**3.Практическая часть.**

**3.1 Постановка задачи:**

Необходимо составить консольное приложение на тему «Библиотеки» в котором будет выполняться вывод необходимой информации для пользователя по его требованию.

**3.2 Описание предметной области:**

Система должна выводить информацию об объекте по команде пользователя. Также выводить всю информацию о Книгах или клиенте.

**3.3 Проектирование библиотеки классов:**

|  |  |
| --- | --- |
| **Книга**  Кол-во книг на складе  Год выпуска  Название  Краткое описание  Стоимость | **Читатель**  ФИО  Адрес  Телефон  Email  Читательского билета |
| **Посещение**  Название книги  Возвращение или взятие книги  Номер читательского | **Читательский билет**  Номер читательского  Срок действия  Взятые книги |

**Реализация библиотеки классов:**

Для книг и посетителей реализованнны два класса book и bookMEM с полями

class Book

{

public string NameOfBook, author, shortdescription;

public string year;

public string coast;

public int howmuch;

}

class bookMEM

{

public string FIO;

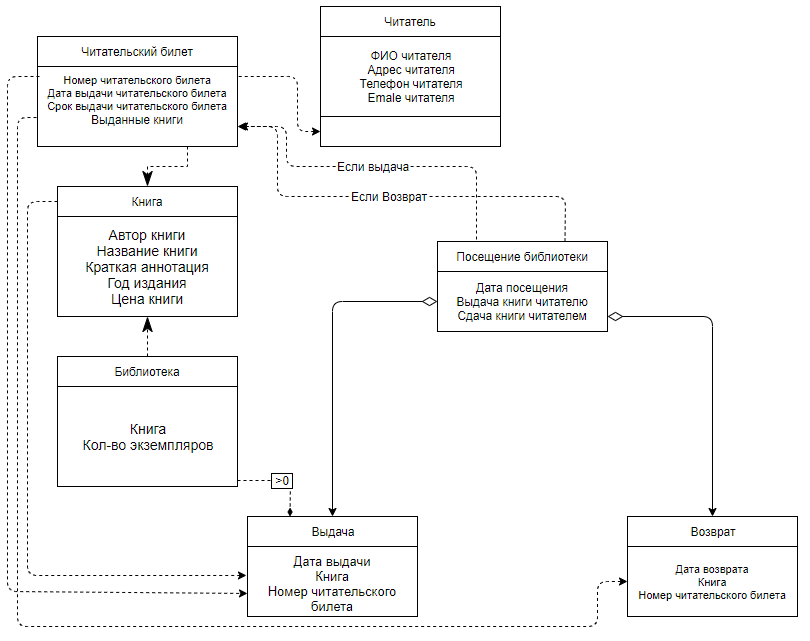
public int numberofbookmem;

DateTime firedate;

}

**3.4 Реализация методов работы с вводом-выводом:**

Для ввода реализован метод по считыванию необходимых данных для последующей записи их в файл, а также при необходимости чтению из него.



При посещении библиотеки пользователь выбирает что ему необходимо сделать взять новую книгу или вернуть старую, все данные записываются в его читательский билет, все читательские билеты хранятся в базе данных программы в виде файлов IDCARD и visitorinfo. При необходимости работник библиотеки может зарегестрировать нового читателя или книгу в базе данных(для этого была создана отдельная ветка кода).

Все данные удобно структурированны что даёт возможность модифицировать код в случаен необходимости. Сам же код имеет множество дыр и может сбоить в любой момент т.к. автору было и так понятно что можно было при вводе приводить всё к нижнему реестру с помощью функции ToLower(), но я на это тратить время не стал.

У каждого читателя есть своя отдельная папка в файловой структуре данных.

namespace Kursach

{

class Program

{

static int papka(string path)

{

DirectoryInfo NewDirectory = new DirectoryInfo(path);

if (!NewDirectory.Exists)

{

NewDirectory.Create();

return 0;

}

else

{

Console.WriteLine("Can't create new derectory, lable is captured");

Console.WriteLine("Input new lable");

return 1;

}

}

static void Main(string[] args)

{

Console.OutputEncoding = Encoding.UTF8;

string CurentSession, CurentChoise;

for (; ; )

{

Console.Clear();

Console.WriteLine("Добро пожаловать в библиотеку");

Console.WriteLine("Кто вы?");

Console.WriteLine("посетитель или работник");

CurentSession = Console.ReadLine();

