Министерство образования и науки Российской Федерации

Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский**

**политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №8**

Дисциплина: «Информатика»

Тема: Структуры данных

Вариант 18

Выполнил:

Студент группы РИС-20-1б

Тараканов Д. М.

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

Пермь

2021 год

# **Цель работы**

Научится работе с двоичными файлами, организацией ввода-вывода структурированной информации и ее хранение на внешних носителях.

# **Постановка задачи**

Задача состоит в разработке программы, а также создание в ней структуры “Книга” с полями название, автор, год издания, количество страниц. Заполнить эту структуру данными. Программа производит удаление 3-ёх элементов из начала файла, а затем добавляет элемент перед элементом с указанным названием.

# **Анализ задачи**

1. Определить какие предстоит выполнить действия:

* Реализовать поиск по названию.
* Реализовать добавление элемента перед элементом с заданным названием.

2. С какими типами данных действие надо сделать, в каком виде эти данные будут представлены, какие будут использованы поля в структуре под эти данные:

* Для названия создано поле типа string.
* Для автора создано поле типа string.
* Для года издания создано поле типа int.
* Для количества страниц создано поле типа int.

struct KNIGA{

string name; //Название книги

string author; //Автор книги

int year; //Год публикации

int pages; //Кол-во страниц

};

3. Разработка структуры KNIGA и её объявление:

* Для решения задачи разработана структура на 100 элементов

KNIGA book[100]; //Создаём структуру KNIGA на 100 элементов

4. Ввод данных в структуру реализован через консоль при помощи цикла while, также в цикле реализован проверка на ввод данных в поля год издания и количество страниц.

* Стоит отметить что название книги может состоять из более чем одного слова поэтому для заполнения этого поля разработана функция getline().

getline(cin, book[i].name);

1. При повторном вызове getline() не дожидается ввода строки, а считывает последний введённый символ в потоке (символ перевода строки в нашем случаи) поэтому была использована функция cin.get() чтоб удалить этот символ.

cin.get(); //Удаление последнего введенного символа в потоке

5. Вывод данных из структуры в консоль реализован функцией out.

void out(struct KNIGA book[], int N, int i = 1) {

while (i < N) {

cout << "\nНазвание книги: " << book[i].name << "\nАвтор книги: " << book[i].author << "\nГод издания: " << book[i].year << "\nКол-во страниц: " << book[i].pages << endl;

i++;

}

};

6. Для иллюзии удаления первых 3-ёх элементов структуры book создана дополнительная структура kniga, также размером на 100 элементов, которая содержит в себе элементы структуры book начиная с 4-ого.

KNIGA kniga[100]; //Создаём дополнительную структуру на 100 элементов

* Удаление элементов реализовано циклом for.

for (i = 4; i < N; i++) {

kniga[j].name = book[i].name;

kniga[j].author = book[i].author;

kniga[j].year = book[i].year;

kniga[j].pages = book[i].pages;

j++;

}

7. Поиск по названию в структуре реализован при помощи цикла while, для программы требуется найти номер искомой книги.

* Также, как и в случаи заполнения данных в структуру используются функции getline() и cin.get().

cout << "\nУкажите название книги перед которой вы хотите вставить новый элемент.\n";

cin.get();

getline(cin, answer);

while ((answer != kniga[n].name) && (n < N)) {

n++;

}

8. После нахождения номера искомой книги программа совершает сдвиг структуры вправо. И затем в это освободившееся место с консоли вводятся новый элемент.

* Стоит отметить что в программе предусмотрена возможность случая, когда искомая книга не окажется в дополнительной структуре kniga, в таком исходе новый элемент добавляется в конец структуры.

if (n>=N){

cout << "\nВведенная вами книга в структуре отсутствует либо удалена.\n";

}

else {

//Сдвиг структуры вправо от искомой

for (i = N; i >= n; i--) {

kniga[i].name = kniga[i - 1].name;

kniga[i].author = kniga[i - 1].author;

kniga[i].year = kniga[i - 1].year;

kniga[i].pages = kniga[i - 1].pages;

