Учреждение образования Федерации профсоюзов Беларуси

«Международный университет «МИТСО»

Рег. № \_\_\_\_\_\_\_\_\_\_ Кафедра информационных систем и технологий

Дата \_\_\_\_\_\_\_\_\_2020

**КУРСОВАЯ РАБОТА**

на тему \_\_\_\_Название\_\_\_Разработать программный модуль «Картотека недвижимости»\_\_\_\_\_\_\_\_

по дисциплине \_Название\_\_\_\_Основы конструирования программ\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| Основные замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Отметка о допуске курсовой работы к  защите:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_  Подпись научного руководителя:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  Урбанович Владислав Сергеевич (Ф.И.О. – полностью)  Курс \_\_1\_\_, группа \_\_\_1920\_\_\_\_  Факультет \_\_экономический\_\_\_  Специальность \_\_Информационные системы и технологии\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Научный руководитель:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (должность, ученая степень, ученое звание)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Ф.И.О. – полностью) |

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc70793473)

[1.ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 4](#_Toc70793474)

[**1.1. Общие положения 4**](#_Toc70793475)

[**1.2. Постановка задачи 4**](#_Toc70793476)

[**1.3. Спецификация входных и выходных данных 4**](#_Toc70793477)

[**1.4. Выбор средства 5**](#_Toc70793478)

[**1.5.Описывается требуемое оборудование и программное обеспечение 6**](#_Toc70793479)

[2.ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРИЛОЖЕНИЯ 7](#_Toc70793480)

[**2.1. Алгоритм решения 7**](#_Toc70793481)

[**2.2. Описание программы 7**](#_Toc70793482)

[**2.3. Результаты работы программы 9**](#_Toc70793483)

[РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 10](#_Toc70793484)

[ЗАКЛЮЧЕНИЕ 10](#_Toc70793485)

[ПРИЛОЖЕНИЕ 11](#_Toc70793486)

# ВВЕДЕНИЕ

Была поставлена задача разработать программный продукт «Учёт выдачи книг в библиотеки», предназначенный для использования библиотекарем. В программном продукте необходимо реализовать учёт книг и читателей.

Книги имеют следующие показатели: название книги, имя автора, фамилия автора.

Читатели имеют следующие показатели: фамилия, имя, отчество и книги, которые взял читатель.

Программа создана для учёта книг в библиотеки.

Цель разработки обеспечить работу программного продукта быстро и максимально эффективно, удовлетворять запросы библиотекаря, а также должен быть возможность пополнять базу данных, для учёта книг. Основная же задача стоит в высокой оптимизация, чтобы библиотекарь мог запустить даже на самом слабом устройстве.

Задачи данного программного продукта: быстро и гибко редактироваться со стороны библиотекаря. С точки зрения использования пользователя программа должна иметь красивый, понятный и интуитивный интерфейс чтобы было пользователи приятно пользоваться программой. Так же требуется предусмотреть возможность добавления новых функций в программе.

Для разработки программы был использован текстовый файл и язык программирования C++ разработанный в среде Visual Studio компаний Microsoft.

Visual Studio-линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой Windows Forms, а также веб-сайты, веб-приложения, веб-службы, как в родном, так и в управляемом кодах для всех платформ, поддерживающих Windows, Windows Mobile, Windows Ce, .Net Framework, Xbox, Windows Phone .Net Compact Framework и Silverling.

# 1.ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1. Общие положения

Большое количество книг требует католагализации и учёта. Данная процедура крайне муторная и затратная. Так как требует большого количества бумажных записей. Именно эти обстоятельства привели к необходимости создания данного программного продукта.

Была поставлена задача разработать программный продукт «Учёт выдачи книг в библиотеки», предназначенный для использования библиотекарем. В программном продукте необходимо реализовать учёт книг и читателей.

Книги имеют следующие показатели: название книги, имя автора, фамилия автора.

Читатели имеют следующие показатели: фамилия, имя, отчество и книги, которые взял читатель.

Данный программный продукт выгодно отличается от других подобных тем что, пользуясь им, пользователь может не только структурированно хранить данные об книгах, но и производить взаимодействие с читателями.

## 1.2. Постановка задачи

Исходя из задач, описанных выше главными требованиями к языку, на котором будет написана программы — это тонкая настройка и возможность использованием минимум вычислительных ресурсов добиваться максимум результата.

## 1.3. Спецификация входных и выходных данных

В данном программном продукте используются следующие типы данных типа данных: sting, int, bool. А также присутствуют пользовательские типы данных. Author, Book, Reader.

boll – используется только внутри программы, для проверки, подсчёта, иннерваций и так далее. Пользователь на данный этого типа влиять и пользоваться не может.

