Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный** **исследовательский политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

направление подготовки: 09.03.01– «Информатика и вычислительная техника»

**Лабораторная работа № 8**

**по дисциплине**

**«Основы алгоритмизации и программирования»**

**на тему**

**«Блоковый ввод-вывод в C++»**

Выполнил студент гр. ЭВТ-22-1бзу

Иванов Александр Станиславович

Проверил:

доцент каф. ИТАС

Полякова О.А.

(оценка) (подпись)

(дата)

Пермь 2022

**Задачи работы**

Цель: Работа с двоичными файлами, организация ввода-вывода структурированной информации и ее хранение на внешних носителях.

Сформировать двоичный файл из элементов, заданной в ва-рианте структуры, распечатать его содержимое, выполнить удаление и добавление элементов в соответствии со своим вариантом, используя для поиска удаляемых или добавляе-мых элементов функцию. Формирование, печать, добавление и удаление элементов оформить в виде функций. Преду-смотреть сообщения об ошибках при открытии файла и вы-полнении операций ввода/вывода.

Структура "Государство":

- название;

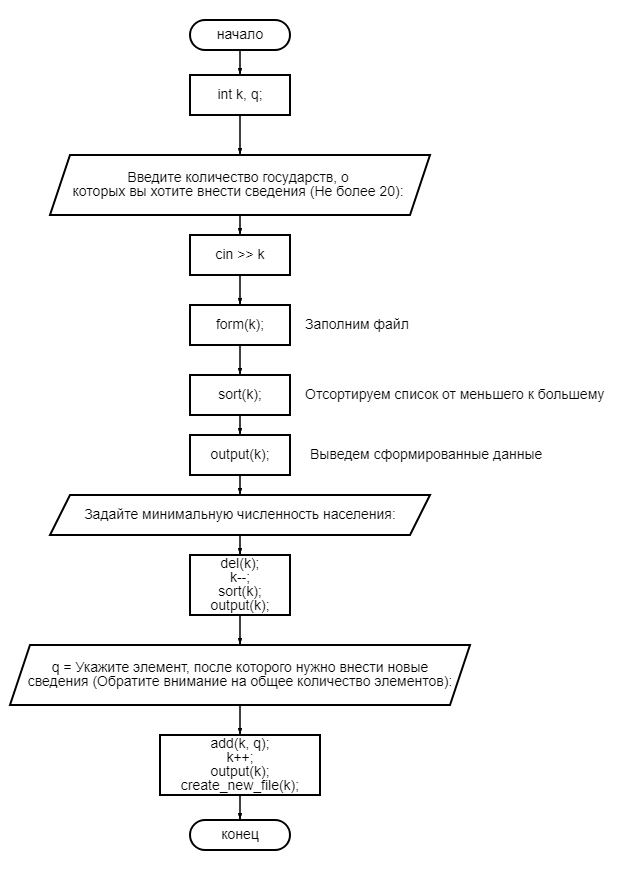
- столица;

- численность населения;

- занимаемая площадь.

Удалить все элементы, у которых численность меньше заданной, добавить элемент после элемента с указанным номером.

**Блок-Схема**

****

**Код**

// Задание №8, Вариант 3.

#include <iostream>

#include <stdio.h> // Для структуры предопределенного типа FILE

#include <windows.h> // Библиотека для русской локализации

#include <typeinfo>

struct State

{

char name[35]; // Название

char capital[35]; // Столица

int population; // Численность населения

float area; // Занимаемая площадь

};

FILE\* pf; // Первый изначальный файл

FILE\* pf2; // Второй измененный файл

State a, Arr[20]; // Структура, массив структуры

// Объявляем методы(функции)

void form(int);

void sort(int);

void output(int);

void del(int);

void create\_new\_file(int);

void add(int, int);

int main()

{

SetConsoleCP(1251); // Ввод с консоли в кодировке 1251 для русской локализации.

SetConsoleOutputCP(1251); // Вывод на консоль в кодировке 1251 для русской локализации.

std::cout << "Блоковый ввод-вывод\n";

int k, q;

//формирование записей в файле

std::cout << "\n\nВведите количество государств, о которых вы хотите внести сведения (Не более 20): ";

std::cin >> k;

if (k <= 0 || k > 20)

{

std::cout << "Вы не указали число, попробуйте еще раз!";

exit(1);

}

// заполним файл

form(k);

//отсортируем

sort(k);

//выведем сформированные данные

output(k);

// Удалить элемент, номер вводится с клавиатуры.

std::cout << "Задайте минимальную численность населения: \n";;

del(k);

k--; // Уменьшаем кол-во блоков(структур)

sort(k);

//create\_new\_file(k);

output(k);

//добавляем элемент после указанной фамилии

std::cout << "Укажите элемент, после которого нужно внести новые сведения (Обратите внимание на общее количество элементов): ";

std::cin >> q;

add(k, q);

k++; // Добавляем новый элемент в структуру

output(k);

create\_new\_file(k); //запишем в файл новые данные

return 0;

}

void form(int k)

{

//открытие файла

if ((pf = fopen("test.dat", "wb")) == NULL) // Создаем двоичный файл для записи.

