Пермский Национальный Исследовательский Политехнический Университет

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

Лабораторная работа № 8

по дисциплине

«Основы алгоритмизации и программирования»

Тема: «Блоковый ввод-вывод»

Вариант № 17

Выполнил:

студент группы РИС-20-2б

Пономарев Егор Витальевич

Проверила:

доцент кафедры ИТАС

Полякова О. А.

Пермь, 2021

**Цель работы**

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

**Постановка задачи**

Сформировать двоичный файл из элементов, заданной в варианте структуры, распечатать его содержимое, выполнить удаление и добавление элементов в соответствии со своим вариантом, используя для поиска удаляемых или добавляемых элементов функцию. Формирование, печать, добавление и удаление элементов оформить в виде функций. Предусмотреть сообщения об ошибках при открытии файла и выполнении операций ввода/вывода.

Структура «Фильм»:

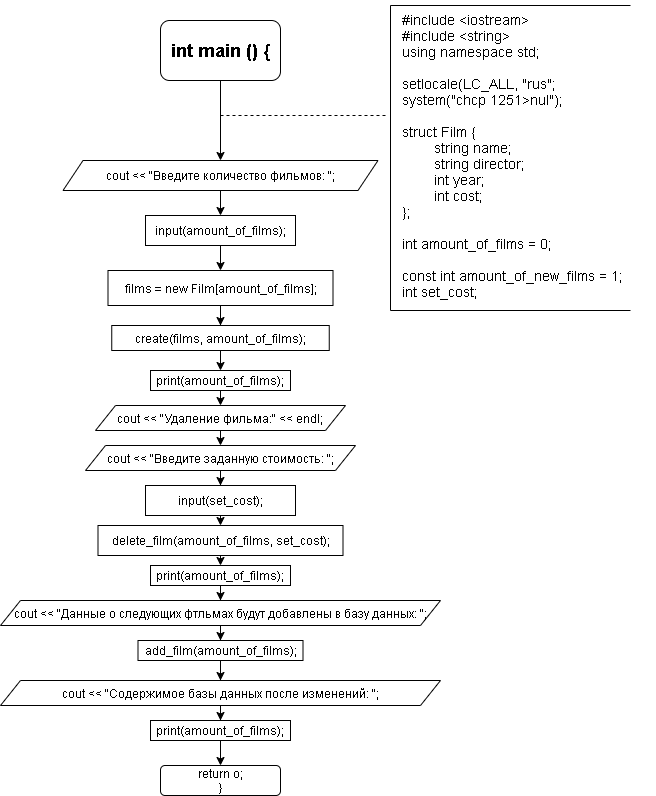
* Название;
* Режиссер;
* Год выпуска;
* Стоимость.

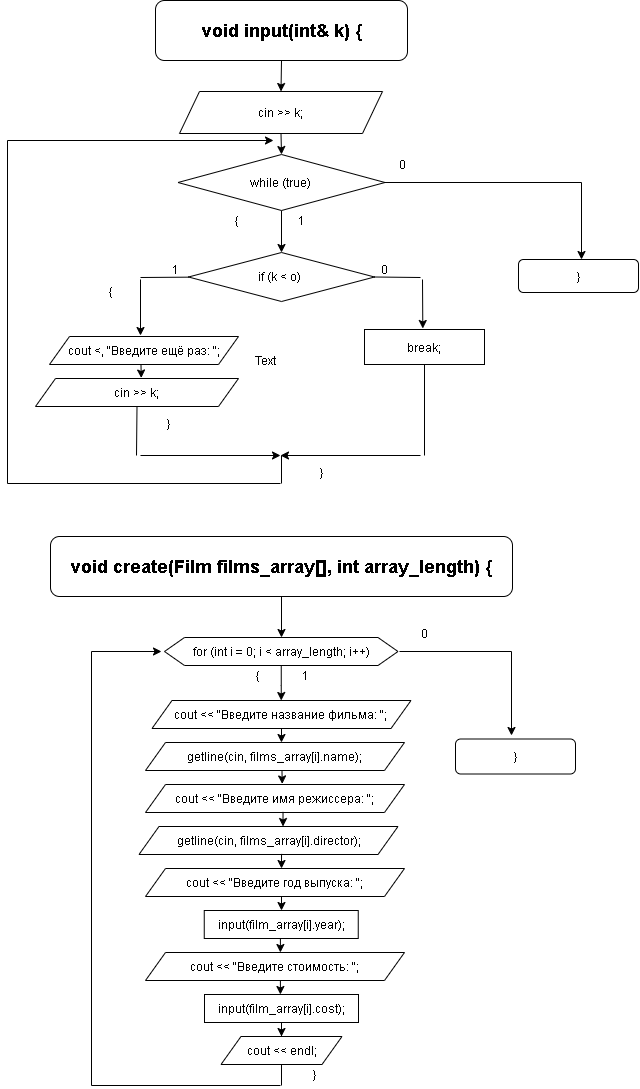
Удалить все элементы, у которых стоимость превышает заданную, добавить элемент в начало файла.