String – используется в двух случаях. Либо для наименования данных (присвоение данным название удобного пользователя). Во втором же случае данный тип данных используется для вывода всех данных на экран.

Int – используется для выбора операции.

Author, Book, Reader – работают в связке. При помощи их и выполняется основной функционал данного программного продукта. Book имеет класс Author. Reader – имею список Book.

## 1.4. Выбор средства

Для выполнения данных задач очень подходит такой язык программирования как с++. Так как его из-за того, что он является весьма низкоуровневым языком программирования, а значит он больше других приближен к машинному коду, из-за этого с++ дает возможность оптимизировать проект так что даже на очень старых устройствах он будет работать.

Среда разработки Microsoft Visual Studio. Вид приложения – консольное. Программа построена на базе объектно-ориентированного программирования. Способ организации данных поля классов. Способ хранения данных – динамические массивы. Каждая логическая завершенная данных подзадача программы реализована в виде методов. Построение программного кода соответствует с <<C++ Code Convention>>. К защите курсовой предоставляется: консольное приложение и пояснительная записка

Windows 10 – операционная система для персональных компьютеров и рабочих станций, разработанная корпорацией Microsoft в рамках семейства Windows NT. После Windows 8.1 система получила номер 10, минуя 9. Серверный аналог Windows 10 – Windows Server 2016. Система призвана стать единой для разных устройств, таких как персональные компьютеры, планшеты, смартфоны, консоли Xbox One и пр. Доступна единая платформа разработки и единый магазин универсальных приложений, совместимых со всеми поддерживаемыми устройствами. Windows 10 поставляется в качестве услуги с выпуском обновлений на протяжении всего цикла поддержки. В течение первого года после выхода системы пользователи могли бесплатно обновиться до Windows 10 на устройствах под управлением лицензионных копий Windows 7, Windows 8.1 и Windows Phone 8.1. Среди значимых нововведений – голосовая помощница Кортана, возможность создания и переключения нескольких рабочих столов и др. Windows 10 – последняя «коробочная» версия Windows, все последующие версии будут распространяться исключительно в цифровом виде.

## 1.5.Описывается требуемое оборудование и программное обеспечение

Для полноценного функционирования разработанного программного приложения, качественного выполнения всех поставленных задач, требуется ПК следующей конфигурации:

* процессор Intel Core i3;
* оперативная память DDR3 512 МБ или больше;
* минимально свободного места на жёстком диске 512 МБ;
* мышь;
* операционная система Windows 7/8/10.

# 2.ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРИЛОЖЕНИЯ

## 2.1. Алгоритм решения

В текущей главе представлено описание алгоритма, который имеет начало алгоритма и конец, внутри алгоритма имеются циклы и иные операций.

Вначале происходит смена кодировки консоли (Рис 2.1.1)

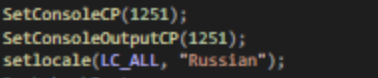


Рис 2.1.1

Далее происходит инициализация классов и списков классов. Далее происходит бесконечный цикл while. Он необходим для реализации повторного выбора операции.

Далее в программе происходит выбор операции по нажатию необходимой цифры.

1. - просмотреть список книг
2. - просмотреть список читателя
3. - добавить книгу

4 - добавить читателя

5 - изменить книгу

6 – изменить читателя

7 – добавить книгу читателю

1. – удалить книгу читателя

При помощи цикла for происходит перебор коллекции books. При каждой итерации цикла происходит вызов метода Print (рис 2.1.2)

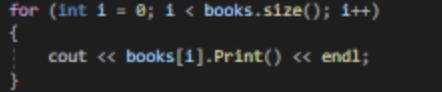


рис 2.1.2

Данный алгоритм работает схожим образом с первым, только перебор происходит коллекции readers (рис 2.1.3).

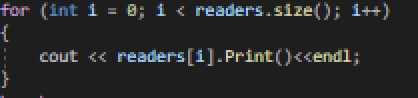


рис 2.1.3

В 3 и 4 пункте будет схожие алгоритмы. Происходит добавление книги книги или читателя. Путём вызова метода AddBook или AddReader к экземпляру класса и последующее его добавление в коллекция (books/reader). Добавление происходит путём поочерёдного присвоением полям класса данных(Рис 2.1.4)

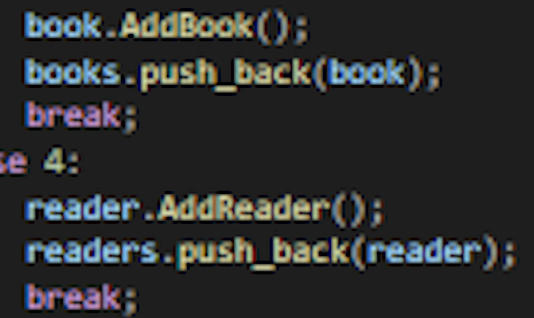


Рис 2.1.4

В пятом пункте происходит изменение книги (рис 2.1.5) путём выбора необходимого номера из списка. Далее происходит перебор коллекции и если элемент найден, то происходит вызов метода AddBlock к выбранному элементу коллекции.

