Video

<https://drive.google.com/file/d/1YQ_nYjABPpk3p1vThyH6fUvPyvtEm8l8/view?usp=sharing>

import java.time.LocalDate;  
import java.util.Scanner;  
  
public class LibraryU1Single {  
 private static final int *MAX\_LIBROS* = 100;  
 private static final int *SUCURSALES* = 3;  
  
 private static final Catalogo *catalogo* = new Catalogo(*MAX\_LIBROS*, *SUCURSALES*);  
 private static final PrestamoList *prestamosActivos* = new PrestamoList();  
 private static final Historial *historial* = new Historial();  
 private static final Scanner *sc* = new Scanner(System.*in*);  
  
  
 static class Libro {  
 int codigo; String titulo; String autor; int stockTotal; boolean activo;  
 Libro(int codigo, String titulo, String autor, int stockTotal) {  
 this.codigo = codigo; this.titulo = titulo; this.autor = autor; this.stockTotal = stockTotal; this.activo = true;  
 }  
 public String toString() {  
 return String.*format*("[%d] '%s' - %s | stock=%d | %s", codigo, titulo, autor, stockTotal, (activo?"ACTIVO":"BAJA"));  
 }  
 }  
  
 static class Catalogo {  
 Libro[] libros; int[][] disp; int usados; final int sucursales;  
 Catalogo(int max, int suc) { libros = new Libro[max]; disp = new int[max][suc]; usados = 0; sucursales = suc;  
 }  
 boolean altaLibro(Libro l, int[] stockPorSucursal) {  
 if (usados >= libros.length) return false;  
 if (buscarIndicePorCodigo(l.codigo) != -1) return false;  
 libros[usados] = l;  
 for (int c = 0; c < sucursales; c++) disp[usados][c] = (c < stockPorSucursal.length ? Math.*max*(0, stockPorSucursal[c]) : 0);  
 usados++; return true;  
 }  
 boolean bajaLogica(int codigo) { int idx = buscarIndicePorCodigo(codigo); if (idx==-1 || libros[idx]==null) return false; libros[idx].activo=false; return true; }  
 int buscarIndicePorCodigo(int codigo) {  
 for (int i = 0; i < usados; i++) if (libros[i]!=null && libros[i].codigo==codigo) return i; return -1;  
 }  
 int[] buscarIndicesPorTitulo(String t) { // O(n)  
 int[] tmp = new int[usados]; int k=0; String needle=t.toLowerCase();  
 for (int i=0;i<usados;i++) if (libros[i]!=null && libros[i].titulo.toLowerCase().contains(needle)) tmp[k++]=i;  
 int[] res = new int[k]; for (int i=0;i<k;i++) res[i]=tmp[i]; return res;  
 }  
 boolean actualizarStock(int idx, int nuevoTotal, int[] nuevaDisp) {  
 if (idx<0 || idx>=usados || libros[idx]==null) return false; libros[idx].stockTotal=Math.*max*(0,nuevoTotal);  
 for (int c=0;c<sucursales;c++) disp[idx][c]=(c<nuevaDisp.length?Math.*max*(0,nuevaDisp[c]):0); return true;  
 }  
 boolean tomarPrestamo(int idx, int suc) {  
 if (idx<0 || idx>=usados) return false; Libro l=libros[idx]; if (l==null||!l.activo||l.stockTotal<=0) return false;  
 if (suc<0 || suc>=sucursales || disp[idx][suc]<=0) return false; l.stockTotal--; disp[idx][suc]--; return true;  
 }  
 boolean devolver(int idx, int suc) {  
 if (idx<0 || idx>=usados || libros[idx]==null) return false; if (suc<0 || suc>=sucursales) return false;  
 libros[idx].stockTotal++; disp[idx][suc]++; return true;  
 }  
 void listarCatalogo() {  
 System.*out*.println("\n=== CATÁLOGO ===");  
 for (int i=0;i<usados;i++) if (libros[i]!=null) {  
 System.*out*.println(i+": "+libros[i]);  
 System.*out*.print(" Disp por sucursal: ");  
 for (int s=0;s<sucursales;s++) System.*out*.print("S"+s+"="+disp[i][s]+(s==sucursales-1?"":", "));  
 System.*out*.println();  
 }  
 }  
 }  
  
