Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Основы алгоритмизации и программирования

Отчёт

по учебной практике(ознакомительной)

на тему

Информационная база библиотеки

Студент И.А.Меша

Руководитель А.М.Ковальчук

МИНСК 2023

**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ .................................................................................................................

1. ЗАДАЧИ ПРОГРАММЫ. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ О РАЗРАБОТКЕ ПРОГРАММЫ………………………………………………………

2. ВХОДНЫЕ ДАННЫЕ.............................................................................................

3. ВЫХОДНЫЕ ДАННЫЕ..........................................................................................

4. РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ.....................................................

4.1. БЛОК-СХЕМА ГОЛОВНОГО МОДУЛЯ..........................................................

4.2. АЛГОРИТМ ПО ШАГАМ ФУНКЦИИ pop()…………………………………

4.3. АЛГОРИТМ ПО ШАГАМ ФУНКЦИИ addQueue()………………………….

4.4. СТРУКТУРА ХРАНЕНИЯ ДАННЫХ В ПРОГРАММЕ…………………….

5. КОД ПРОГРАММЫ...............................................................................................

5.1. main.cpp.................................................................................................................

5.2. functions.cpp...........................................................................................................

6. РЕЗУЛЬТАТЫ.........................................................................................................

ЗАКЛЮЧЕНИЕ ..........................................................................................................

СПИСОК ЛИТЕРАТУРЫ .........................................................................................

**ПРИЛОЖЕНИЕ А**....................................................................................................

**ПРИЛОЖЕНИЕ Б**....................................................................................................

**ПРИЛОЖЕНИЕ В**....................................................................................................

**ПРИЛОЖЕНИЕ Г**…………………………………………………………………

**ВВЕДЕНИЕ**

Язык C является одним из самых быстрых и популярных языков во всём мире. С позволяет осуществлять большинство преобразований типов. Контроль за выполнением этих преобразований, а также проверка некоторых ошибок (например, выход за границы массива) возлагается на программиста. Реализованная в С возможность напрямую манипулировать битами, байтами, словами и указателями необходима для программирования на системном уровне. Язык С считается структурированным языком. Отличительной чертой структурированного языка является разделение кода и данных. Одним из способов решения этой проблемы является использование подпрограмм (функций), широко использующих локальные переменные. Необходимо отметить, что излишнее использование глобальных переменных может приводить к фатальным ошибкам. Как и ряд других структурированных языков, С поддерживает ряд операторов цикла, условных операторов и операторов ветвления.

Язык С содержит стандартные библиотеки, предоставляющие функции, выполняющие наиболее типичные задачи. Эти библиотеки легко могут быть подключены, а также дополнены. Язык С позволяет разбивать программу на части и выполнять их раздельную компиляцию. Откомпилированные таким образом файлы объединяются для создания полного объектного кода. Преимущество раздельной компиляции в том, что при изменении одного файла не требуется перекомпиляции всей программы.

1. **ЗАДАЧИ ПРОГРАММЫ. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ О РАЗРАБОТКЕ ПРОГРАММЫ.**

Основными задачами данной программы является хранение информации о книгах в библиотеке, информации о зарегистрированных пользователях, а также реализации основных функций взаимодействия пользователя с книгами, а именно: взятие книг из библиотеки, возвращение книг в библиотеку, удаление информации о книге, удаление информации о пользователе, добавление пользователя в очередь (в случае отсутствия книги), присвоение статуса “должника” пользователю в случае не сдачи книги в день дедлайна, просмотр информации о пользователе и взятых им книг и т.д.

Для реализации данной программы были использованы следующие динамические структуры данных:

1. АВЛ-дерево – сбалансированное двоичное дерево поиска. Преимуществом АВЛ-дерева перед другими динамическими структурами данных является выполнение основных операций, таких как удаление, добавление и просмотр элемента за алгоритмическую сложность O(log(n)).
2. Двусвязный список - структура данных, состоящая из элементов одного типа, связанных между собой последовательно посредством указателей. В этом списке проще производить удаление и перестановку элементов, так как легко доступны адреса тех элементов списка, указатели которых направлены на изменяемый элемент.
3. Очередь – динамическая структура данных, работающая по принципу FIFO (первым пришел, первым вышел). Позволяет быстро удалять элемент в конце и добавлять элемент в начало.
4. **ВХОДНЫЕ ДАННЫЕ**

