import java.time.LocalDate;  
import java.util.Scanner;  
  
class Libro {  
 int codigo;  
 String titulo;  
 String autor;  
 int stock;  
 boolean activo;  
  
 public Libro(int codigo, String titulo, String autor, int stock) {  
 this.codigo = codigo;  
 this.titulo = titulo;  
 this.autor = autor;  
 this.stock = stock;  
 this.activo = true;  
 }  
}  
  
class Prestamo {  
 int codigoLibro;  
 String usuario;  
 LocalDate fecha;  
 boolean devuelto;  
 Prestamo siguiente;  
  
 public Prestamo(int codigoLibro, String usuario) {  
 this.codigoLibro = codigoLibro;  
 this.usuario = usuario;  
 this.fecha = LocalDate.now();  
 this.devuelto = false;  
 this.siguiente = null;  
 }  
}  
  
class Operacion {  
 String tipo;  
 LocalDate fecha;  
 Operacion anterior;  
 Operacion siguiente;  
  
 public Operacion(String tipo) {  
 this.tipo = tipo;  
 this.fecha = LocalDate.now();  
 }  
}  
  
  
public class Main {  
  
 static final int MAX\_LIBROS = 100;  
 static Libro[] catalogo = new Libro[MAX\_LIBROS];  
 static int contadorLibros = 0;  
  
 static Prestamo prestamosActivos = null;  
 static Operacion historial = null;  
 static Operacion ultimoHistorial = null;  
  
 static Scanner sc = new Scanner(System.in);  
  
  
  
 static void altaLibro() {  
 if (contadorLibros >= MAX\_LIBROS) {  
 System.out.println("Catálogo lleno.");  
 return;  
 }  
 System.out.print("Código: ");  
 int codigo = sc.nextInt(); sc.nextLine();  
 for (int i = 0; i < contadorLibros; i++) {  
 if (catalogo[i].codigo == codigo && catalogo[i].activo) {  
 System.out.println("Ya existe un libro con ese código.");  
 return;  
 }  
 }  
 System.out.print("Título: ");  
 String titulo = sc.nextLine();  
 System.out.print("Autor: ");  
 String autor = sc.nextLine();  
 System.out.print("Stock: ");  
 int stock = sc.nextInt();  
 catalogo[contadorLibros++] = new Libro(codigo, titulo, autor, stock);  
 agregarOperacion("ALTA");  
 System.out.println("Libro agregado.");  
 }  
  
 static void bajaLibro() {  
 System.out.print("Código a dar de baja: ");  
 int codigo = sc.nextInt();  
 for (int i = 0; i < contadorLibros; i++) {  
 if (catalogo[i].codigo == codigo && catalogo[i].activo) {  
 catalogo[i].activo = false;  
 agregarOperacion("BAJA");  
 System.out.println("Libro dado de baja.");  
 return;  
 }  
 }  
 System.out.println("No encontrado.");  
 }  
  
 static Libro buscarLibroPorCodigo(int codigo) {  
 for (int i = 0; i < contadorLibros; i++) {  
 if (catalogo[i].codigo == codigo && catalogo[i].activo) return catalogo[i];  
 }  
 return null;  
 }  
  
 static void listarCatalogo() {  
 System.out.println("Catálogo de libros:");  
 for (int i = 0; i < contadorLibros; i++) {  
 if (catalogo[i].activo) {  
 System.out.println(catalogo[i].codigo + " - " + catalogo[i].titulo + " (" + catalogo[i].stock + ")");  
 }  
 }  
 }  
  
  
  
 static void prestarLibro() {  
 System.out.print("Código del libro: ");  
 int codigo = sc.nextInt(); sc.nextLine();  
 Libro libro = buscarLibroPorCodigo(codigo);  
 if (libro == null || libro.stock <= 0) {  
 System.out.println("No disponible.");  
 return;  
 }  
 System.out.print("Usuario: ");  
 String usuario = sc.nextLine();  
  
 Prestamo nuevo = new Prestamo(codigo, usuario);  
 nuevo.siguiente = prestamosActivos;  
 prestamosActivos = nuevo;  
  
 libro.stock--;  
 agregarOperacion("PRESTAMO");  
 System.out.println("Préstamo registrado.");  
 }  
  
 static void devolverLibro() {  
 System.out.print("Código del libro: ");  
 int codigo = sc.nextInt(); sc.nextLine();  
 Prestamo actual = prestamosActivos, anterior = null;  
 while (actual != null) {  
 if (actual.codigoLibro == codigo && !actual.devuelto) {  
 actual.devuelto = true;  
 Libro libro = buscarLibroPorCodigo(codigo);  
 if (libro != null) libro.stock++;  
  
 if (anterior == null) prestamosActivos = actual.siguiente;  
 else anterior.siguiente = actual.siguiente;  
 agregarOperacion("DEVOLUCION");  
 System.out.println("Devolución registrada.");  
 return;  
 }  
 anterior = actual;  
 actual = actual.siguiente;  
 }  
 System.out.println("Préstamo no encontrado.");  
 }  
  
 static void listarPrestamos() {  
 Prestamo actual = prestamosActivos;  
 System.out.println("Préstamos activos:");  
 while (actual != null) {  
 if (!actual.devuelto) {  
 System.out.println("Libro " + actual.codigoLibro + " prestado a " + actual.usuario + " (" + actual.fecha + ")");  
 }  
 actual = actual.siguiente;  
 }  
 }  
  
  
  
 static void agregarOperacion(String tipo) {  
 Operacion nueva = new Operacion(tipo);  
 if (historial == null) {  
 historial = nueva;  
 ultimoHistorial = nueva;  
 } else {  
 ultimoHistorial.siguiente = nueva;  
 nueva.anterior = ultimoHistorial;  
 ultimoHistorial = nueva;  
 }  
 }  
  
 static void listarHistorialAdelante() {  
 Operacion actual = historial;  
 System.out.println("Historial (adelante):");  
 while (actual != null) {  
 System.out.println(actual.tipo + " - " + actual.fecha);  
 actual = actual.siguiente;  
 }  
 }  
  
 static void listarHistorialAtras() {  
 Operacion actual = ultimoHistorial;  
 System.out.println("Historial (atrás):");  
 while (actual != null) {  
 System.out.println(actual.tipo + " - " + actual.fecha);  
 actual = actual.anterior;  
 }  
 }  
  
  
 public static void main(String[] args) {  
 int opcion;  
 do {  
 System.out.println("\n--- Menú Biblioteca ---");  
 System.out.println("1. Alta libro");  
 System.out.println("2. Baja libro");  
 System.out.println("3. Listar catálogo");  
 System.out.println("4. Prestar libro");  
 System.out.println("5. Devolver libro");  
 System.out.println("6. Listar préstamos activos");  
 System.out.println("7. Historial adelante");  
 System.out.println("8. Historial atrás");  
 System.out.println("0. Salir");  
 System.out.print("Opción: ");  
 opcion = sc.nextInt();  
 sc.nextLine();  
 switch (opcion) {  
 case 1: altaLibro(); break;  
 case 2: bajaLibro(); break;  
 case 3: listarCatalogo(); break;  
 case 4: prestarLibro(); break;  
 case 5: devolverLibro(); break;  
 case 6: listarPrestamos(); break;  
 case 7: listarHistorialAdelante(); break;  
 case 8: listarHistorialAtras(); break;  
 }  
 } while (opcion != 0);  
 }  
}