{

std::cout << "\nОшибка при открытии файла";

exit(1);

}

// Формирование записей в файле

std::cout << "\nВведите сведения о " << k << " государствах \n";

for (int i = 0; i < k; i++)

{

std::cout << "Введите название государства: \n";

std::cin >> a.name;

std::cout << "Введите столицу: \n";

std::cin >> a.capital;

std::cout << "Введите численность населения: \n";

std::cin >> a.population;

std::cout << "Введите занимаемую площадь: \n";

std::cin >> a.area;

std::cout << "\n";

fwrite(&a, sizeof(State), 1, pf); // Записываем прямо в структуру

if (ferror(pf)) exit(1);

}

fclose(pf);

//считывание информации с файла

if ((pf = fopen("test.dat", "rb")) == NULL) exit(1); // Открываем двоичный файл для чтения.

int j = 0;

while (!feof(pf) && j < k) // Считываем бинарный файл до самого конца

{

fread(&Arr[j], sizeof(State), 1, pf);

j++;

}

fclose(pf);

return;

}

void sort(int k) // Функция сортирует структуру от меньшего к большему, сделано для того чтобы структуры корректно взаимозаменилсь между собой при удалении(В с++ нельзя явно удалить элементы массива, поэтому я делаю замещение к примеру 2 массива верного на 1 не верный и убираю 2 массив),

// и добавлении новых элементов

{

State tmp;

for (int i = k - 1; i >= 0; i--)

{

for (int j = 0; j < i; j++)

{

// сравниваем элементы массива структур по сумме баллов студента

if (Arr[j].population > Arr[j + 1].population)

{

tmp = Arr[j];

Arr[j] = Arr[j + 1];

Arr[j + 1] = tmp;

}

}

}

}

void output(int k) // Выводим данные с файла

{

std::cout << "\nДанные о " << k << " Государствах\n";

for (int i = 0; i < k; i++)

{

std::cout << "\nНазвание государства: " << Arr[i].name << "\n";

std::cout << "Столица: " << Arr[i].capital << "\n";

std::cout << "Численность населения: " << Arr[i].population << "\n";

std::cout << "Занимаемая площадь: " << Arr[i].area << "\n";

std::cout << "\n";

}

return;

}

void create\_new\_file(int k) // Формируем новый измененный файл

{

if ((pf2 = fopen("new\_test.dat", "wb")) == NULL) // Создаем двоичный файл для записи.

{

std::cout << "\nОшибка при открытии файла";

exit(1);

}

for (int i = 0; i < k; i++)

{

fwrite(&Arr[i], sizeof(State), 1, pf); // Записываем в созданную структуру, но заменив значения

if (ferror(pf)) exit(1);

}

fclose(pf);

//считывание информации с файла

if ((pf = fopen("new\_test.dat", "rb")) == NULL) exit(1); // Открываем двоичный файл для чтения.

int j = 0;

while (!feof(pf2) && j < k)

{

fread(&Arr[j], sizeof(State), 1, pf2);

j++;

}

fclose(pf2);

return;

}

void del(int k) // Удаляем необходимый элемент(-ы)

{

int u;

std::cin >> u; // Переменная забирает значение из метода main из коносоли.

for (int i = 0; i < k; i++)

{

if (Arr[i].population <= u)

{

for (int j = i; j + 1 < k; ++j) // Удаляю только тот массив, который не подошел под условие, Остольные не трогаю.

