|  |
| --- |
| Member |
| -name : string  -icNo : int  -libraryNo : string  -memberType : string  -dateofBirth : int  -telNo : int  -address : string  -state : string  -postcode : int  -country : string  -MAX\_NUMBER\_OF\_LOANITEMS = 10 :int |
| +loanItem: Item[ ]  +Member(theName : string, theIC : int, theLibrary : string, theType : string, theDOB : int,  theTel : int, theAddress : string, theState : string, thePostCode : int, theCountry : string)  +Member(aName : string)  +BorrowItem(borrow : Item) |

|  |
| --- |
| Item |
| -name : string  -code : string  -expday : DateTime  -borrowed : bool |
| +Item(aName : string, aCode : string, theExpDay : DateTime, theBorrowed : bool)  +AddExpiryDate() |

|  |
| --- |
| Book |
|  |
| +Book(name : string, code : string,  expDay : DateTime, borrowed : bool  +AddExpiryDate() |

|  |
| --- |
| DVD |
|  |
| +Book(name : string, code : string,  expDay : DateTime, borrowed : bool  +AddExpiryDate() |

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Collections;

using System.IO;

public class Member

{

private string name;

public string Name

{

get { return name; }

set

{

name = value;

}

}

private int icNo;

public int IcNo

{

get { return icNo; }

set

{

icNo = value;

}

}

private string libraryNo;

public string LibraryNo

{

get { return libraryNo; }

set

{

libraryNo = value;

}

}

private string memberType;

public string MemberType

{

get { return memberType; }

set

{

memberType = value;

}

}

private int dateofBirth;

public int DateOfBirth

{

get { return dateofBirth; }

set

{

dateofBirth = value;

}

}

private int telNo;

public int TelNo

{

get { return telNo; }

set

{

telNo = value;

}

}

private string address;

public string Address

{

get { return address; }

set

{

address = value;

}

}

private string state;

public string State

{

get { return state; }

set

{

state = value;

}

}

private int postcode;

public int Postcode

{

get { return postcode; }

set

{

postcode = value;

}

}

private string country;

public string Country

{

get { return country; }

set

{

country = value;

}

}

private int itemLoan;

public int ItemLoan

{

get { return itemLoan; }

}

public Item[] loanItem;

private static readonly int MAX\_NUMBER\_OF\_LOANITEMS = 10;

public Member(string theName, int theIc, string theLibrary, string theType, int theDOB, int theTel, string theAddress, string theState, int thePostCode, string theCountry)

{

name = theName;

icNo = theIc;

libraryNo = theLibrary;

memberType = theType;

dateofBirth = theDOB;

telNo = theTel;

address = theAddress;

state = theState;

postcode = thePostCode;

country = theCountry;

itemLoan = 0;

loanItem = new Item[MAX\_NUMBER\_OF\_LOANITEMS];

}

public Member(string aName)

{

name = aName;

}

public void BorrowItem(Item borrow)

{

for (itemLoan = 0; itemLoan < MAX\_NUMBER\_OF\_LOANITEMS; itemLoan++)

{

loanItem[itemLoan] = borrow;

itemLoan++;

}

}

}

public abstract class Item

{

private string name;

public string Name

{

get { return name; }

set

{

name = value;

}

}

private string code;

public string Code

{

get { return code; }

set

{

code = value;

}

}

private DateTime expday;

public DateTime ExpDay

{

get { return expday; }

set

{

expday = value;

}

}

private bool borrowed;

public bool Borrowed

{

get { return borrowed; }

set

{

borrowed = value;

}

}

public Item(string aName, string aCode, DateTime theExpDay, bool theBorrowed)

{

Name = aName;

Code = aCode;

ExpDay = theExpDay;

Borrowed = theBorrowed;

}

public abstract void AddExpiryDate();

}

public class Book : Item

{

public Book(string name, string code, DateTime expDay, bool borrowed) : base(name, code, expDay, borrowed)

{ }

public override void AddExpiryDate()

{

ExpDay = DateTime.Now.AddDays(14);

Borrowed = true;

}

}

public class DVD : Item

{

public DVD(string name, string code, DateTime expDay, bool borrowed) : base(name, code, expDay, borrowed)

{ }

public override void AddExpiryDate()

{

ExpDay = DateTime.Now.AddDays(7);

Borrowed = true;

}

}

namespace WindowsFormsApp1

{

public partial class LibrarySystem : Form

{

private ArrayList MemberList, BookList, DvdList;

private Member borrower;

private Book book;

private DVD dvd;

private Member members;

public LibrarySystem()

{

InitializeComponent();

MemberList = new ArrayList();

BookList = new ArrayList();

DvdList = new ArrayList();

}

private void btnCreateMember\_Click(object sender, EventArgs e)