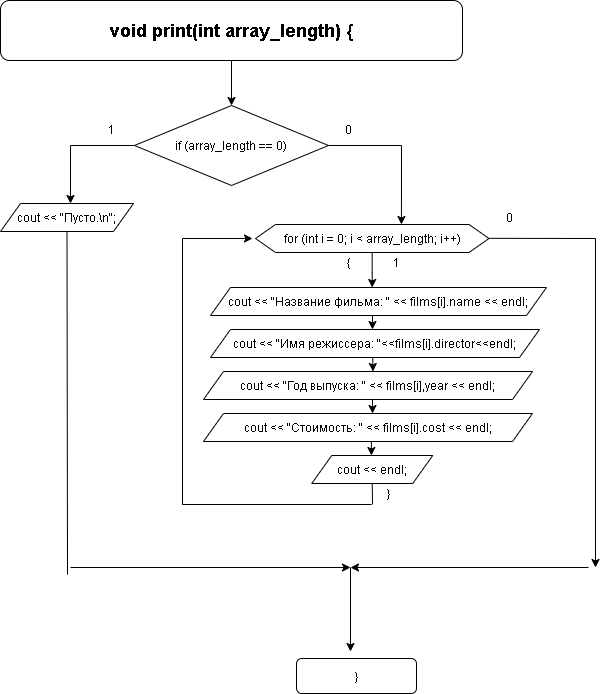
**Анализ задачи**

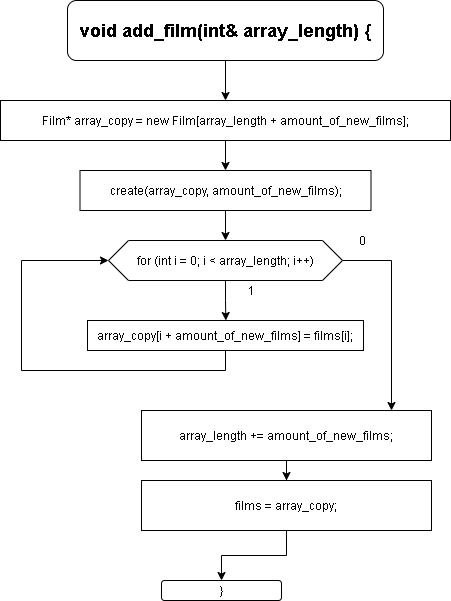
1. Вводим количество фильмов в переменную **int amount\_of\_films**.
2. Любой ввод целочисленных значений в программе осуществляется через специальную функцию **void input (int& k)**, которая проверяет введённые данные на корректность (если число меньше 0, выводится сообщение: «Ошибка, введите ещё раз»). В качестве аргумента функции используется ссылка.
3. Создаем структуру данных **struct Film**, полями которой являются **string name** (название фильма), **string director** (имя режиссера), **int year** (год выпуска), **int cost** (стоимость фильма).
4. Прописываем константное значение **const int amount\_of\_new\_films = 1** (столько фильмов необходимо добавить в начало).
5. Создаем динамический массив структур. В качестве длины массива указываем количество фильмов **amount\_of\_films**.
6. Заполняем массив структур данными с помощью функции **void create (Film films\_array[], int array\_length).**
7. Печатаем созданный массив структур в консоли.
8. Вводим установленную стоимость **int set\_cost**, дороже которой фильм не может стоить.Для этого используем функцию **void delete\_film (amount\_of\_films, set\_cost).**
9. Проходим по всем элементам массива с помощью вложенных циклов. Если находим фильм, который необходимо удалить, сдвигаем все элементы влево и уменьшаем длину массива на 1 (**array\_length --**).
10. Создаем новый массив структур с новой длиной массива. Проходим в цикле по новому массиву от начала до конца и присваиваем текущему элементу созданного массива соответствующий элемент старого массива.
11. Повторяем шаги 9-10 до тех пор, пока не удалим все фильмы, стоимость которых больше заданной.
12. Печатаем массив в консоли.
13. Обращаемся к функции **void add\_film (amount\_of\_films)** для добавления данных нового фильма.
14. Создаем новый динамический массив структур для хранения старых фильмов и добавления одного нового.
15. Вводим данные о новом фильме в новый массив в самое начало.
16. Присваиваем элементам нового массива элементы старого массива (начиная с индекса 1, так как нулевой элемент занят новым фильмом).
17. Увеличиваем длину массива на 1. И заменяем старый массив новым.
18. Печатаем обновленный массив в консоли.
19. Ввод строковой информации в программе осуществляется с помощью **getline ()**. Вывод данных в консоли – **cout**, ввод числовой информации в консоли – **cin.**
20. В программе используются циклы со счетчиком for, циклы с предусловием while, операторы условно ветвления if-else, оператор разрыва break.

