
        this.issueDate = issueDate;
        this.director = director;
    }
    getDetails(): string {
        return `DVD: ${this.getTitle()} (ID: ${this.itemId}), Issue Date: ${this.issueDate}, Director: ${this.director}`;
    }
}
```

```
class car extends LibraryItem {
    private issueDate: string
    private manufacturer: string;
    constructor(title: string, itemId: string, issueDate: string, manufacturer: string) {
        super(title, itemId);
        this.issueDate = issueDate;
        this.manufacturer = manufacturer;
    }
    getDetails(): string {
        return `car: ${this.getTitle()} (ID: ${this.itemId}), Issue Date: ${this.issueDate}, Manufacturer: ${this.manufacturer}`;
    }
}
```

```
class AudioBook extends LibraryItem {
    private issueDate: string;
    private author: string;
    constructor(title: string, itemId: string, issueDate: string, author: string) {
        super(title, itemId);
        this.issueDate = issueDate;
        this.author = author;
    }
    getDetails(): string {
        return `AudioBook: ${this.getTitle()} (ID: ${this.itemId}), Issue Date: ${this.issueDate}, Author: ${this.author}`;
    }
}
```

# FEATURE



## 1. เพิ่มและจัดการ Library Items

- Book, Magazine, DVD, AudioBook และ Car (เป็นของแปลกใหม่)

## 2. เพิ่มและจัดการ Library Members

- เก็บข้อมูลชื่อและรหัสสมาชิก

## 3. ฟังก์ชันหลัก

- การยืมสิ่งของ → ตรวจสอบสถานะว่าพร้อมใช้งาน
- การคืนสิ่งของ → คืนเข้าระบบและเปลี่ยนสถานะเป็น available
- การตรวจสอบรายการสิ่งของที่ยืมอยู่