{

//To check is there any blank spaces which is not filled by user

if (tbxMName.Text.Length == 0 || tbxICNo.Text.Length == 0 || tbxLibraryNo.Text.Length == 0 || tbxMType.Text.Length == 0 || tbxMDOB.Text.Length == 0

|| tbxMTel.Text.Length == 0 || rtbxMAddress.Text.Length == 0 || tbxMState.Text.Length == 0 || tbxMPostcode.Text.Length == 0 || tbxMCountry.Text.Length == 0)

{ MessageBox.Show("Please fill up all blanks."); }

else

{

string name = tbxMName.Text;

int ic = Convert.ToInt32(tbxICNo.Text);

string lib = tbxLibraryNo.Text;

string type = tbxMType.Text;

int dob = Convert.ToInt32(tbxMDOB.Text);

int tel = Convert.ToInt32(tbxMTel.Text);

string address = rtbxMAddress.Text;

string state = tbxMState.Text;

int postcode = Convert.ToInt32(tbxMPostcode.Text);

string country = tbxMCountry.Text;

members = new Member(name, ic, lib, type, dob, tel, address, state, postcode, country);

MemberList.Add(members);

MessageBox.Show("Successfully Added!");

tbxMName.Clear();

tbxICNo.Clear();

tbxLibraryNo.Clear();

tbxMType.Clear();

tbxMDOB.Clear();

tbxMTel.Clear();

rtbxMAddress.Clear();

tbxMState.Clear();

tbxMPostcode.Clear();

tbxMCountry.Clear();

}

}

private void btnAdd\_Click(object sender, EventArgs e)

{

//To check is the radio button checked at least one by the user

if (rbnBook.Checked)

{

//To double check is there any blank spaces

if (tbxAddName.Text.Length == 0 || tbxAddCode.Text.Length == 0)

{ MessageBox.Show("Please fill in the blanks."); }

else

{

string name = tbxAddName.Text;

string code = tbxAddCode.Text;

book = new Book(name, code, DateTime.Now, false);

BookList.Add(book);

MessageBox.Show("Added successfully.");

tbxAddName.Clear();

tbxAddCode.Clear();

}

}

else if (rbnDVD.Checked)

{

//To double check is there any blank spaces

if (tbxAddName.Text.Length == 0 || tbxAddCode.Text.Length == 0)

{ MessageBox.Show("Please fill in the blanks."); }

else

{

string name = tbxAddName.Text;

string code = tbxAddCode.Text;

dvd = new DVD(name, code, DateTime.Now, false);

DvdList.Add(dvd);

MessageBox.Show("Added successfully.");

tbxAddName.Clear();

tbxAddCode.Clear();

}

}

else

{ MessageBox.Show("Please tick BOOK or DVD!!"); }

}

//Search member method

public Member FindAMember(string SearchMember)

{

Member member = null;

bool found = false;

int i = 0;

int count = MemberList.Count;

while (i < count && !found)

{

member = (Member)MemberList[i];

if (member.Name == SearchMember)

{

found = true;

}

else

{ i++; }

}

if (found)

{ return member; }

else

{ return null; }

}

private void btnSearch\_Click(object sender, EventArgs e)

{

string search = tbxSearch.Text;

borrower = FindAMember(search);

if (borrower != null)

MessageBox.Show("Name: " + borrower.Name + "\nItems: " + borrower.loanItem);

else

MessageBox.Show("No member found!");

}

private void btnCheckMember\_Click(object sender, EventArgs e)

{

string borrow = tbxBName.Text;

members = FindAMember(borrow);

if(members != null)

{

tbxBLibraryNo.Text = members.LibraryNo;

gbxBBook.Enabled = true;

btnBorrowBook.Enabled = false;

gbxBDVD.Enabled = true;

btnBorrowDVD.Enabled = false;

}

else if(members == null)

{

MessageBox.Show("No record found!");

}

else

{ MessageBox.Show("Max item borrowed reached! Unable to borrow more items."); }

}

//Method to search for book

public Book FindABook(string SearchBook)

{

Book books = null;

bool found = false;

int i = 0;

int count = BookList.Count;

while (i < count && !found)

{

books = (Book)BookList[i];

if (books.Code == SearchBook)

{ found = true; }

else

{ i++; }

}

if (found)

{ return books; }

else

{ return null; }

}

//Method to find Dvd

public DVD FindADvd(string SearchDvd)

{

DVD dvds = null;

bool found = false;

int i = 0;

int count = DvdList.Count;

while (i < count && !found)

{

dvds = (DVD)DvdList[i];

if (dvds.Code == SearchDvd)

{ found = true; }

else

{ i++; }

}

if (found)

{ return dvds; }

else

{ return null; }

}

private void btnBookCheck\_Click(object sender, EventArgs e)

{

string bookCode = tbxBBookCode.Text;