Если же данного элемента нету, то происходит уведомление и введение значение повторно

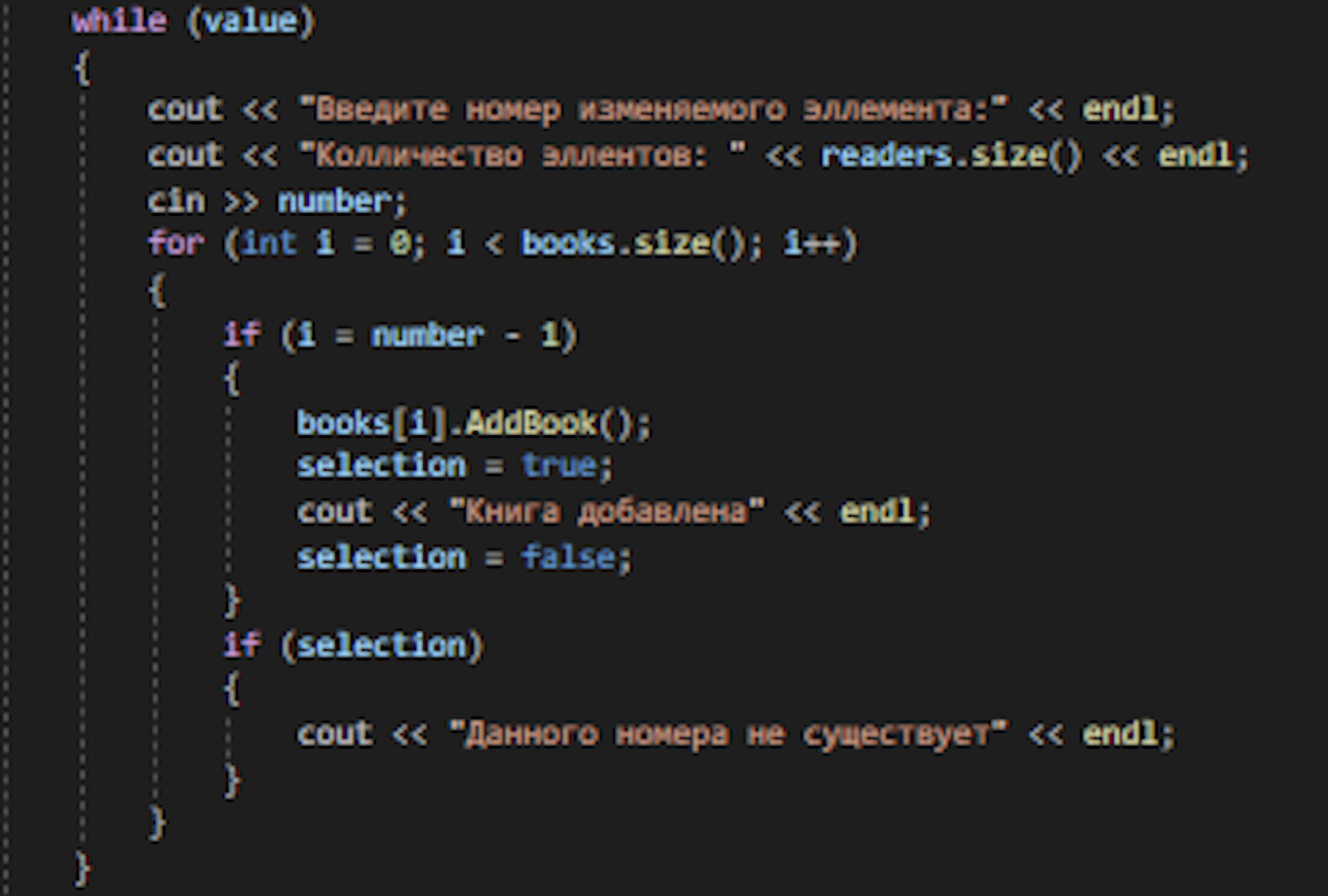


рис 2.1.5

В шестом пункте происходит изменение читателя (рис 2.1.6) путём выбора необходимого номера из списка. Далее происходит перебор коллекции и если элемент найден, то происходит вызов метода AddReader к выбранному элементу коллекции.

