Java - MultiThread Exercise

Rodrigo Castellanos Rodríguez - A01643147

Desarrollo e implantación de sistemas de software (Gpo 102)

13 de abril, 2025

Este sistema simula una biblioteca concurrente donde múltiples usuarios interactúan simultáneamente con la colección de libros. La arquitectura consta de tres códigos principales:

- **BookThreads:** Representa libros individuales con estado de préstamo y fechas de vencimiento
- **LibraryThreads:** Gestiona la colección completa de libros y usuarios, coordinando los préstamos
- **PatronThreads:** Simula el comportamiento de usuarios que piden y devuelven libros aleatoriamente

El sistema utiliza sincronización para garantizar la seguridad en entornos multihilo.

Clase BookThreads

- checkOut(int daysToDue): Marca el libro como prestado y establece fecha de vencimiento
    - Sincronizado para evitar préstamos simultáneos
    - Lanza excepción si el libro ya está prestado
- returnBook(): Registra la devolución del libro
    - Sincronizado para garantizar actualización segura del estado
- isOverdue(): Verifica si el préstamo está vencido
    - Comparación thread-safe de fechas

Clase LibraryThreads

- checkOutBook(PatronThreads patron, BookThreads book, int daysToDue): Coordina el préstamo de un libro
    - Verifica disponibilidad y registra la transacción
- returnBook(PatronThreads patron, BookThreads book): Maneja la devolución de libros
    - Actualiza estados
    - Sincronizado para evitar conflictos con préstamos simultáneos
- calculateFine(PatronThreads patron): Calcula multas por retraso
    - Thread-safe mediante sincronización

Clase PatronThreads

- run(): Punto de entrada del hilo
    - Ejecuta acciones aleatorias en bucle
    - Maneja interrupciones adecuadamente
- borrowRandomBook(): Selecciona y pide un libro disponible aleatorio
    - Acceso sincronizado a la lista de libros
- returnRandomBook(): Devuelve un libro aleatorio de su posesión
    - Notifica si genera multa por retraso
- performRandomAction(): Lógica para decidir la siguiente acción

Requisitos Previos

- Java JDK 8 o superior
- Entorno de desarrollo como VS Code

Para ejecutar la aplicación primero los códigos se deben compilar con el comando:

- javac *.java

Después ya puedes correr la aplicación con:

- java Main.java