

OOP_TEST

67162110315-5 สมภพโชค ไชษฐ์ 14

67162110559-4 อัศภาคย์ ภูผา 30

FEATURE

การจัดการสมาชิกห้องสมุด

- สมาชิกสามารถ ยืมหนังสือ
- มีการเก็บประวัติการยืม

ระบบบันทึกการยืม

- บันทึกว่าใครยืมหนังสือเล่มไหน
- รองรับการเรียกดูหนังสือที่ยืมไปแล้ว

ระบบจัดการห้องสมุด

- เพิ่มหนังสือ
- เพิ่ม DVD
- เพิ่มสมาชิก
- แสดงรายการหนังสือ (ชื่อ, ID, ผู้แต่ง)
- แสดง รายละเอียดเชิงลึก (ชื่อคนยืม + หนังสือ + ผู้แต่ง + ID)

ILIBRARYITEM (INTERFACE)

```
export interface ILibraryItem {  
  getId(): number;  
  getTitle(): string;  
  getInfo(): string;  
}
```

ILibraryItem (Interface)

ใช้กำหนดมาตรฐานกลางของ “สิ่งของในห้องสมุด”

- บังคับว่าทุก Item (เช่น Book, DVD) ต้องมี ID, Title และฟังก์ชัน getInfo()
- ทำให้การจัดการสื่อทุกชนิดในห้องสมุดทำได้ในรูปแบบเดียวกัน

LIBRARYITEM (ABSTRACT CLASS)

```
export abstract class LibraryItem implements ILibraryItem {  
  private id: number;  
  private title: string;  
  
  Windsurf: Refactor | Explain | Generate JSDoc | ✕  
  constructor(id: number, title: string) {  
    this.id = id;  
    this.title = title;  
  }  
  
  Tabnine | Edit | Test | Explain | Document | Windsurf: Refactor | Explain | Generate JSDoc | ✕  
  public getId(): number {  
    return this.id;  
  }  
  
  Tabnine | Edit | Test | Explain | Document | Windsurf: Refactor | Explain | Generate JSDoc | ✕  
  public getTitle(): string {  
    return this.title;  
  }  
  
  Tabnine | Edit | Test | Explain | Document  
  public abstract getInfo(): string;  
}
```

คลาสแม่สำหรับสื่อทุกชนิด

- เก็บ ID และ Title
- มีเมธอดพื้นฐาน เช่น
 getId(), getTitle()
- มี getInfo() เป็น abstract →
 ให้คลาสลูกเขียนเองตาม
 ลักษณะของตัวเอง

BOOK (CLASS)

```
Windsurf: Refactor | Explain
export class Book extends LibraryItem implements IBook {
  private author: string;

Windsurf: Refactor | Explain | Generate JSDoc | ✕
  constructor(id: number, title: string, author: string) {
    super(id, title);
    this.author = author;
  }

Tabnine | Edit | Test | Explain | Document | Windsurf: Refactor | Explain | Generate JSDoc | ✕
  public getAuthor(): string {
    return this.author;
  }

Tabnine | Edit | Test | Explain | Document | Windsurf: Refactor | Explain | Generate JSDoc | ✕
  public getInfo(): string {
    return `Book [ID: ${this.getId()}, Title: ${this.getTitle()}, Author: ${this.author}]`;
  }
}
```

แทนวัตถุ “หนังสือ” ในห้องสมุด

- เก็บ ชื่อเรื่อง (Title) และ ผู้แต่ง (Author)
- ใช้สืบทอดจาก LibraryItem
- มี getInfo() แสดงรายละเอียดหนังสือ

DVD (CLASS)

```
export class DVD extends LibraryItem implements ILibraryItem {
  private duration: number;

  constructor(id: number, title: string, duration: number) {
    super(id, title);
    this.duration = duration;
  }

  public getDuration(): number {
    return this.duration;
  }

  public getInfo(): string {
    return `DVD [ID: ${this.getId()}, Title: ${this.getTitle()}, Duration: ${this.duration} mins]`;
  }
}
```