Console.Clear();

switch (CurentSession)

{

case "посетитель": {

Console.WriteLine("Здравствуйте посетитель");

Console.WriteLine("Введите ваше ФИО");

String CurentUser = Console.ReadLine();

if (Directory.Exists("visitors" + "\\" + CurentUser))

{

Console.WriteLine("Цель вашего визита?");

Console.WriteLine("Чтобы взять книгу, введите \"Claim\" ");

Console.WriteLine("Чтобы взять книгу, введите \"Return\" ");

string curentmove = Console.ReadLine();

switch (curentmove.ToLower())

{

case "claim":

{

Console.Clear();

Console.WriteLine("Введите название книги которую хотите взять: ");

string curentfoudedbook;

curentfoudedbook = Console.ReadLine();

Console.Clear();

if (File.Exists("books" + "\\" + curentfoudedbook + ".txt"))

{

Book curentclaimingbook = new Book();

curentclaimingbook.NameOfBook = curentfoudedbook;

string[] curentfile = File.ReadAllLines("books" + "\\" + curentfoudedbook + ".txt");

curentclaimingbook.year = curentfile[2].Substring(13);

curentclaimingbook.shortdescription = curentfile[1].Substring(18);

curentclaimingbook.coast = curentfile[3].Substring(12);

curentclaimingbook.howmuch = Convert.ToInt32(curentfile[4].Substring(16));

int howmuch = curentclaimingbook.howmuch - 1;

StreamWriter curenthowmuchbook = new StreamWriter("books" + "\\" + curentfoudedbook + ".txt");

curenthowmuchbook.WriteLine("Название: " + curentclaimingbook.NameOfBook);

curenthowmuchbook.WriteLine("Краткое описание: " + curentclaimingbook.shortdescription);

curenthowmuchbook.WriteLine("Год выпуска: " + curentclaimingbook.year);

curenthowmuchbook.WriteLine("Цена книги: " + curentclaimingbook.coast);

curenthowmuchbook.WriteLine("Книг на складе: " + howmuch);

curenthowmuchbook.Close();

Console.WriteLine("Взята книга: " + curentclaimingbook.NameOfBook);

StreamWriter curentclaimuser = new StreamWriter("visitors" + "\\" + CurentUser + "\\" + "IDCARD.txt", true);

curentclaimuser.WriteLine("Взята книга: " + curentclaimingbook.NameOfBook);

curentclaimuser.Close();

}

else Console.WriteLine("Данной книги нет на наших складых");

string temp = Console.ReadLine();

}

break;

case "return":

{

Console.Clear();

Console.WriteLine("Какую книгу вы хотите вернуть?");

string[] claimedbooks = File.ReadAllLines("visitors" + "\\" + CurentUser + "\\" + "IDCARD.txt");

for (int i = 4; i < claimedbooks.Length; i++)

{

Console.WriteLine(claimedbooks[i]);

}

string returningbook = Console.ReadLine();

for (int u = 4; u< claimedbooks.Length;u++)

{

if (claimedbooks[u].Substring(13).Trim() == returningbook)

{

claimedbooks[u] = claimedbooks[u].Substring(claimedbooks[u].Length);

}

}

Console.WriteLine("Книга возвращена");

StreamWriter curentretbookurnuser = new StreamWriter("visitors" + "\\" + CurentUser + "\\" + "IDCARD.txt");

for (int e= 0;e < claimedbooks.Length;e++)

{

curentretbookurnuser.WriteLine(claimedbooks[e]);

}

curentretbookurnuser.Close();

FileInfo bookfile = new FileInfo("books" + "\\" + returningbook + ".txt");

if (bookfile.Exists)

{

string[] retbooks = File.ReadAllLines("books" + "\\" + returningbook + ".txt");

string findingbook = retbooks[0].Substring(9).Trim();

if (findingbook == returningbook)

{

StreamWriter curentreturnuser = new StreamWriter("books" + "\\" + returningbook + ".txt");

bookMEM bookret = new bookMEM();

retbooks [4] = "Книг на складе: " + Convert.ToString(Convert.ToInt32(retbooks[4].Substring(16)) + 1);

for (int r = 0; r < retbooks.Length; r++)

{

curentreturnuser.WriteLine(retbooks[r]);

}

curentreturnuser.Close();

}

else break;

}

}

break;

}

}

else Console.WriteLine("Мы не нашли ваш читательский билет, " + "обратитесь к сотруднику библиотеки для регистрации нового");