 enum TipoOp { *ALTA*, *BAJA*, *PRESTAMO*, *DEVOLUCION* }  
  
 static class Prestamo {  
 int codigoLibro; String usuario; LocalDate fecha; boolean devuelto; Prestamo next;  
 Prestamo(int c,String u,LocalDate f){codigoLibro=c;usuario=u;fecha=f;}  
 public String toString(){return String.*format*("Prestamo{libro=%d, usuario=%s, fecha=%s, devuelto=%s}",codigoLibro,usuario,fecha,devuelto);}  
 }  
 static class PrestamoList {  
 Prestamo head, tail;  
 void insertarAlFinal(Prestamo p){ if(head==null){head=tail=p;} else {tail.next=p; tail=p;} }  
 boolean devolverYRemover(String u,int c){  
 Prestamo prev=null,cur=head;  
 while(cur!=null){  
 if(!cur.devuelto&&cur.usuario.equals(u)&&cur.codigoLibro==c){  
 cur.devuelto=true;  
 if(prev==null) head=cur.next; else prev.next=cur.next;  
 if(cur==tail) tail=prev;  
 return true;  
 }  
 prev=cur; cur=cur.next;  
 }  
 return false;  
 }  
 void imprimir(){  
 System.*out*.println("\n=== PRÉSTAMOS ACTIVOS ===");  
 if(head==null){System.*out*.println("(lista vacía)"); return;}  
 for(Prestamo cur=head;cur!=null;cur=cur.next) System.*out*.println(cur);  
 }  
 }  
  
 static class HistNode { TipoOp tipo; String detalle; LocalDate fecha; HistNode prev,next;  
 HistNode(TipoOp t,String d){tipo=t;detalle=d;fecha= LocalDate.*now*();}  
 public String toString(){ return String.*format*("[%s] %s (%s)",fecha,tipo,detalle); }  
 }  
 static class Historial {  
 HistNode head, tail;  
 void agregar(TipoOp t,String d){  
 HistNode n=new HistNode(t,d);  
 if(head==null){head=tail=n;} else {tail.next=n; n.prev=tail; tail=n;}  
 }  
 void listarAdelante(){  
 System.*out*.println("\n=== HISTORIAL → (adelante) ===");  
 if(head==null){System.*out*.println("(vacío)"); return;}  
 for(HistNode c=head;c!=null;c=c.next) System.*out*.println(c);  
 }  
 void listarAtras(){  
 System.*out*.println("\n=== HISTORIAL ← (atrás) ===");  
 if(tail==null){System.*out*.println("(vacío)"); return;}  
 for(HistNode c=tail;c!=null;c=c.prev) System.*out*.println(c);  
 }  
 }  
  
  
 private static int leerEntero(String msg){  
 while(true){  
 System.*out*.print(msg);  
 try{ return Integer.*parseInt*(*sc*.nextLine().trim()); }  
 catch(Exception e){ System.*out*.println("→ Entrada inválida. Intenta de nuevo."); }  
 }  
 }  
 private static String leerLinea(String msg){ System.*out*.print(msg); return *sc*.nextLine(); }  
  