แทนวัตถุ “DVD” ในห้องสมุด

- เก็บ ชื่อเรื่อง (Title) และ ระยะเวลา (Duration)
- ใช้สืบทอดจาก LibraryItem
- มี getInfo() แสดงรายละเอียด DVD

LOANRECORD (CLASS)

```
export class LoanRecord implements ILoanRecord {  
  private book: Book;  
  
  Windsurf: Refactor | Explain | Generate JSDoc | ✕  
  constructor(book: Book) {  
    this.book = book;  
  }  
  
  Windsurf: Refactor | Explain | Generate JSDoc | ✕  
  public getBook(): Book {  
    return this.book;  
  }  
}
```

ใช้เก็บประวัติการยืมหนังสือ

- บันทึกว่า “สมาชิกคนหนึ่ง ยืมหนังสือเล่มไหน”
- ทำให้สามารถตรวจสอบย้อนหลังได้ว่าใครยืมอะไร

LIBRARYMEMBER (CLASS)

```
export class LibraryMember implements ILibraryMember {
  private name: string;
  private loans: LoanRecord[] = [];

  Windsurf: Refactor | Explain | Generate JSDoc | X
  constructor(name: string) {
    this.name = name;
  }

  Tabnine | Edit | Test | Explain | Document | Windsurf: Refactor | Explain | Generate JSDoc | X
  public getName(): string {
    return this.name;
  }

  Tabnine | Edit | Test | Explain | Document | Windsurf: Refactor | Explain | Generate JSDoc | X
  public borrowBook(book: Book): void {
    this.loans.push(new LoanRecord(book));
  }

  Tabnine | Edit | Test | Explain | Document | Windsurf: Refactor | Explain | Generate JSDoc | X
  public getLoans(): LoanRecord[] {
    return this.loans;
  }

  Tabnine | Edit | Test | Explain | Document | Windsurf: Refactor | Explain | Generate JSDoc | X
  public getInfo(): string {
    const borrowed = this.loans.map(l => l.getBook().getTitle()).join(", ") || "None";
    return `Member: ${this.name}, Borrowed Books: [${borrowed}]`;
  }
}
```

แทน “สมาชิกห้องสมุด”

- เก็บ ชื่อสมาชิก
- สามารถ ยืมหนังสือ ได้ → ข้อมูลถูกเก็บเป็น LoanRecord
- แสดงข้อมูลว่า “สมาชิกคนนี้ ยืมหนังสืออะไรบ้าง”

LIBRARY (CLASS)

```
private books: Book[] = [];  
private dvds: DVD[] = [];  
private members: LibraryMember[] = [];  
  
Tabnine | Edit | Test | Explain | Document | Windsurf: Refactor | Explain | Generate JSDoc | X  
public addBook(book: Book): void {  
    this.books.push(book);  
}  
  
Tabnine | Edit | Test | Explain | Document | Windsurf: Refactor | Explain | Generate JSDoc | X  
public addDVD(dvd: DVD): void {  
    this.dvds.push(dvd);  
}  
  
Tabnine | Edit | Test | Explain | Document | Windsurf: Refactor | Explain | Generate JSDoc | X  
public addMember(member: LibraryMember): void {  
    this.members.push(member);  
}  
  
// (1) ชื่อหนังสือ ไอดี และผู้แต่ง  
Tabnine | Edit | Test | Explain | Document | Windsurf: Refactor | Explain | X  
public showBooks(): void {  
    console.log("📖 Books in Library:");  
    this.books.forEach(book => console.log(book.getInfo()));  
}  
  
// (2) คนยืม + รายชื่อหนังสือที่ยืม  
Tabnine | Edit | Test | Explain | Document | Windsurf: Refactor | Explain | X  
public showMembersAndLoans(): void {  
    console.log("👤 Members and their loans:");  
    this.members.forEach(member => console.log(member.getInfo()));  
}  
  
// (3) รายละเอียดคืม  
Tabnine | Edit | Test | Explain | Document | Windsurf: Refactor | Explain | X  
public showDetailedLoans(): void {  
    console.log("📋 Detailed Loans:");  
    this.members.forEach(member => {  
        member.getLoans().forEach(loan => {  
            const book = loan.getBook();  
            console.log(`Member: ${member.getName()} | Book: ${book.getTitle()} | Author: ${book.getAuthor()} | ID: ${book.getId()}`);  
        });  
    });  
}
```