}

break;

case "работник":

Console.WriteLine("Приступим к работе");

for (int i = Convert.ToInt32(File.ReadAllText("Statistic.txt")); ;)

{

Console.WriteLine("Добавить новый элемент(new) или промотреть информацию о книге или читателю?");

Console.WriteLine("new or info");

CurentChoise = Console.ReadLine();

Console.Clear();

switch (CurentChoise.ToLower())

{

case "new": {

Console.WriteLine("Хотете добавить новый читательский билет(IDCARD) или книгу?(book)");

CurentChoise = Console.ReadLine();

switch (CurentChoise.ToLower())

{

case "idcard":

{

File.WriteAllText("Statistic.txt", Convert.ToString(i + 1));

Console.Clear();

Console.WriteLine("Введите ФИО нового ID\_CARD");

string curentIDCARD;

curentIDCARD = Console.ReadLine();

while (papka("visitors" + "\\" + curentIDCARD) == 1) {

curentIDCARD = Console.ReadLine();

} ;

string curentPath;

curentPath = "visitors" + "\\" + curentIDCARD + "\\" + "VisitorsInfo.txt";

FileInfo curentfile = new FileInfo(curentPath);

StreamWriter curent = new StreamWriter(curentPath);

curent.WriteLine("ФИО: " + curentIDCARD);

Console.WriteLine("Введите адрес читателя");

curent.WriteLine("Адрес: " + Console.ReadLine());

Console.WriteLine("Введите номер телефона читателя");

curent.WriteLine("Номер телефона: " + Console.ReadLine());

Console.WriteLine("Введите Emale");

curent.WriteLine("Emale: " + Console.ReadLine());

curent.Close();

curentPath = "visitors" + "\\" + curentIDCARD + "\\" + "IDCARD.txt";

FileInfo curentfilecard = new FileInfo(curentPath);

StreamWriter curentcard = new StreamWriter(curentPath);

curentcard.WriteLine("FIO: " + curentIDCARD);

curentcard.WriteLine("ID: " + i);

curentcard.WriteLine("Date of Registration: " + DateTime.Today);

curentcard.WriteLine("Fire Date: " + DateTime.Today.AddYears(2));

curentcard.Close();

}

break;

case "book":

{

Console.Clear();

Console.WriteLine("Введите название книги");

string curentidbook = Console.ReadLine().ToLower();

string curentPath;

curentPath = "books" + "\\" + curentidbook + ".txt";

FileInfo curentbookfile = new FileInfo(curentPath);

StreamWriter curentbook = new StreamWriter(curentPath);

curentbook.WriteLine("Название: " + curentidbook);

Console.WriteLine("Введите краткое описание");

curentbook.WriteLine("Краткое описание: " + Console.ReadLine());

Console.WriteLine("Введите год выпуска");

curentbook.WriteLine("Год выпуска: " + Console.ReadLine());

Console.WriteLine("Введите цену книги");

curentbook.WriteLine("Цена книги: " + Console.ReadLine());

Console.WriteLine("Сколько книг есть на складе?");

curentbook.WriteLine("Книг на складе: " + Console.ReadLine());

curentbook.Close();

}

break;

}

}

Break;

case "info": {

Console.WriteLine("Что хотите узнать: о читательском билете(IDCARD) или книге?(book)");

CurentChoise = Console.ReadLine();

switch (CurentChoise.ToLower())

{

case "idcard":

{

Console.Clear();

Console.WriteLine("Введите ФИО IDCARD");

string curentfoichoise = Console.ReadLine();

string[] fileinfo = File.ReadAllLines("visitors" + "\\" + curentfoichoise + "\\" + "IDCARD.txt");

for (int x = 0;x < fileinfo.Length; x++)

{

Console.WriteLine(fileinfo[x]);

}

Console.ReadKey();

}

break;

case "book":

{

Console.Clear();

Console.WriteLine("Введите название книги");

string curentbookchoise = Console.ReadLine();

string[] fileinfo = File.ReadAllLines("book" + "\\" + curentbookchoise + ".txt");

for (int x = 0; x < fileinfo.Length; x++)

{

Console.WriteLine(fileinfo[x]);

Остаток кода это куча break и один ReadKey так что думаю это будет здесь лишним.

1. **Заключение**

В ходе работы я закрепил свои знания синтаксиса и навык создания классов для реализации сложных баз данных на конкретных примерах.