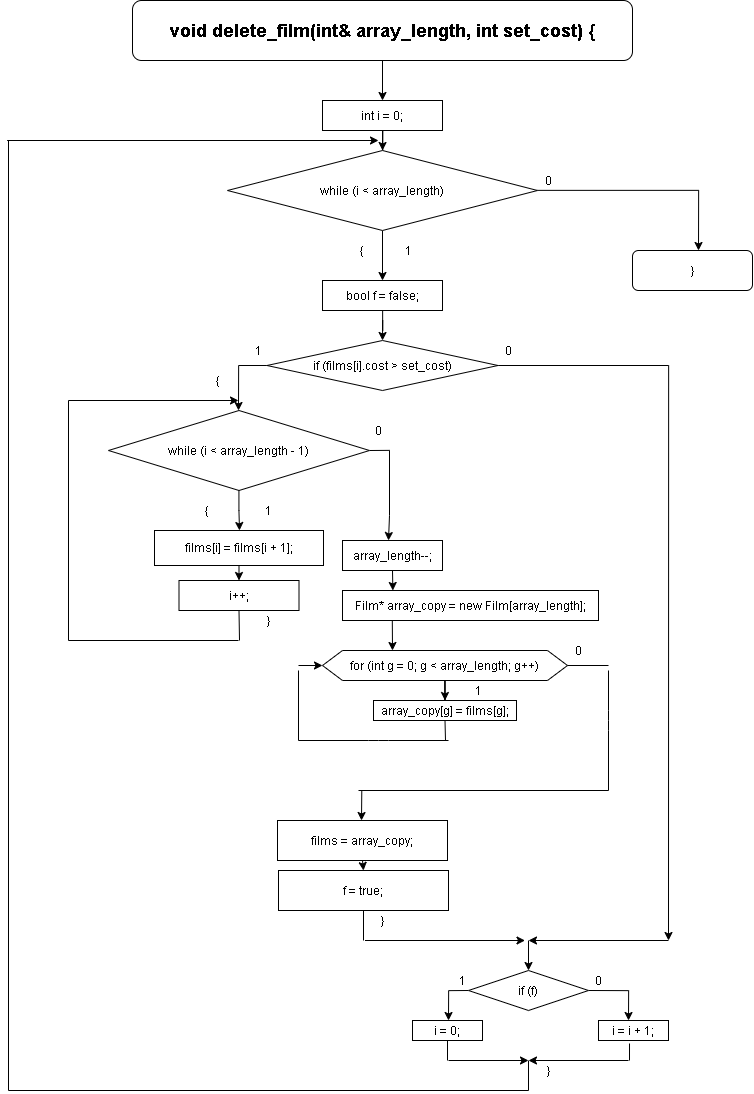
**Блок-схема**











**Код C++**

#include <iostream>

#include <string>

using namespace std;

struct Film {

string name;

string director;

int year;

int cost;

};

Film\* films;

const int amount\_of\_new\_films = 1;

void input(int& k) {

cin >> k;

while (true) {

if (k < 0) {

cout << "Ошибка, введите ещё раз: ";

cin >> k;

}

else

break;

}

}

void create(Film films\_array[], int array\_length) {

cout << "Заполните данные о фильмах:\n";

for (int i = 0; i < array\_length; i++) {

cout << "Введите название фильма: ";

cin.ignore();

getline(cin, films\_array[i].name);

cout << "Введите имя режиссера: ";

getline(cin, films\_array[i].director);

cout << "Введите год выпуска: ";

input(films\_array[i].year);

cout << "Введите стоимость фильма: ";

input(films\_array[i].cost);

cout << endl;

}

}

void print(int array\_length) {

if (array\_length == 0)

cout << "Пусто.\n";

else {

cout << endl;

cout << "Информация о фильмах:\n";

for (int i = 0; i < array\_length; i++) {

cout << "Название фильма: " << films[i].name << endl;

cout << "Имя режиссера: " << films[i].director << endl;

cout << "Год выпуска: " << films[i].year << endl;

cout << "Стоимость: " << films[i].cost << endl;

cout << endl;

}

}

}

void add\_film(int& array\_length) {

Film\* array\_copy = new Film[array\_length + amount\_of\_new\_films];