- การแสดง Library Summary (ภาพรวมสิ่งของ + รายชื่อสมาชิก)

# OOP CONCEPTS

- Encapsulation (การห่อหุ้ม)
  - > ใช้ private และ protected เพื่อควบคุมการเข้าถึงข้อมูล เช่น \_title, \_available
- Inheritance (การสืบทอด)
  - > Book, Magazine, DVD, AudioBook, Car → สืบทอดจาก LibraryItem
- Abstraction (การทำให้เป็นนามธรรม)
  - > LibraryItem เป็น abstract class → บังคับให้ subclass ต้อง implement getDetails()
- Interface (สัญญา)
  - > Borrowable กำหนด method ที่เกี่ยวกับการยืม-คืน เช่น borrow(), returnItem()
- Aggregation (ความสัมพันธ์แบบมีส่วนประกอบ)
  - > Library มี Members และ Item
  - > LibraryMember มี borrowedItems

# CLASS DIAGRAM

1. แสดง โครงสร้างทั้งหมดของระบบ

2. จุดเด่นใน Diagram:

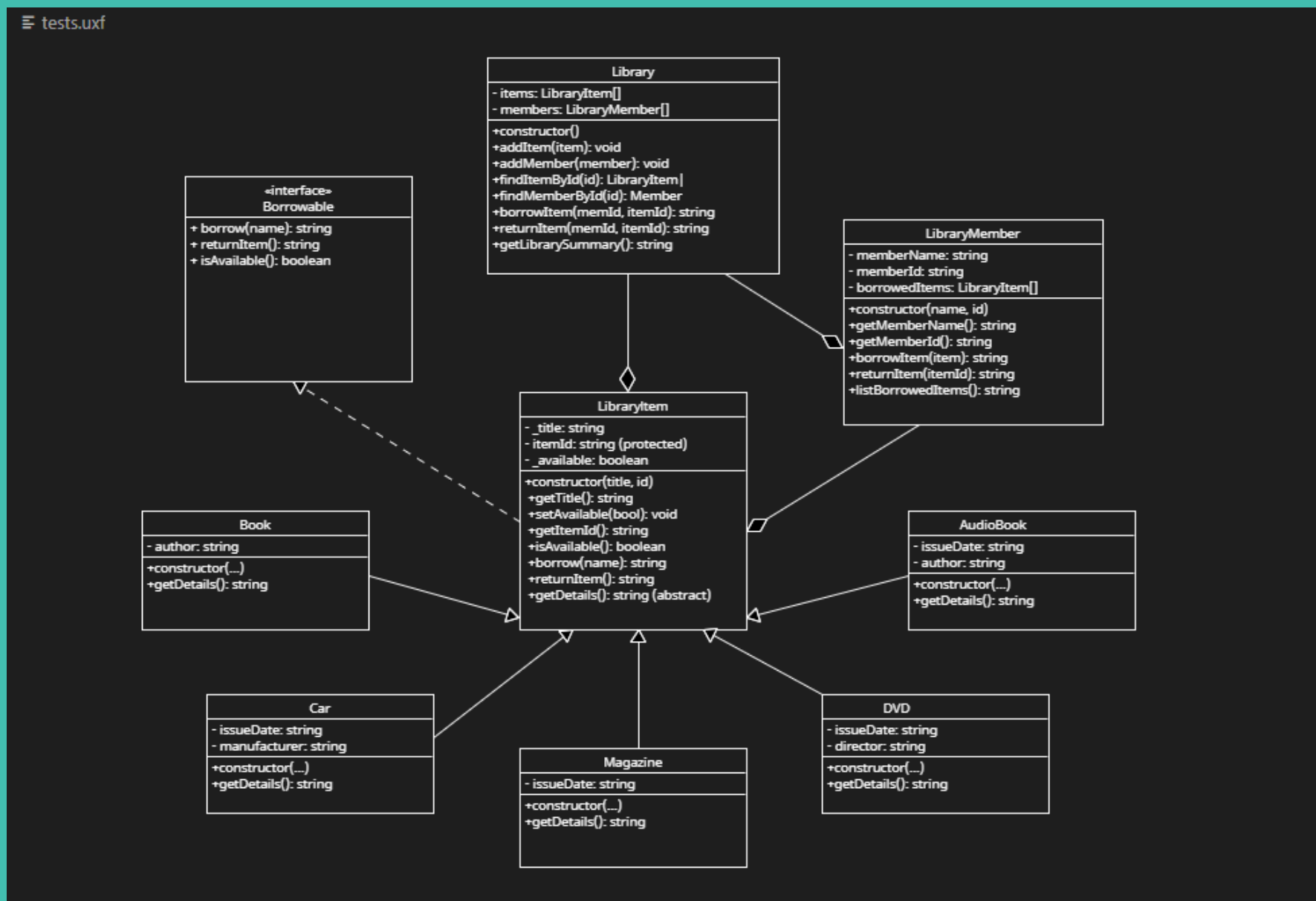
Borrowable <|.. LibraryItem

LibraryItem <|-- Book/Magazine/DVD/AudioBook/Car

Library <>-- LibraryItem

Library <>-- LibraryMember

LibraryMember <>-- LibraryItem (borrowedItems)



## อธิบาย CLASS หลัก (LIBRARYITEM + BORROWABLE)

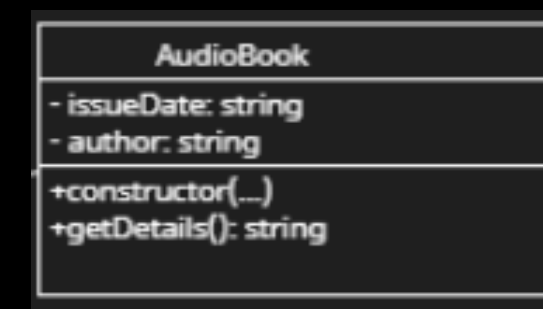
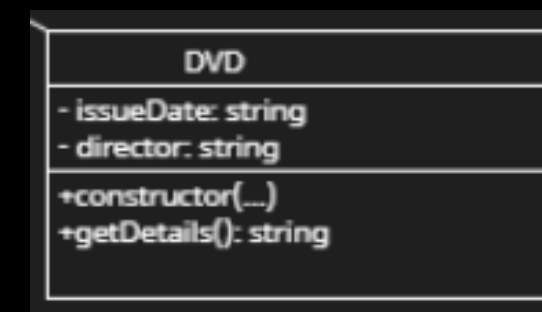
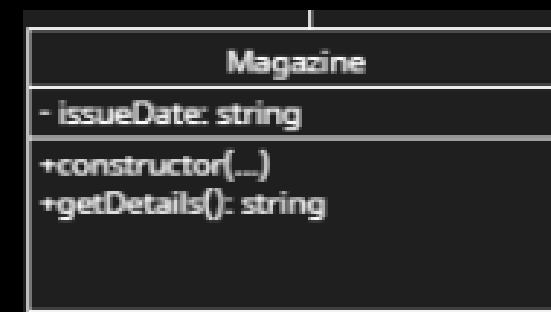
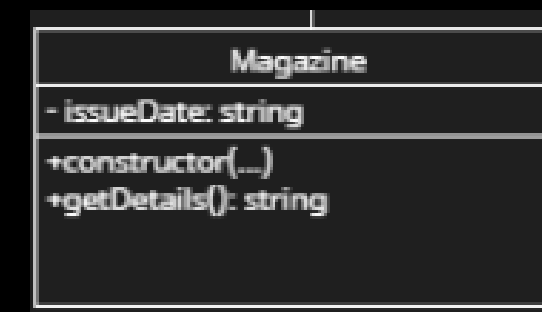
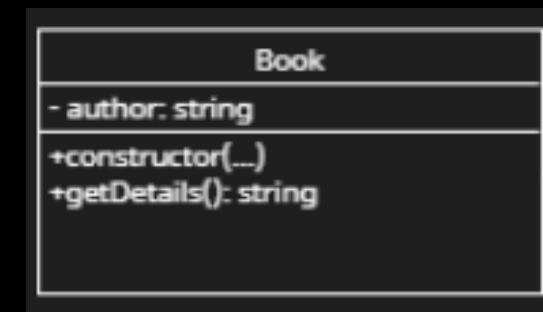
- LibraryItem (abstract class)  
Attribute: \_title, itemId, \_available  
Method:  
    borrow(memberName) → ยืมสิ่งของ  
    returnItem() → คืนสิ่งของ  
    isAvailable() → ตรวจสอบสถานะ  
    getDetails() → ให้ subclass override
- Borrowable (interface)  
    borrow(), returnItem(), isAvailable()