Если же данного элемента нету, то происходит уведомление и введение значение повторно

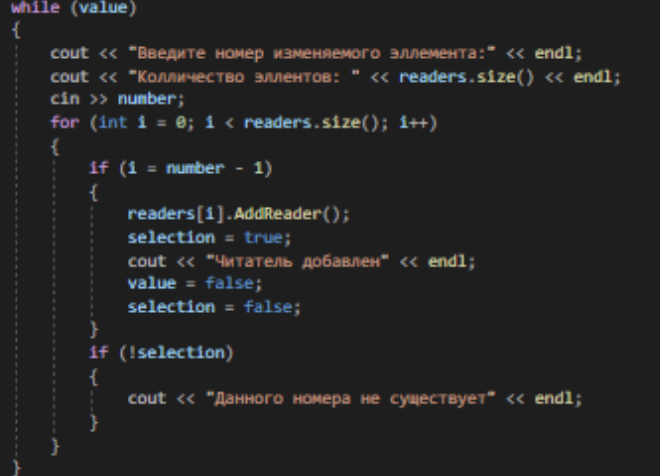


рис 2.1.6

В седьмом пункте происходит вначале перебор коллекции readers, а потом перебор коллекции books. При успешном нахождении обоих элементов данных коллекций происходит присвоение конкретному эллементу коллекции readers, конкретной книги из коллекции books (рис 2.1.7).

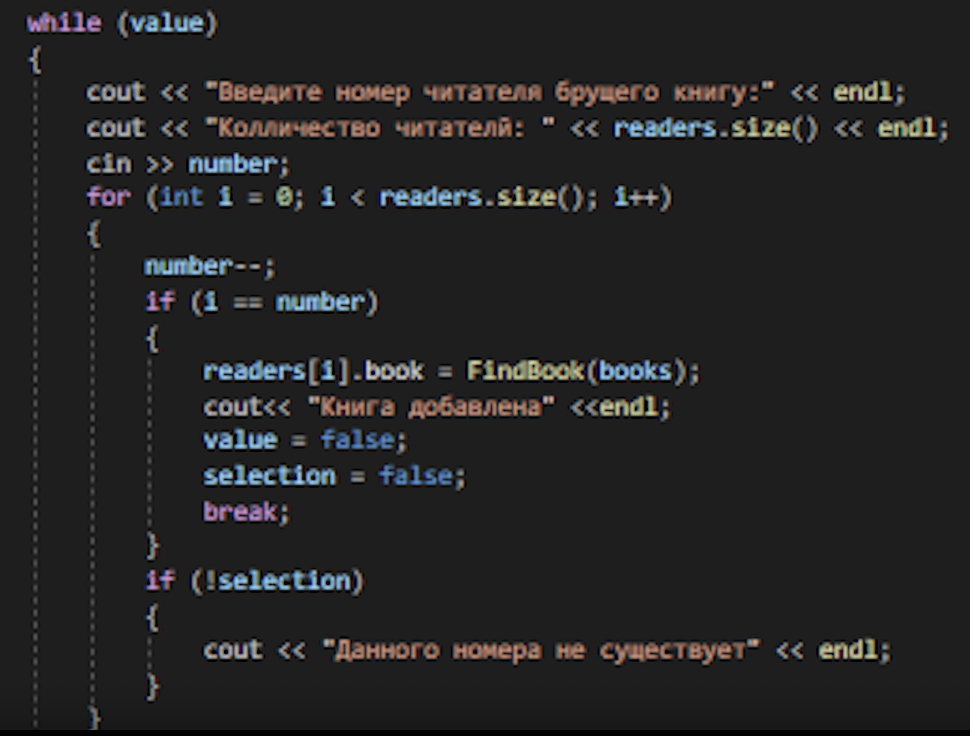


Рис 2.1.7

В восьмом пункте алгоритм построен похожим образом, за исключением того что происходит не присвоение конкретной книги, а происходит присвоение читателю пустого объекта класса book.

## 2.2. Описание программы

В данной программы используются следующие библиотеки:

#include <iostream> - основная библиотека, которая реализует основные возможность языка С++.

#include <string> - библиотека для возможности работы с типом данных string.

#include <deque> - данная библиотека необходима для реализации работы с двухсторонней очередью.

В данном программном продукте используется 3 класса book, Author, Reader.