คลาสหลักที่ดูแลการทำงานของห้องสมุดทั้งหมด

- เก็บ รายการหนังสือ, DVD และสมาชิก
- เพิ่มหนังสือ/ดีวีดี/สมาชิกได้
- แสดงข้อมูลได้หลายรูปแบบ เช่น
 - รายชื่อหนังสือทั้งหมด
 - รายชื่อสมาชิก + หนังสือที่ยืม
 - รายละเอียดเต็มของการยืม

INDEX.TS (MAIN PROGRAM)

```
import { Library } from "./Library";
import { Book } from "./Book";
import { DVD } from "./DVD";
import { LibraryMember } from "./LibraryMember";

const library = new Library();

const b1 = new Book(1, "Harry Potter", "J.K. Rowling");
const b2 = new Book(2, "The Hobbit", "J.R.R. Tolkien");
const b3 = new Book(3, "1984", "George Orwell");
library.addBook(b1);
library.addBook(b2);
library.addBook(b3);

library.addDVD(new DVD(101, "Inception", 148));

const m1 = new LibraryMember("Alice");
const m2 = new LibraryMember("Bob");
library.addMember(m1);
library.addMember(m2);

m1.borrowBook(b1);
m1.borrowBook(b3);
m2.borrowBook(b2);

console.log("\n=== 1. ชื่อหนังสือ ไอดีหนังสือ และชื่อผู้แต่ง ===");
library.showBooks();




console.log("\n=== 2. ชื่อคนยืมหนังสือและหนังสือที่ยืมไป ===");
library.showMembersAndLoans();

console.log("\n=== 3. รายละเอียดชื่อคนยืมหนังสือ + ชื่อหนังสือ + ผู้แต่ง + ไอดี ===");
library.showDetailedLoans();
```