```
✓ interface Borrowable {  
    borrow(memberName: string): string;  
    returnItem(): string;  
    isAvailable(): boolean;  
}  
  
✓ abstract class LibraryItem implements Borrowable {  
    private _title: string;  
    protected itemId: string;  
    private _available: boolean;
```

# CLASS ย่อย (ITEMS)

- Book: เพิ่ม **author**
- Magazine: เพิ่ม **issueDate**
- DVD: เพิ่ม **issueDate, director**
- AudioBook: เพิ่ม **issueDate, author**
- Car: เพิ่ม **issueDate, manufacturer**  
(เป็น **Feature** แปลกใหม่ที่ระบบห้องสมุด  
ทั่วไปไม่มี)

ทุก **class** จะ **override** **getDetails()** → แสดง  
รายละเอียดในรูปแบบเฉพาะของตน



# LIBRARY MEMBER

```
class LibraryMember {  
  
    private memberName: string;  
    private memberId: string;  
    private borrowedItems: LibraryItem[];  
  
    constructor(memberName: string, memberId: string) {  
        this.borrowedItems = [];  
        this.memberName = memberName;  
        this.memberId = memberId;  
    }  
  
    getMemberName(): string {  
        return this.memberName;  
    }  
  
    getMemberId(): string {  
        return this.memberId;  
    }  
  
    borrowItem(item: LibraryItem): string {  
        if (item.isAvailable()) {  
            const borrowMessage = item.borrow(this.memberName);  
            this.borrowedItems.push(item);  
            return borrowMessage;  
        } else {  
            return "Item not available";  
        }  
    }  
}
```

- Attribute: memberName, memberId, borrowedItems[]
- Method:
  - borrowItem(item) → ยืมสิ่งของ ถ้าว่างจะเพิ่มเข้า borrowedItems
  - returnItem(itemId) → คืนสิ่งของออกจาก borrowedItems
  - listBorrowedItems() → แสดงรายการสิ่งของที่สมาชิกคนนั้นยืมอยู่
- จุดเด่น: รองรับการจัดการ หลายรายการ ต่อสมาชิก



# การขยายในอนาคต



- เพิ่มประเภท Item ใหม่ได้ง่ายเพราะมี Inheritance
- ขยายระบบให้ซับซ้อนขึ้น เช่น:
  - Reservation System (จองสิ่งของ)
  - Fine Management (ค่าปรับกรณีคืนช้า)
  - Online Access (เว็บ/แอปต่อ Database จริง)
- การออกแบบด้วย OOP → ทำให้โค้ด ยืดหยุ่น และง่ายต่อการ บำรุงรักษาในอนาคต