Класс **Author** (рис. 2.21):

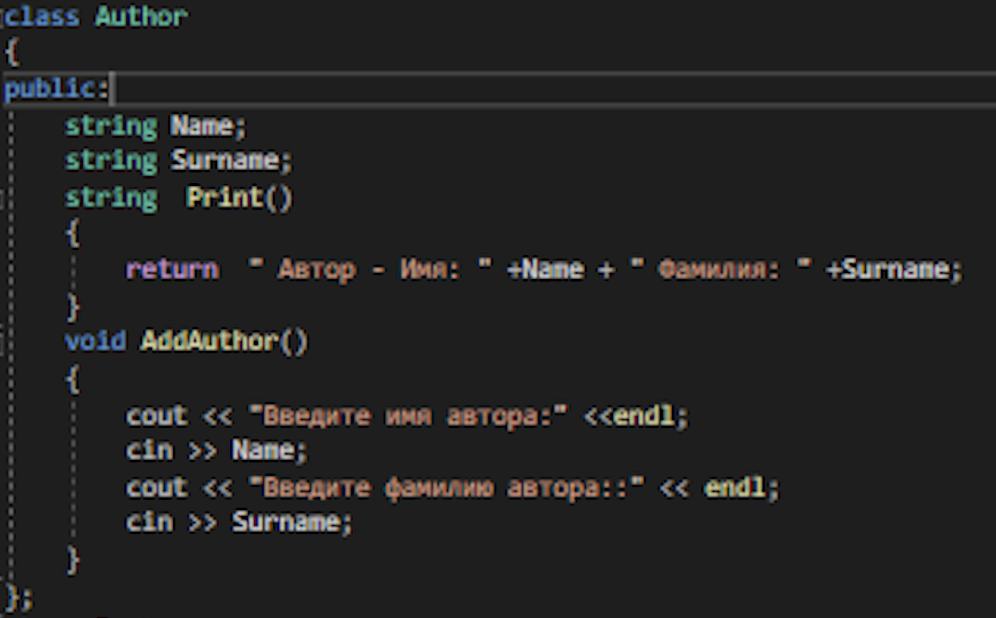


Рис. 2.21

Данный класс состоит из следующих переменных: Name (Имя автора), Surname (фамилия автора).

Методы:

string **Print()** – возвращает строку с данными корыте имеет объект класса.

void **AddAuthor()** -его смысл заключается в присвоении каждой переменной класса значения через консольный ввод.

Класс **Books**(рис 2.2.2)**:**

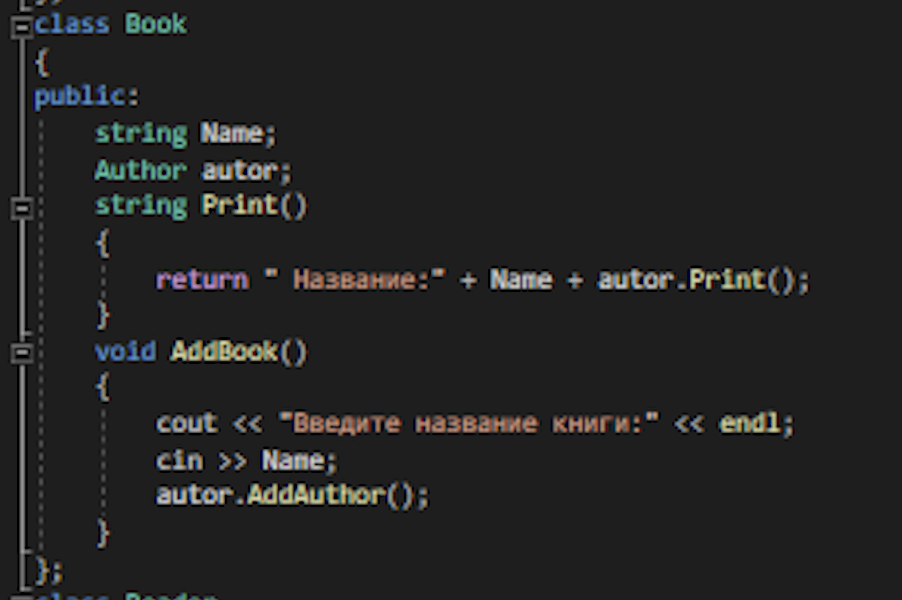
****

рис 2.2.2

Данный класс состоит из 2 переменных Name (Название книги), author (автор книги).

Методы:

string **Print()** – возвращает строку с данными корыте имеет объект класса.

void **AddAuthor()** -его смысл заключается в присвоении каждой переменной класса значения через консольный ввод.

Класс **Reader** (рис. 2.21):