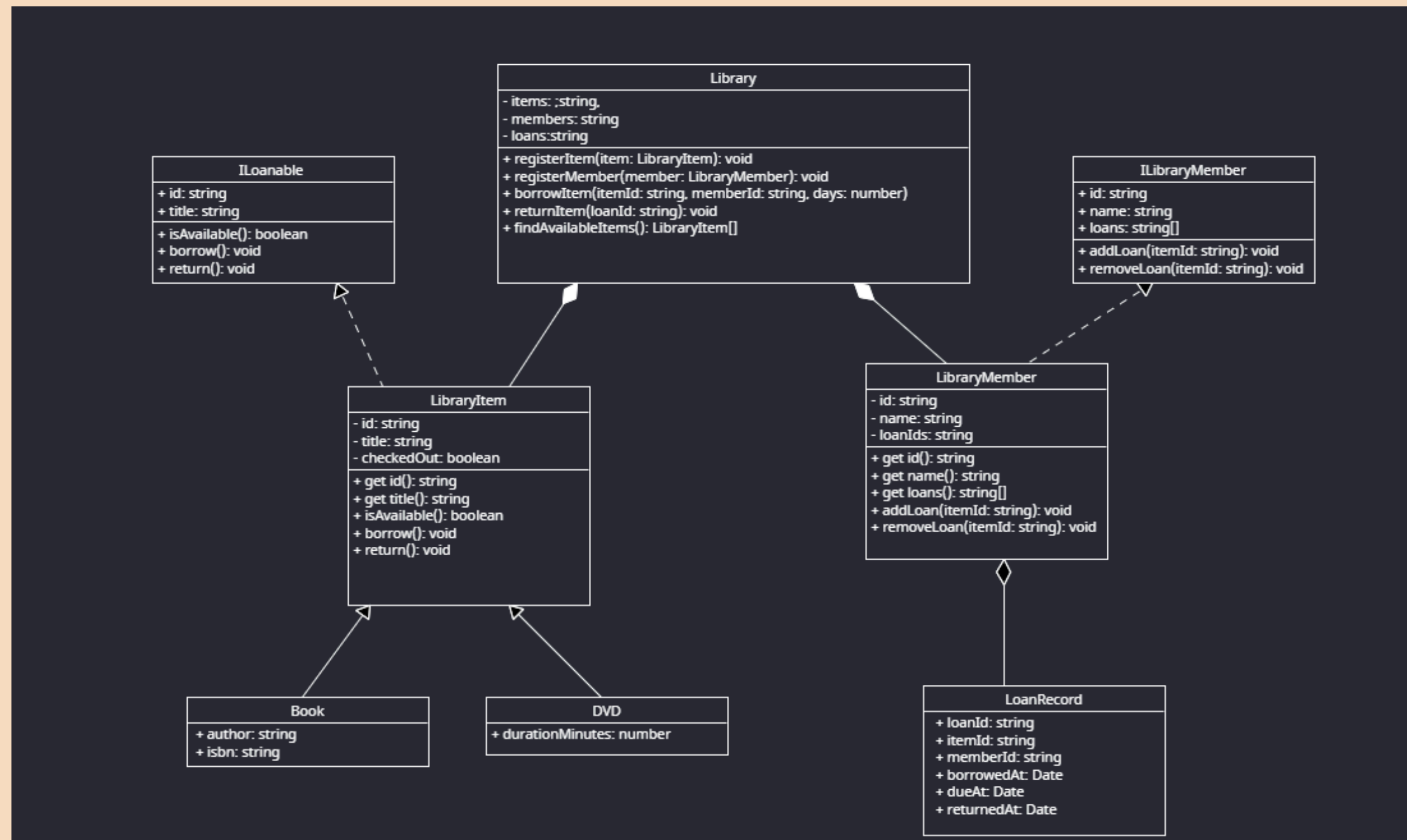
ไฟล์หลักสำหรับ “ทดสอบระบบ”

- สร้างห้องสมุด
- เพิ่มหนังสือและ DVD
- สมัครสมาชิก
- ให้สมาชิกยืมหนังสือ
- แสดงผลลัพธ์ (รายชื่อหนังสือ, รายชื่อผู้ยืม, รายละเอียดการยืม)

แสดงผลลัพธ์

```
\n=== 1. ชื่อหนังสือ ไอดีหนังสือ และ ชื่อผู้แต่ง ===  
 Books in Library:  
Book [ID: 1, Title: Harry Potter, Author: J.K. Rowling]  
Book [ID: 2, Title: The Hobbit, Author: J.R.R. Tolkien]  
Book [ID: 3, Title: 1984, Author: George Orwell]  
\n=== 2. ชื่อคนยืมหนังสือและหนังสือที่ยืมไป ===  
 Members and their loans:  
Member: Alice, Borrowed Books: [Harry Potter, 1984]  
Member: Bob, Borrowed Books: [The Hobbit]  
\n=== 3. รายละเอียดชื่อคนยืมหนังสือ + ชื่อหนังสือ + ผู้แต่ง + ไอดี ===  
 Detailed Loans:  
Member: Alice | Book: Harry Potter | Author: J.K. Rowling | ID: 1  
Member: Alice | Book: 1984 | Author: George Orwell | ID: 3  
Member: Bob | Book: The Hobbit | Author: J.R.R. Tolkien | ID: 2
```


UML DIAGRAM



THANK YOU

Samira Hadid