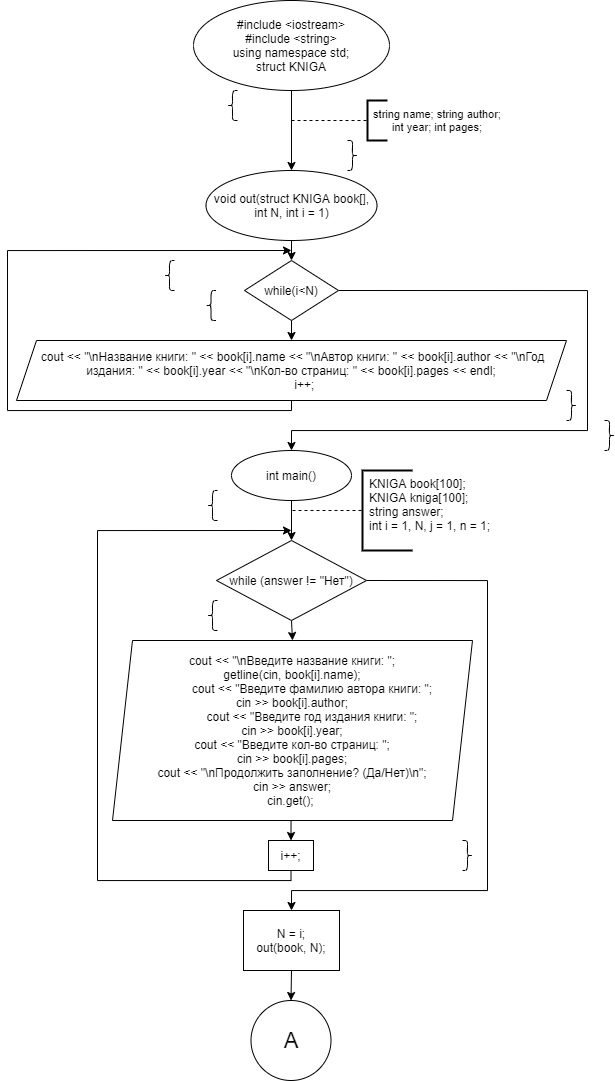
}

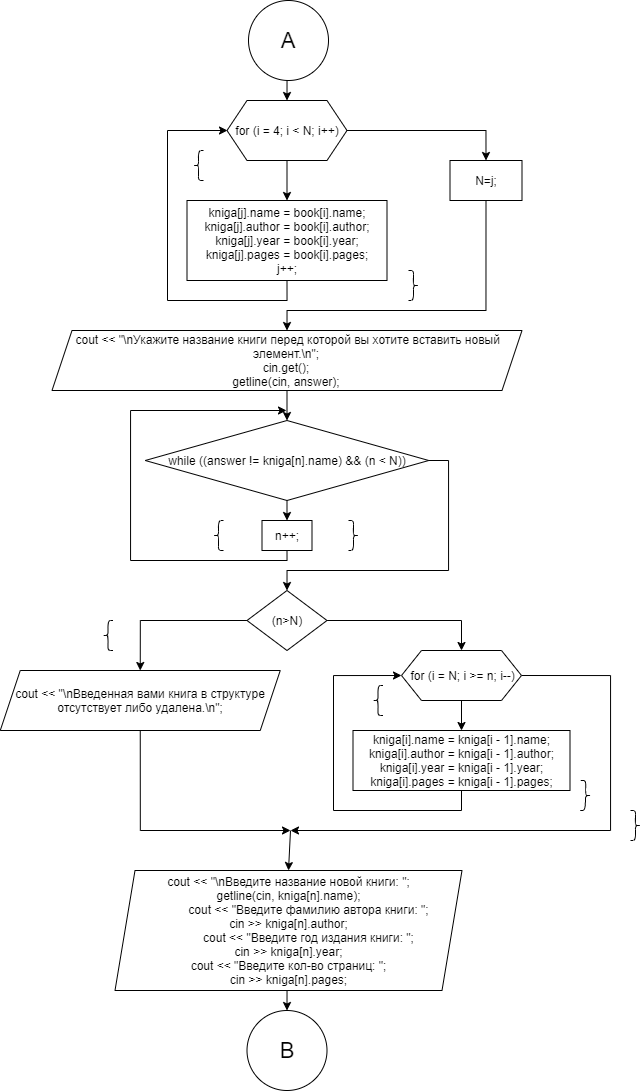
}

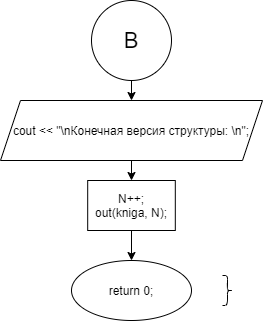
9. Последний шаг в работе программы — это вывод изменённой по заданию изначальной структуры book в консоль.