Book book = FindABook(bookCode);

if(bookCode != null)

{

lblBBookName.Text = book.Name;

btnBorrowBook.Enabled = true;

}

else

{

MessageBox.Show("No Result Found!");

}

}

private void btnDVDCheck\_Click(object sender, EventArgs e)

{

string dvdCode = tbxBDvdCode.Text;

DVD dvd = FindADvd(dvdCode);

if(dvdCode != null)

{

lblBDvdName.Text = dvd.Name;

btnBorrowDVD.Enabled = true;

}

else

{

MessageBox.Show("No Result Found!");

}

}

private void LibrarySystem\_Load(object sender, EventArgs e)

{

string name;

int ic;

string lib;

string type;

int dob;

int tel;

string address;

string state;

int postcode;

string country;

string code;

DateTime expDay;

bool borrowed;

Member members;

Book books;

DVD dvds;

StreamReader reader;

try

{

if(File.Exists("member.txt"))

{

reader = new StreamReader("member.txt");

name = reader.ReadLine();

while(name != null)

{

ic = Convert.ToInt32(reader.ReadLine());

lib = reader.ReadLine();

type = reader.ReadLine();

dob = Convert.ToInt32(reader.ReadLine());

tel = Convert.ToInt32(reader.ReadLine());

address = reader.ReadLine();

state = reader.ReadLine();

postcode = Convert.ToInt32(reader.ReadLine());

country = reader.ReadLine();

members = new Member(name, ic, lib, type, dob, tel, address, state, postcode, country);

MemberList.Add(members);

name = reader.ReadLine();

}

reader.Close();

}

if (File.Exists("book.txt"))

{

reader = new StreamReader("book.txt");

name = reader.ReadLine();

while(name != null)

{

code = reader.ReadLine();

expDay = Convert.ToDateTime(reader.ReadLine());

borrowed = Convert.ToBoolean(reader.ReadLine());

books = new Book(name, code, expDay, borrowed);

BookList.Add(books);

name = reader.ReadLine();

}

reader.Close();

}

if (File.Exists("dvd.txt"))

{

reader = new StreamReader("dvd.txt");

name = reader.ReadLine();

while (name != null)

{

code = reader.ReadLine();

expDay = Convert.ToDateTime(reader.ReadLine());

borrowed = Convert.ToBoolean(reader.ReadLine());

dvds = new DVD(name, code, expDay, borrowed);

DvdList.Add(dvd);

name = reader.ReadLine();

}

reader.Close();

}

}

catch (IOException)

{

MessageBox.Show("Error opening file!");

}

}

private void LibrarySystem\_FormClosing(object sender, FormClosingEventArgs e)

{

StreamWriter writer;

if(File.Exists("member.txt"))

try

{

writer = new StreamWriter("member.txt", true);

writer.WriteLine(MemberList);

writer.Close();

}

catch(IOException exc)

{

MessageBox.Show("Error: " + exc.Message);

}

if(File.Exists("book.txt"))

try

{

writer = new StreamWriter("book.txt", true);

writer.WriteLine(BookList);

writer.Close();

}

catch (IOException exc)

{

MessageBox.Show("Error: " + exc.Message);

}

if (File.Exists("dvd.txt"))

try

{

writer = new StreamWriter("dvd.txt", true);

writer.WriteLine(DvdList);

writer.Close();

}

catch (IOException exc)

{

MessageBox.Show("Error: " + exc.Message);

}

else

try

{

writer = new StreamWriter("member.txt");

writer.WriteLine(MemberList);

writer = new StreamWriter("book.txt");

writer.WriteLine(BookList);

writer = new StreamWriter("dvd.txt");

writer.WriteLine(DvdList);

writer.Close();

}

catch (IOException exc)

{

MessageBox.Show("Error: " + exc.Message);

}

}

private void btnBorrowBook\_Click(object sender, EventArgs e)

{

string name = tbxBName.Text;

string itemName = lblBBookName.Text;

string itemCode = tbxBBookCode.Text;

borrower = new Member(name);

Book borrow = new Book(itemName, itemCode, DateTime.Now.AddDays(14), true);

borrower.BorrowItem(borrow);

MessageBox.Show("Item borrowed.");

lblBBookReturnDate2.Text = borrow.ExpDay.Date.ToString();

}

private void btnBorrowDVD\_Click(object sender, EventArgs e)

{

string name = tbxBName.Text;

string itemName = lblBDvdName.Text;

string itemCode = tbxBDvdCode.Text;

borrower = new Member(name);

DVD borrow = new DVD(itemName, itemCode, DateTime.Now.AddDays(7), true);

borrower.BorrowItem(borrow);

MessageBox.Show("Item borrowed.");

lblBDvdReturnDate2.Text = borrow.ExpDay.Date.ToString();

}

}

}

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new LibrarySystem());

}

}