create(array\_copy, amount\_of\_new\_films);

for (int i = 0; i < array\_length; i++)

array\_copy[i + amount\_of\_new\_films] = films[i];

array\_length += amount\_of\_new\_films;

films = array\_copy;

}

void delete\_film(int& array\_length, int set\_cost) {

int i = 0;

while (i < array\_length) {

bool f = false;

if (films[i].cost > set\_cost) {

while (i < array\_length - 1) {

films[i] = films[i + 1];

i++;

}

array\_length--;

Film\* array\_copy = new Film[array\_length];

for (int g = 0; g < array\_length; g++)

array\_copy[g] = films[g];

films = array\_copy;

f = true;

}

if (f)

i = 0;

else

i = i + 1;

}

}

int main()

{

setlocale(LC\_ALL, "rus");

system("chcp 1251>nul");

cout << "Введите количество фильмов: ";

int amount\_of\_films = 0;

input(amount\_of\_films);

films = new Film[amount\_of\_films];

create(films, amount\_of\_films);

print(amount\_of\_films);

cout << "Удаление фильма:\n";

cout << "Введите заданную стоимость, не выше которой должен быть фильм: ";

int set\_cost;

input(set\_cost);

delete\_film(amount\_of\_films, set\_cost);

print(amount\_of\_films);

cout << "Данные о следующих фильмах будут добавлены в базу данных:\n";

add\_film(amount\_of\_films);

cout << "Содержимое базы данных после внесенных изменений:\n";

print(amount\_of\_films);

return 0;

}

**Результаты выполнения**