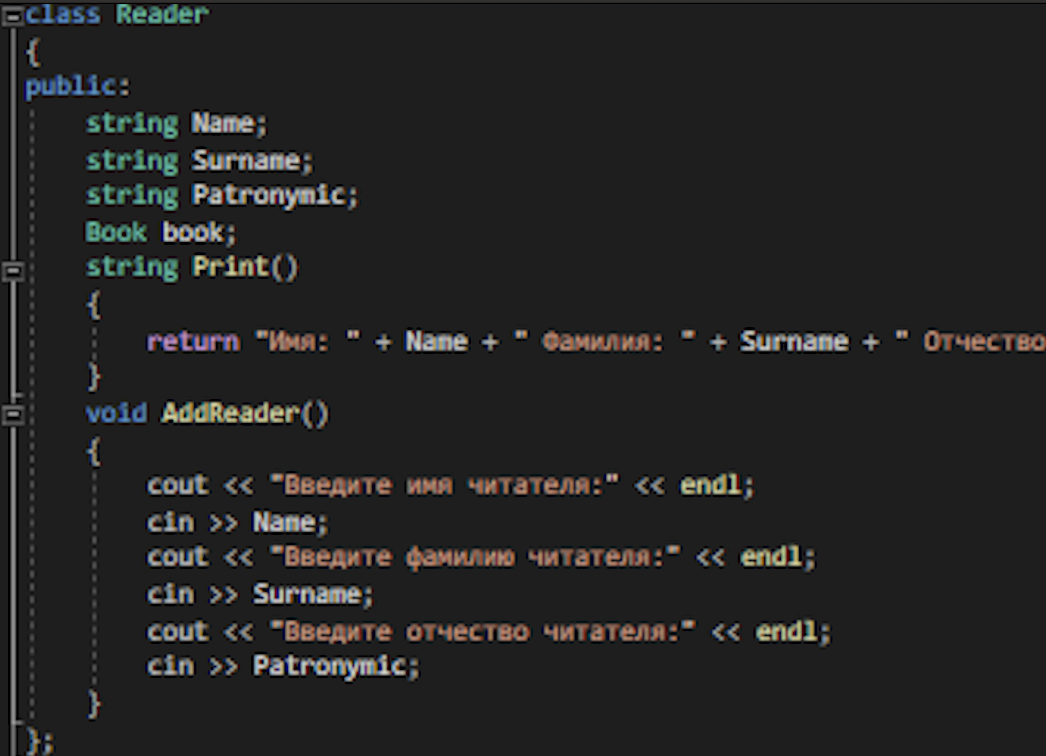


Рис. 2.21

Данный класс состоит из следующих переменных: Name (Имя читателя), Surname (фамилия читателя), Patronymic (отчество читателя).

Методы:

string **Print()** – возвращает строку с данными корыте имеет объект класса.

void **AddAuthor()** -его смысл заключается в присвоении каждой переменной класса значения через консольный ввод.

## 2.3. Результаты работы программы

Для описания данного я буду опираться на главу “Алгоритм программы” так как основная реализация программного продукта за счёт механизма switch.

1. просмотреть список книг (рис 2.3.1)



рис 2.3.1

1. просмотреть список читателя (рис 2.3.2)



рис 2.3.2

3- добавить книгу (рис 2.3.3)

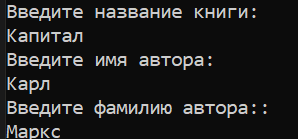


рис 2.3.3

1. - добавить читателя (рис 2.3.4)

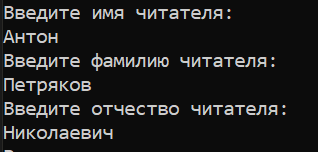


рис 2.3.4

1. - изменить книгу (рис 2.3.5)

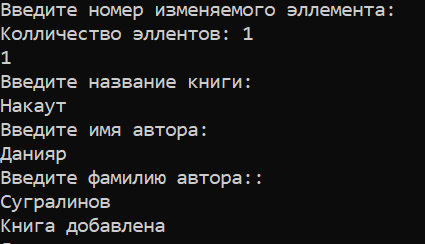


рис 2.3.5

1. – изменить читателя (рис 2.3.6)

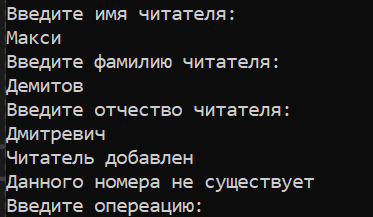


рис 2.3.6

1. – добавить книгу читателю (рис 2.3.7)

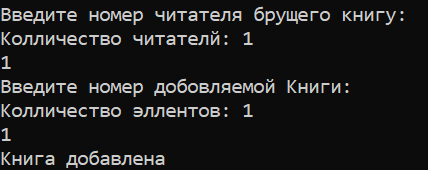


рис 2.3.7

8– удалить книгу читателя (рис 2.3.8)

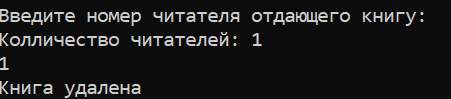


рис 2.3.8

# 3.РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Так как была поставлена задача разработать консольное приложение, а значит при разработке интерфейса возникают все проблемы для консольных приложений. Этот факт необходимо учитывать при разработке интерфейса.