{

strcpy(Arr[j].name, Arr[j + 1].name);

strcpy(Arr[j].capital, Arr[j + 1].capital);

Arr[j].population = Arr[j + 1].population;

Arr[j].area = Arr[j + 1].area;

}

}

}

return;

}

void add(int k, int q)

{

// q - номер элемента

if (q > k)

{

std::cout << "\nТакого элемента в списке нет";

}

else

{

for (int i = q; i > k; i--) // Заменяем предыдущий элемент

{

strcpy(Arr[i].name, Arr[i - 1].name);

strcpy(Arr[i].capital, Arr[i - 1].capital);

Arr[i].population = Arr[i - 1].population;

Arr[i].area = Arr[i - 1].area;

}

// Вставляем данные после выбранного массива

std::cout << "Введите название государства: \n";

std::cin >> Arr[q].name;

std::cout << "Введите столицу: \n";

std::cin >> Arr[q].capital;

std::cout << "Введите численность населения: \n";

std::cin >> Arr[q].population;

std::cout << "Введите занимаемую площадь: \n";

std::cin >> Arr[q].area;

std::cout << "\n";

}

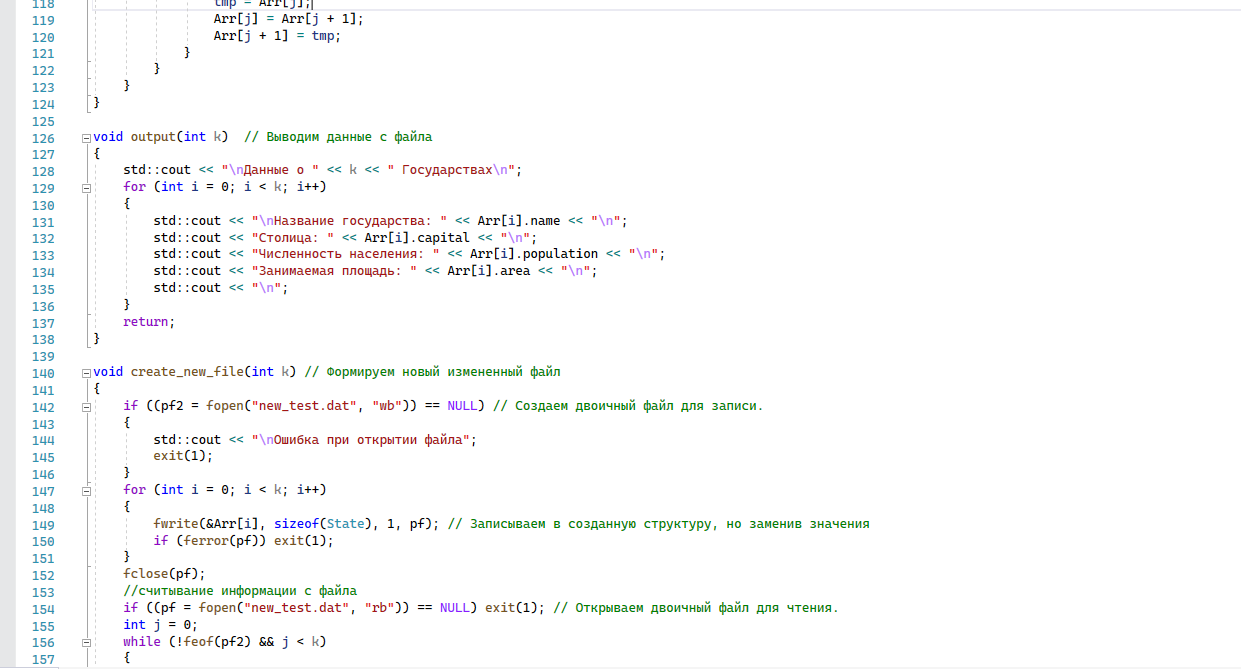
return;