  
 private static void cargarLibro(){  
 System.*out*.println("\n-- Alta de libro --");  
 int codigo=*leerEntero*("Código: "); String titulo=*leerLinea*("Título: "); String autor=*leerLinea*("Autor: "); int total=*leerEntero*("Stock total: ");  
 int[] stockSuc=new int[*SUCURSALES*]; for(int s=0;s<*SUCURSALES*;s++) stockSuc[s]=*leerEntero*("Stock S"+s+": ");  
 boolean ok=*catalogo*.altaLibro(new Libro(codigo,titulo,autor,total),stockSuc);  
 if(ok){ *historial*.agregar(TipoOp.*ALTA*,"Libro "+codigo+" – "+titulo); System.*out*.println("Alta correcta."); }  
 else System.*out*.println("No se pudo dar de alta (lleno o duplicado).");  
 }  
 private static void eliminarLibro(){  
 System.*out*.println("\n-- Baja lógica --");  
 int codigo=*leerEntero*("Código: "); boolean ok=*catalogo*.bajaLogica(codigo);  
 if(ok){ *historial*.agregar(TipoOp.*BAJA*,"Libro "+codigo); System.*out*.println("Baja aplicada."); }  
 else System.*out*.println("Código no encontrado.");  
 }  
 private static void buscarLibroPorTitulo(){  
 System.*out*.println("\n-- Buscar por título --");  
 String q=*leerLinea*("Título (o parte): "); int[] idxs=*catalogo*.buscarIndicesPorTitulo(q);  
 if(idxs.length==0){ System.*out*.println("Sin coincidencias."); return;}  
 for(int idx:idxs) System.*out*.println(idx+": "+*catalogo*.libros[idx]);  
 }  
 private static void prestarLibro(){  
 System.*out*.println("\n-- Prestar libro --");  
 int codigo=*leerEntero*("Código: "); String usuario=*leerLinea*("Usuario: "); int suc=*leerEntero*("Sucursal (0.."+(*SUCURSALES*-1)+"): ");  
 int idx=*catalogo*.buscarIndicePorCodigo(codigo);  
 if(idx==-1){System.*out*.println("No existe ese código."); return;}  
 if(!*catalogo*.libros[idx].activo){System.*out*.println("Libro en baja lógica."); return;}  
 if(*catalogo*.libros[idx].stockTotal<=0){System.*out*.println("Sin stock."); return;}  
 if(!*catalogo*.tomarPrestamo(idx,suc)){System.*out*.println("No disponible en esa sucursal."); return;}  
 *prestamosActivos*.insertarAlFinal(new Prestamo(codigo,usuario,LocalDate.*now*()));  
 *historial*.agregar(TipoOp.*PRESTAMO*,"Usuario "+usuario+" tomó libro "+codigo+" (S"+suc+")");  
 System.*out*.println("Préstamo registrado.");  
 }  
 private static void devolverLibro(){  
 System.*out*.println("\n-- Devolver libro --");  
 int codigo=*leerEntero*("Código: "); String usuario=*leerLinea*("Usuario: "); int suc=*leerEntero*("Sucursal (0.."+(*SUCURSALES*-1)+"): ");  
 int idx=*catalogo*.buscarIndicePorCodigo(codigo);  
 if(idx==-1){System.*out*.println("No existe ese código."); return;}  
 boolean okPrest=*prestamosActivos*.devolverYRemover(usuario,codigo);  
 if(!okPrest){ System.*out*.println("No hay préstamo activo para ese usuario."); return;}  
 if(!*catalogo*.devolver(idx,suc)){ System.*out*.println("No fue posible actualizar stock."); return;}  
 *historial*.agregar(TipoOp.*DEVOLUCION*,"Usuario "+usuario+" devolvió libro "+codigo+" (S"+suc+")");  
 System.*out*.println("Devolución registrada.");  
 }  
 private static void listarPrestamos(){ *prestamosActivos*.imprimir(); }  
 private static void listarHistorialAdelante(){ *historial*.listarAdelante(); }  
 private static void listarHistorialAtras(){ *historial*.listarAtras(); }  
 private static void actualizarStock(){  
 System.*out*.println("\n-- Actualizar stock --");  
 int codigo=*leerEntero*("Código: "); int idx=*catalogo*.buscarIndicePorCodigo(codigo);  
 if(idx==-1){System.*out*.println("No encontrado."); return;}  
 int nuevoTotal=*leerEntero*("Nuevo stock total: "); int[] ns=new int[*SUCURSALES*];  
 for(int s=0;s<*SUCURSALES*;s++) ns[s]=*leerEntero*("Nueva disp S"+s+": ");  
 boolean ok=*catalogo*.actualizarStock(idx,nuevoTotal,ns);  
 System.*out*.println(ok?"Actualización correcta.":"Falló actualización.");  
 }  
  
 private static int menu(){  
 System.*out*.println("\n============================");  
 System.*out*.println(" Biblioteca U1 – Menú ");  
 System.*out*.println("============================");  
 System.*out*.println("1) Alta de libro");  
 System.*out*.println("2) Baja lógica de libro");  
 System.*out*.println("3) Buscar libro por título");  
 System.*out*.println("4) Listar catálogo");  
 System.*out*.println("5) Prestar libro");  
 System.*out*.println("6) Devolver libro");  
 System.*out*.println("7) Listar préstamos activos");  
 System.*out*.println("8) Historial → (adelante)");  
 System.*out*.println("9) Historial ← (atrás)");  
 System.*out*.println("10) Actualizar stock");  
 System.*out*.println("0) Salir");  
 return *leerEntero*("Elige opción: ");  
 }  
  
 private static void seedDemo(){  
 *catalogo*.altaLibro(new Libro(101,"Programación Estructurada","A. Turing",3), new int[]{2,1,0});  
 *catalogo*.altaLibro(new Libro(102,"Estructuras de Datos","N. Wirth",2), new int[]{1,1,0});  
 *catalogo*.altaLibro(new Libro(103,"Algoritmos","R. Sedgewick",1), new int[]{1,0,0});  
 }  
  
 public static void main(String[] args){  
 *seedDemo*();  
 boolean salir=false;  
 while(!salir){  
 switch(*menu*()){  
 case 1 -> *cargarLibro*();  
 case 2 -> *eliminarLibro*();  
 case 3 -> *buscarLibroPorTitulo*();  
 case 4 -> *catalogo*.listarCatalogo();  
 case 5 -> *prestarLibro*();  
 case 6 -> *devolverLibro*();  
 case 7 -> *listarPrestamos*();  
 case 8 -> *listarHistorialAdelante*();  
 case 9 -> *listarHistorialAtras*();  
 case 10 -> *actualizarStock*();  
 case 0 -> { System.*out*.println("Saliendo… ¡Hasta luego!"); salir=true; }  
 default -> System.*out*.println("Opción inválida.");  
 }  
 }  
 }  
}