#define SIZE 25 //размер информационных полей

struct BookInfo{ //информация о книге

char category[SIZE]; //категория книги

char name[SIZE]; //название книги

char nameAuthor[SIZE]; //имя автора

int amount; //кол-во книг

};

struct AVLTreeBook{ //АВЛ-дерево (книги)

struct BookInfo data; //информация о книге

struct AVLTreeBook\* left; //указатель на левый узел дерева

struct AVLTreeBook\* right; //указатель на правый узел дерева

int height; //высота дерева

};

struct library{ //библиотека

struct AVLTreeBook\* ptr1; //указатель на корень дерева(книги)

struct AVLTreeUser\* ptr2; //указатель на корень дерева(пользователи)

struct queue\* ptr3; //указатель на очередь

};

struct time{

char takeTime[SIZE]; //время взятия книги

char endTime[SIZE]; //время сдачи книги

char currentTime[SIZE]; //текущее время

};

struct UserInfo{ //информация о пользователе

char name[SIZE]; //имя пользователя

char numPhone[15]; //номер пользователя

char numberCard[15]; //номер карточки пользователя

char address[SIZE]; //адрес проживания пользователя

bool debtor; //является ли пользователь должником

};

struct infoRegistration{ //список с информацией о регистрации

struct infoRegistration\* next; //указатель на следующий элемент

struct infoRegistration\* prev; //указатель на предыдущий элемент

struct dop{ // информация о книге и времени

struct BookInfo data; //информация о книге

struct time timeData; //информация о времени

}registration;

};

struct AVLTreeUser{ //АВЛ-дерево (пользователь)

struct UserInfo data; //информация о пользователе

struct infoRegistration\* head; //указатель на список книг

struct AVLTreeUser\* left; //указатель на левый узел дерева

struct AVLTreeUser\* right; //указатель на правый узел дерева

int height;

};

struct queue{ //очередь

struct UserInfo infU; //информация о пользователе

struct BookInfo infB; //информация о книге

int amount; //кол-во книг

int flag; //хватает ли книг для выдачи

struct queue\* next; //указатель на следующий элемент

struct queue\* prev; //указатель на предыдущий элемент

};

1. **ВЫХОДНЫЕ ДАННЫЕ**

Выходными данными являются три бинарных файла:

users.bin – файл с информацией о зарегистрированных пользователях в библиотеке

books.bin – файл с информацией о книгах в библиотеке

list.bin – файл с очередью за книгами в библиотеке

1. **РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ**

**4.1. БЛОК-СХЕМА ГОЛОВНОГО МОДУЛЯ:**

Блок-схема головного модуля представлен в **ПРИЛОЖЕНИИ А.**

* 1. **. АЛГОРИТМ ПО ШАГАМ ФУНКЦИИpop():**

Алгоритм по шагам представлен в **ПРИЛОЖЕНИИ Б.**

* 1. **. АЛГОРИТМ ПО ШАГАМ ФУНКЦИИ addQueue ():**

Алгоритм по шагам представлен в **ПРИЛОЖЕНИИ В.**

* 1. **. СТРУКТУРА ХРАНЕНИЯ ДАННЫХ В ПРОГРАММЕ:**

Структура хранения данных в программе представлена в **ПРИЛОЖЕНИИ Г.**

**ЗАКЛЮЧЕНИЕ**

В результате выполнения учебной практики были выполнены первоначально заданные цели, реализованы основные функции взаимодействия пользователя с книгами, закреплены теоретические и практические знания в применении динамических структур данных.

На сегодняшний день существует огромное количество баз данных, которые хранят и анализируют информацию, а также позволяют пользователям взаимодействовать с ней. Данная программа предназначена для таких же целей.

**ПРИЛОЖЕНИЕ А**

(обязательное)

**ПРИЛОЖЕНИЕ Б**

(обязательное)

**ПРИЛОЖЕНИЕ В**

(обязательное)

**ПРИЛОЖЕНИЕ Г**

(обязательное)