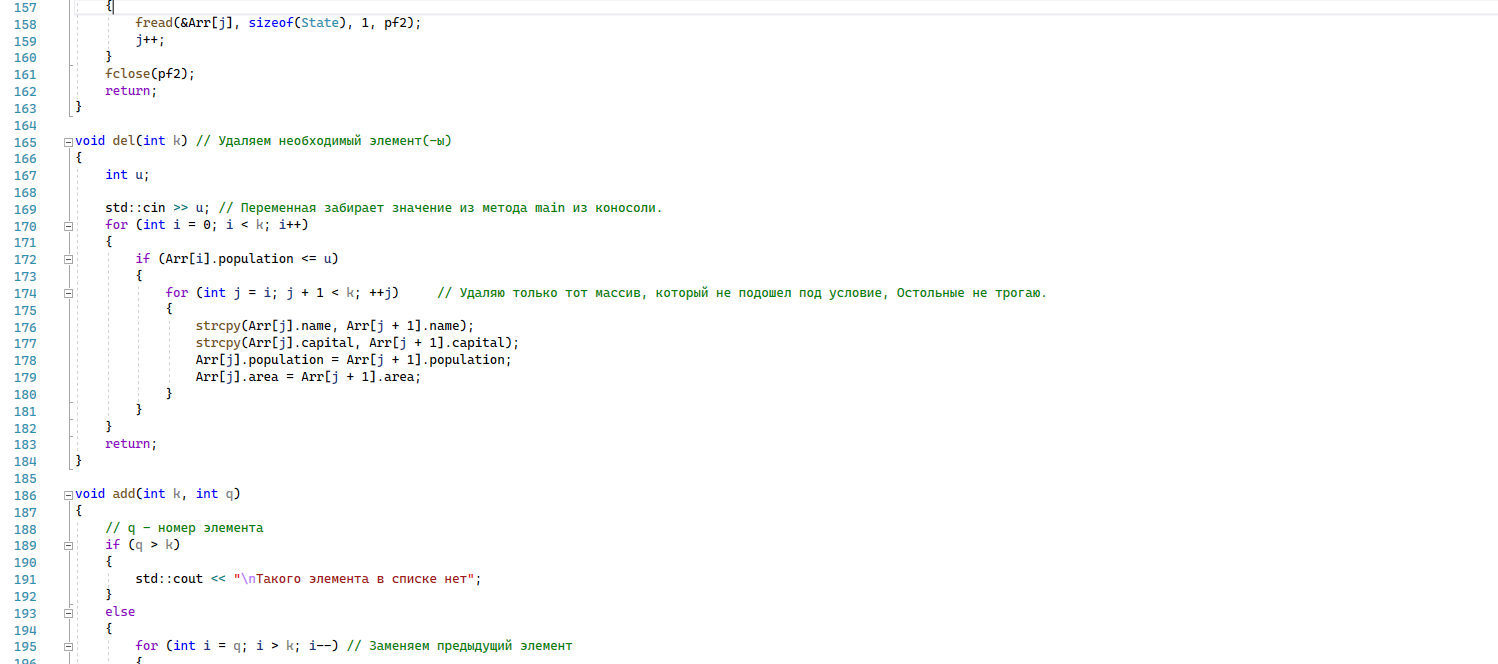
}

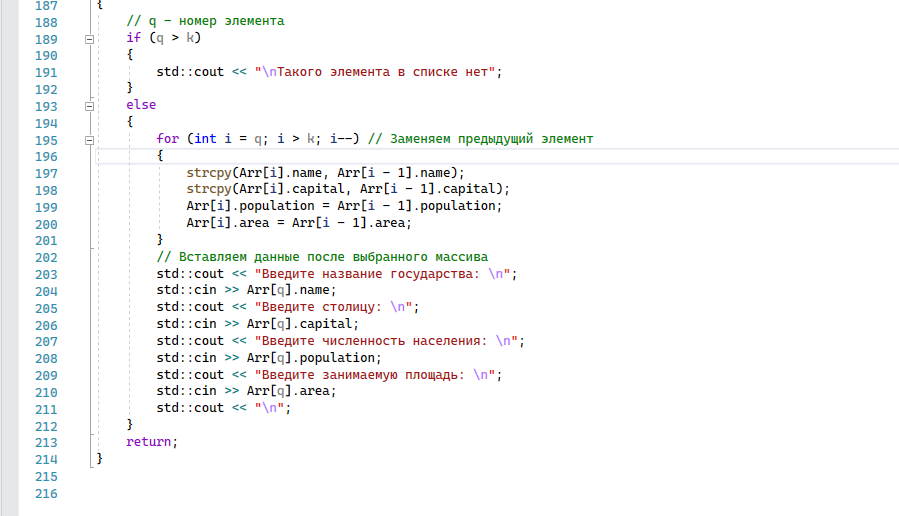
**Скриншоты**

****