При запуске программы на консоль выводиться выбор операции. Для осуществления выбора необходимо ввести соответствующую цифру в консоль.

При выборе пункта № 1 на консоль выводиться список книг.



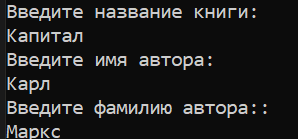
3.1

При выборе пункта № 2 на консоль выводиться список читателей.



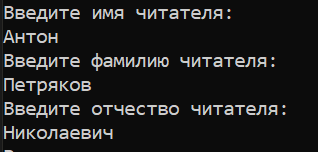
3.2

При выборе пункта № 3 в консоль надо ввести последовательно данные книги (Название, фамилия и имя автора)



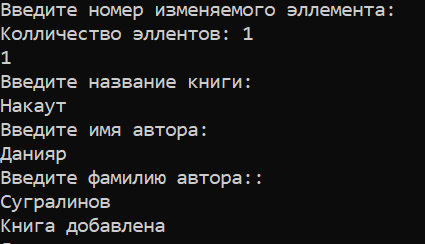
3.3

При выборе пункта № 4 в консоль надо ввести последовательно данные читателю (имя, фамилию отчество).



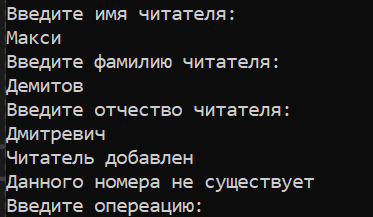
3.4

При выборе пункта № 5 в консоль необходимо внести номер изменяемой книги, а затем внести его новые данные



3.5

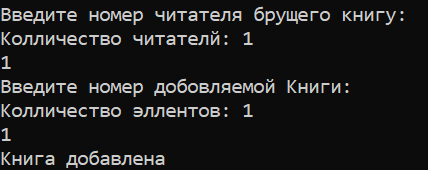
При выборе пункта № 6 в консоль необходимо внести номер изменяемого читателя, а затем внести его новые данные



3.6

При выборе пункта № 7 в консоль необходимо ввести номер читателя, которому будет добавляться книга, а затем номер добавляемое книге.

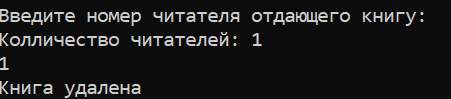
В случае если данные были введены неверно, то пользователю необходимо ввести данные повторно.



3.7

При выборе пункта № 8 в консоль необходимо ввести номер читателя, у которого будет удаляться книга.

В случае если данные были введены неверно, то пользователю необходимо ввести данные повторно.



3.8

# ЗАКЛЮЧЕНИЕ

Исходя из поставленных целей и задач было сконструировано, разработана и протестирован программный модуль «Учёт выдачи книг в библиотеки»,», предназначенный для использования библиотекарем.

Были реализованы задачи:

1. Учёт книг и читателей.
2. Изменение книг и читателей.
3. Просмотр книг и читателей.
4. Добавление книги читателю.
5. Удаление книги читателю.

Цель была выполнены, программный продукт имеет следующие преимущества:

1. Работа программного продукта происходит быстро и максимально эффективно. Программа удовлетворяет запросам пользователя.
2. Имеет возможность пополнять базу данных, для расширения книг и товаров.
3. Так же программа имеет высокую оптимизацию. Из-за этого программа работает весьма быстра и запускается даже на самом слабом устройстве.

Так же данный программный продукт выгодно отличается от других представленных на рынке, возможностью не только тем что можно производить учёт книг, но и взаимодействовать с читателем.

Таким образом программа работает без сбоев и все задачи были выполнены.

# СПИСОК ЛИТЕРАТУРЫ

1. Коплиен Дж. Мультипарадигменное проектирование для С++ / Коплиен Дж. – Питер, 2005.
2. Обзор обновлений и новых функций Windows 10 [Электронный ресурс]. – Microsoft, 2020. – Режим доступа: https://www.microsoft.com/ru-ru/windows/features. – Дата доступа: 22.05.2020.
3. Объектно-ориентированный анализ и проектирование с примерами приложений / Гради Буч [и др.]. – 3-е изд. – М.: ООО «И.Д. Вильямс», 2008. – 720 с.
4. Стивен Прата Язык программирования C++ / Стивен Прата – Вильямс, 2012.
5. Страуструп Б. Программирование. Принципы и практика с использованием C++ / Страуструп Б. – 2-е изд. – Вильямс, 2016.
6. Model-View-Presenter [Электронный ресурс]. – Википедия, 2020. – Режим доступа: https://ru.wikipedia.org/wiki/Model-View-Presenter. – Дата доступа: 22.04.2020.
7. SFML [Электронный ресурс]. – SFML, 2020. – Режим доступа: https://www.sfml-dev.org. – Дата доступа: 22.04.2020.