* Вывод реализуется функцией out.

**Блок схема**







# **Код программы на языке C++**

#include <iostream>

#include <string>

using namespace std;

struct KNIGA{

string name; //Название книги

string author; //Автор книги

int year; //Год публикации

int pages; //Кол-во страниц

};

//Функция вывода элементов структуры

void out(struct KNIGA book[], int N, int i = 1) {

while (i < N) {

cout << "\nНазвание книги: " << book[i].name << "\nАвтор книги: " << book[i].author << "\nГод издания: " << book[i].year << "\nКол-во страниц: " << book[i].pages << endl;

i++;

}

};

int main()

{

system("chcp 1251");

KNIGA book[100]; //Создаём структуру KNIGA на 100 элементов

KNIGA kniga[100]; //Создаём дополнительную структуру на 100 элементов

string answer;

int i = 1, N, j = 1, n = 1;

//Ввод элементов в структуру с проверкой правильности введенных данных

cout << "\nХотите ввести элементы вручную? (Да/Нет)\n";

cin >> answer;

if (answer != "Нет") {

while (answer != "Нет") {

cout << "\nВведите название книги: ";

getline(cin, book[i].name);

cout << "Введите фамилию автора книги: ";

cin >> book[i].author;

cout << "Введите год издания книги: ";

do {

cin >> book[i].year;

if ((book[i].year > 2021) || (book[i].year < 1)) {

cout << "Введён некорректный год! \nВедите год издания книги: ";

}

} while ((book[i].year > 2021) || (book[i].year < 1));

cout << "Введите кол-во страниц: ";

do {

cin >> book[i].pages;

if (book[i].pages < 5) {

cout << "Введено некорректное кол - во страниц! \nВедите кол-во страниц: ";

}

} while (book[i].pages < 5);

i++;

cout << "\nПродолжить заполнение? (Да/Нет)\n";

cin >> answer;

cin.get(); //Удаление последнего введенного символа в потоке

};

}

else {

book[1] = { "Горе от ума", "Грибоедов", 1825, 98 };

book[2] = { "Евгений Онегин", "Пушкин", 1825, 400 };

book[3] = { "Герой нашего времени", "Лермонтов", 1840, 423 };

book[4] = { "Мертвые души", "Гоголь", 2011, 352 };

book[5] = { "Приключения Шерлока Холмса", "Дойль", 1892, 307 };

book[6] = { "Мастер и Маргарита", "Булгаков", 1966, 504 };

i = 7;

}

// Вывод элементов структуры после заполнения при помощи функции

N = i;

out(book, N);

//Удаление первых 3-ёх элементов структуры

for (i = 4; i < N; i++) {

kniga[j].name = book[i].name;

kniga[j].author = book[i].author;

kniga[j].year = book[i].year;

kniga[j].pages = book[i].pages;

j++;

}

//Поиск по названию книги

N = j;

cout << "\nУкажите название книги перед которой вы хотите вставить новый элемент.\n";

cin.get();

getline(cin, answer);

while ((answer != kniga[n].name) && (n < N)) {

n++;

}

if (n>N){

cout << "\nВведенная вами книга в структуре отсутствует либо удалена.\n";

}

else {

//Сдвиг структуры вправо от искомой

for (i = N; i >= n; i--) {

kniga[i].name = kniga[i - 1].name;

kniga[i].author = kniga[i - 1].author;

kniga[i].year = kniga[i - 1].year;

kniga[i].pages = kniga[i - 1].pages;

}

}

cout << "\nВведите название новой книги: ";

getline(cin, kniga[n].name);

cout << "Введите фамилию автора книги: ";

cin >> kniga[n].author;

cout << "Введите год издания книги: ";

do {

cin >> kniga[n].year;

if ((kniga[n].year > 2021) || (kniga[n].year < 1)) {

cout << "Введён некорректный год! \nВведите правильный год издания книги: ";

}

} while ((kniga[n].year > 2021) || (kniga[n].year < 1));

cout << "Введите кол-во страниц: ";

do {

cin >> kniga[n].pages;

if (kniga[n].pages < 5) {

cout << "Введено некорректное кол - во страниц! \nВведите правильное кол - во страниц: ";

}

} while (kniga[n].pages < 5);

//Вывод результата работы программы

N++;

cout << "\nКонечная версия структуры: \n";

out(kniga, N);

cout << endl;

system("pause");

return 0;

}

# **Скриншоты тестов**