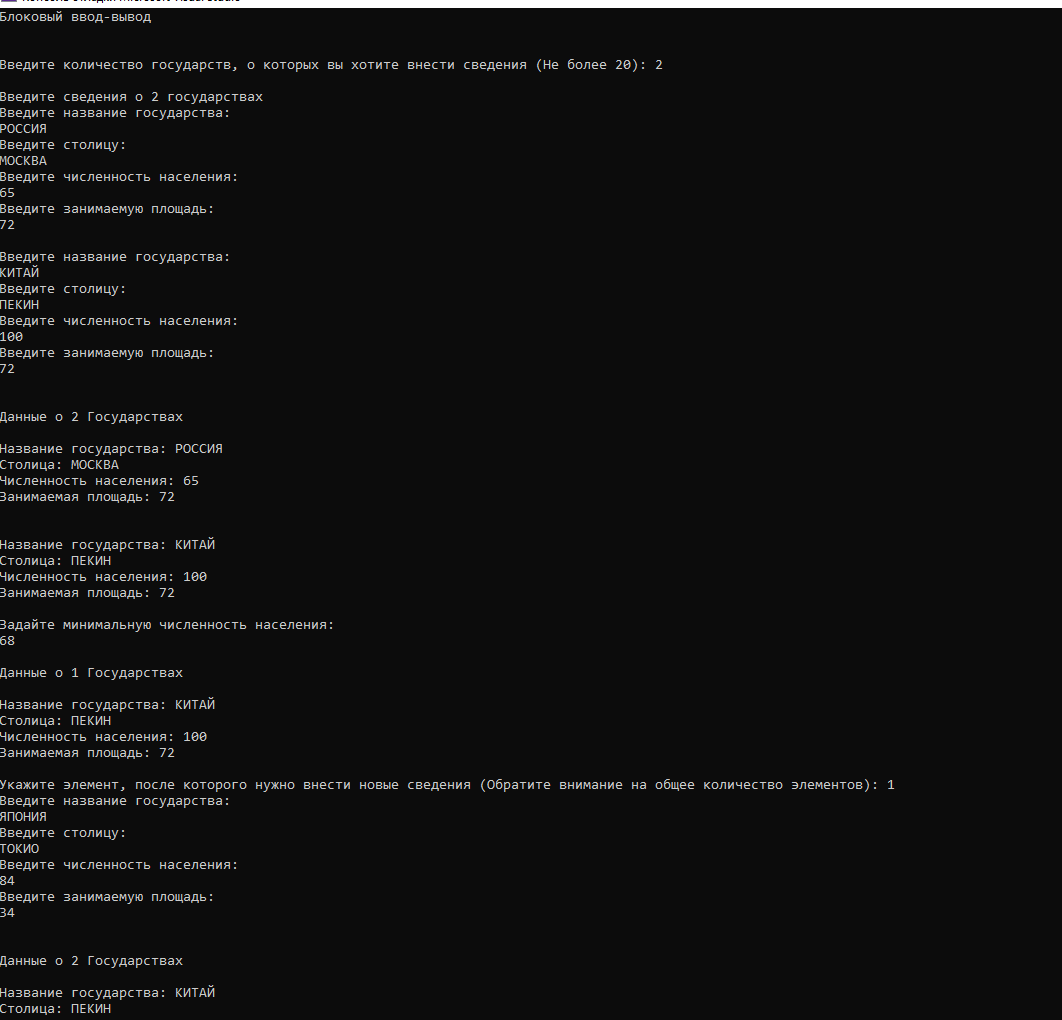
****

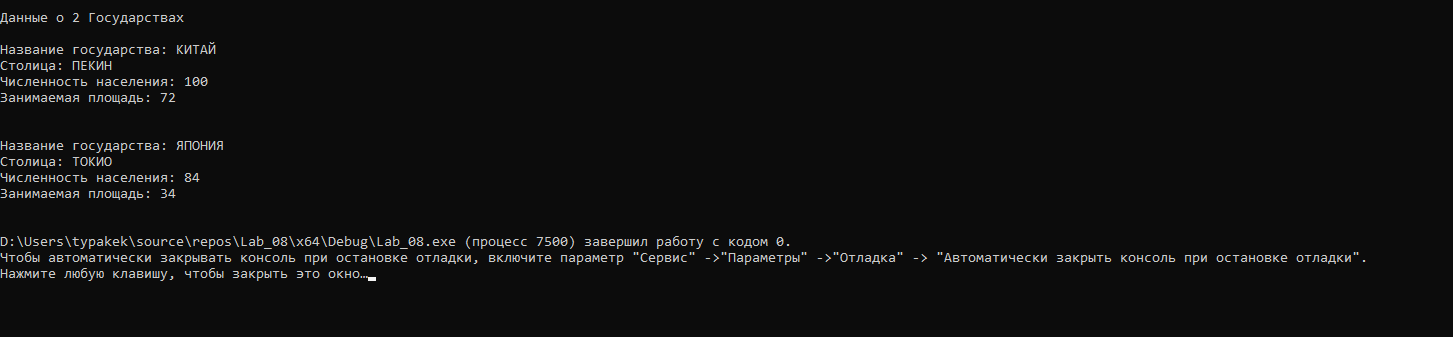
****

****

****

**Результат выполнения**

****

****