JAVA PROJECTS2

2.Library Management System

- Description: Develop a library management system with features like book cataloging, borrower management, due date tracking, and fine calculation.

1.Class Book

```
public class Book {
    private String title;
    private String author;
    private String isbn;
    private boolean isBorrowed;

    public Book(String title, String author, String isbn) {
        this.title = title;
        this.author = author;
        this.isbn = isbn;
        this.isBorrowed = false;
    }

    public String getTitle() {
        return title;
    }

    public String getAuthor() {
        return author;
    }

    public String getIsbn() {
        return isbn;
    }
```

```
    public boolean isBorrowed() {
        return isBorrowed;
    }

    public void borrow() {
        isBorrowed = true;
    }

    public void returnBook() {
        isBorrowed = false;
    }

    @Override
    public String toString() {
        return title + " by " + author + " (ISBN: " + isbn + ")";
    }
}
```

2.Class Borrower

```
import java.util.HashMap;
import java.util.Map;

public class Borrower {
    private String name;
    private Map<Book, Integer> borrowedBooks; // Book and number of days overdue

    public Borrower(String name) {
        this.name = name;
        this.borrowedBooks = new HashMap<>();
```

```
}

public String getName() {
    return name;
}

public void borrowBook(Book book) {
    borrowedBooks.put(book, 0); // 0 days overdue at the beginning
    book.borrow();
}

public void returnBook(Book book) {
    borrowedBooks.remove(book);
    book.returnBook();
}

public void addOverdueDays(Book book, int days) {
    if (borrowedBooks.containsKey(book)) {
        borrowedBooks.put(book, borrowedBooks.get(book) + days);
    }
}

public int calculateFine() {
    int fine = 0;
    for (int days : borrowedBooks.values()) {
        fine += days * 2; // Assuming fine is $2 per day overdue
    }
    return fine;
}
```

```
    public Map<Book, Integer> getBorrowedBooks() {
        return borrowedBooks;
    }
}
```

3.Library Class

```
import java.util.ArrayList;
import java.util.List;

public class Library {
    private List<Book> books;
    private List<Borrower> borrowers;

    public Library() {
        books = new ArrayList<>();
        borrowers = new ArrayList<>();
    }

    public void addBook(Book book) {
        books.add(book);
    }

    public void addBorrower(Borrower borrower) {
        borrowers.add(borrower);
    }

    public Book findBookByIsbn(String isbn) {
        for (Book book : books) {
            if (book.getIsbn().equals(isbn)) {
```

```
                return book;
            }
        }
        return null;
    }

    public Borrower findBorrowerByName(String name) {
        for (Borrower borrower : borrowers) {
            if (borrower.getName().equals(name)) {
                return borrower;
            }
        }
        return null;
    }

    public List<Book> getBooks() {
        return books;
    }

    public List<Borrower> getBorrowers() {
        return borrowers;
    }
}
```

4.Main Method/Class

```
public class Main {
    public static void main(String[] args) {
        // Create library
        Library library = new Library();
```

```
// Add books to library
Book book1 = new Book("1984", "George Orwell", "123456789");
Book book2 = new Book("To Kill a Mockingbird", "Harper Lee", "987654321");
library.addBook(book1);
library.addBook(book2);

// Add borrowers to library
Borrower borrower1 = new Borrower("Alice");
Borrower borrower2 = new Borrower("Bob");
library.addBorrower(borrower1);
library.addBorrower(borrower2);

// Borrower borrows a book
borrower1.borrowBook(book1);

// Display borrowed books
System.out.println("Borrowed books by " + borrower1.getName() + ":");
for (Book book : borrower1.getBorrowedBooks().keySet()) {
    System.out.println(book);
}

// Return a book
borrower1.returnBook(book1);

// Display available books
System.out.println("\nAvailable books in the library:");
for (Book book : library.getBooks()) {
    if (!book.isBorrowed()) {
        System.out.println(book);
```

```
        }
    }

    // Add overdue days and calculate fine
    borrower1.borrowBook(book2);
    borrower1.addOverdueDays(book2, 5);
    System.out.println("\nFine for " + borrower1.getName() + ": $" +
borrower1.calculateFine());
  }
}
```

OUTPUT

```
Borrowed books by Alice:
1984 by George Orwell (ISBN: 123456789)

Available books in the library:
1984 by George Orwell (ISBN: 123456789)
To Kill a Mockingbird by Harper Lee (ISBN: 987654321)

Fine for Alice: $10
```