# ПРИЛОЖЕНИЕ

#include <iostream>

#include <string>

#include <deque>

#include <Windows.h>

using namespace std;

class Author

{

public:

string Name;

string Surname;

string Print()

{

return " Автор - Имя: " +Name + " Фамилия: " +Surname;

}

void AddAuthor()

{

cout << "Введите имя автора:" <<endl;

cin >> Name;

cout << "Введите фамилию автора::" << endl;

cin >> Surname;

}

};

class Book

{

public:

string Name;

Author autor;

string Print()

{

return " Название:" + Name + autor.Print();

}

void AddBook()

{

cout << "Введите название книги:" << endl;

cin >> Name;

autor.AddAuthor();

}

};

class Reader

{

public:

string Name;

string Surname;

string Patronymic;

Book book;

string Print()

{

return "Имя: " + Name + " Фамилия: " + Surname + " Отчество: " + Patronymic + "\n" + book.Print();

}

void AddReader()

{

cout << "Введите имя читателя:" << endl;

cin >> Name;

cout << "Введите фамилию читателя:" << endl;

cin >> Surname;

cout << "Введите отчество читателя:" << endl;

cin >> Patronymic;

}

};

Book FindBook(deque<Book> books)

{

bool selection = false;

int number;

while (true)

{

cout << "Введите номер добовляемой Книги:" << endl;

cout << "Колличество эллентов: " << books.size() << endl;

cin >> number;

for (int i = 0; i < books.size(); i++)

{

if (i == number - 1)

{

selection = true;

return books[i];

}

if (selection)

{

cout << "Данного номера не существует" << endl;

}

}

}

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setlocale(LC\_ALL, "Russian");

Book bookEmpty;

deque<Book> books;

deque<Reader> readers;

while (true)

{

int Number;

bool selection = false;

bool value = true;

Book book;

Reader reader;

cout << "Введите опереацию:" << endl;

cout << "1 - Просмотреть список книг" << endl;

cout << "2 - Просмотреть список читателей" << endl;

cout << "3 - Добавить книгу" << endl;

cout << "4 - Добавить читателя" << endl;

cout << "5 - Изменить книгу" << endl;

cout << "6 - Изменить читателя " << endl;

cout << "7 - Добавить книгу читателю" << endl;

cout << "8 - Удалить книгу читателя" << endl;

cin >> Number;

int number;

switch (Number)

{

case 1:

for (int i = 0; i < books.size(); i++)

{

cout << books[i].Print() << endl;

}

break;

case 2:

for (int i = 0; i < readers.size(); i++)

{

cout << readers[i].Print()<<endl;

}

break;

case 3:

book.AddBook();

books.push\_back(book);

break;

case 4:

reader.AddReader();

readers.push\_back(reader);

break;

case 5:

while (value)

{

cout << "Введите номер изменяемого эллемента:" << endl;

cout << "Колличество эллентов: " << books.size() << endl;

cin >> number;

for (int i = 0; i < books.size(); i++)

{

if (i == number -1)

{

books[i].AddBook();

selection = true;

cout << "Книга добавлена" << endl;

value = false;

}

if (selection)

{

cout << "Данного номера не существует" << endl;

}

}

}

break;

case 6:

while (value)

{

cout << "Введите номер изменяемого эллемента:" << endl;

cout << "Колличество эллентов: " << readers.size() << endl;

cin >> number;

for (int i = 0; i < readers.size(); i++)

{

if (i == number - 1)

{

readers[i].AddReader();

selection = true;

cout << "Читатель добавлен" << endl;

value = false;

}

if (selection)

{

cout << "Данного номера не существует" << endl;

}

}

}

break;

case 7:

while (value)

{

cout << "Введите номер читателя брущего книгу:" << endl;

cout << "Колличество читателй: " << readers.size() << endl;

cin >> number;

for (int i = 0; i < readers.size(); i++)

{

number--;

if (i == number)

{

readers[i].book = FindBook(books);

cout<< "Книга добавлена" <<endl;

value = false;

selection = false;

break;

}

if (!selection)

{

cout << "Данного номера не существует" << endl;

}

}

}

break;

case 8:

while (value)

{

cout << "Введите номер читателя отдающего книгу:" << endl;

cout << "Колличество читателей: " << readers.size() << endl;

cin >> number;

for (int i = 0; i < readers.size(); i++)

{

if (i == number - 1)

{

readers[i].book = bookEmpty;

cout << "Книга удалена" << endl;

selection = true;

value = false;

}

if (!selection)

{

cout << "Данного номера не существует" << endl;

}

}

}

break;

default:

cout << "Неверно выедена операция";

break;

}

}

}