using System;

using System.Collections.Generic;

using System.Xml;

namespace LibraryManagementSystem

{

// Book class using Pascal for its name and properties, Hungarian for its private fields.

public class Book

{

private string strTitle;

private string strAuthor;

private int intYearPublished;

private string strISBN;

public string Title { get { return strTitle; } set { strTitle = value; } }

public string Author { get { return strAuthor; } set { strAuthor = value; } }

public int YearPublished { get { return intYearPublished; } set { intYearPublished = value; } }

public string ISBN { get { return strISBN; } set { strISBN = value; } }

public Book(string title, string author, int year, string isbn)

{

strTitle = title;

strAuthor = author;

intYearPublished = year;

strISBN = isbn;

}

public override string ToString()

{

return $"{Title} by {Author} ({YearPublished}), ISBN: {ISBN}";

}

}

public class Member

{

private string strName;

private int intMemberID;

public string Name { get { return strName; } set { strName = value; } }

public int MemberID { get { return intMemberID; } set { intMemberID = value; } }

public Member(int memberId, string name)

{

intMemberID = memberId;

strName = name;

}

public override string ToString()

{

return $"Member: {Name} (ID: {MemberID})";

}

}

public class BorrowTransaction

{

private int intTransactionID;

private Member member;

private Book book;

private DateTime dtBorrowDate;

private DateTime dtReturnDate;

public int TransactionID { get { return intTransactionID; } }

public Member Borrower { get { return member; } }

public Book BorrowedBook { get { return book; } }

public DateTime BorrowDate { get { return dtBorrowDate; } }

public DateTime ReturnDate { get { return dtReturnDate; } set { dtReturnDate = value; } }

public BorrowTransaction(int transactionId, Member member, Book book, DateTime borrowDate)

{

intTransactionID = transactionId;

this.member = member;

this.book = book;

dtBorrowDate = borrowDate;

dtReturnDate = DateTime.MinValue;

}

public void CompleteTransaction(DateTime returnDate)

{

dtReturnDate = returnDate;

}

public override string ToString()

{

string returnInfo = (dtReturnDate == DateTime.MinValue) ? "Not returned" : dtReturnDate.ToShortDateString();

return $"Transaction {TransactionID}: {Borrower.Name} borrowed \"{BorrowedBook.Title}\" on {BorrowDate.ToShortDateString()}, Return: {returnInfo}";

}

}

public class Library

{

// Lists using Hungarian notation

private List<Book> lstBooks;

private List<Member> lstMembers;

private List<BorrowTransaction> lstTransactions;

private int intNextTransactionID;

public Library()

{

lstBooks = new List<Book>();

lstMembers = new List<Member>();

lstTransactions = new List<BorrowTransaction>();

intNextTransactionID = 1;

}

public void AddBook(Book book)

{

lstBooks.Add(book);

}

public void RegisterMember(Member member)

{

lstMembers.Add(member);

}

public BorrowTransaction BorrowBook(int memberId, string isbn)

{

Member member = lstMembers.Find(m => m.MemberID == memberId);

Book book = lstBooks.Find(b => b.ISBN == isbn);

if (member == null || book == null)

{

throw new Exception("Member or Book not found");

}

BorrowTransaction transaction = new BorrowTransaction(intNextTransactionID++, member, book, DateTime.Now);

lstTransactions.Add(transaction);

return transaction;

}

public void ReturnBook(int transactionId)

{

BorrowTransaction transaction = lstTransactions.Find(t => t.TransactionID == transactionId);

if (transaction == null)

{

throw new Exception("Transaction not found");

}

transaction.CompleteTransaction(DateTime.Now);

}

public void ExportLibraryData(string filePath)

{

XmlWriterSettings settings = new XmlWriterSettings { Indent = true };

using (XmlWriter writer = XmlWriter.Create(filePath, settings))

{

writer.WriteStartDocument();

writer.WriteStartElement("Library");

writer.WriteStartElement("Books");

foreach (var book in lstBooks)

{

writer.WriteStartElement("Book");

writer.WriteElementString("Title", book.Title);

writer.WriteElementString("Author", book.Author);

writer.WriteElementString("YearPublished", book.YearPublished.ToString());

writer.WriteElementString("ISBN", book.ISBN);

writer.WriteEndElement();

}

writer.WriteEndElement(); // Books

writer.WriteStartElement("Members");

foreach (var member in lstMembers)

{

writer.WriteStartElement("Member");

writer.WriteElementString("Name", member.Name);

writer.WriteElementString("MemberID", member.MemberID.ToString());

writer.WriteEndElement();

}

writer.WriteEndElement(); // Members

writer.WriteStartElement("Transactions");

foreach (var transaction in lstTransactions)

{

writer.WriteStartElement("Transaction");

writer.WriteElementString("TransactionID", transaction.TransactionID.ToString());

writer.WriteElementString("Borrower", transaction.Borrower.Name);

writer.WriteElementString("BookTitle", transaction.BorrowedBook.Title);

writer.WriteElementString("BorrowDate", transaction.BorrowDate.ToString());

writer.WriteElementString("ReturnDate", transaction.ReturnDate == DateTime.MinValue ? "Not returned" : transaction.ReturnDate.ToString());

writer.WriteEndElement();

}

writer.WriteEndElement(); // Transactions

writer.WriteEndElement(); // Library

writer.WriteEndDocument();

}

}

}

public class Program

{

public static void Main(string[] args)

{

Library lib = new Library();

// Add sample books

lib.AddBook(new Book("1984", "George Orwell", 1949, "ISBN-1234567890"));

lib.AddBook(new Book("Brave New World", "Aldous Huxley", 1932, "ISBN-0987654321"));

// Register sample members

lib.RegisterMember(new Member(1, "John Doe"));

lib.RegisterMember(new Member(2, "Jane Smith"));

// Borrow a book

BorrowTransaction trans = lib.BorrowBook(1, "ISBN-1234567890");

Console.WriteLine(trans);

// Return the book

lib.ReturnBook(trans.TransactionID);

Console.WriteLine(trans);

// Export library data to XML

lib.ExportLibraryData("library.xml");

Console.WriteLine("Library data exported to library.xml");

}

}

}