Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ**

**по лабораторной работе №8**

Дисциплина: основы алгоритмизации и программирования

Тема: «Блоковый ввод-вывод» в С++

Выполнил работу:

студент группы РИС-20-2Б

Морозова Екатерина Максимовна

Проверила

Доцент кафедры ИТАС

Полякова О.А.

Пермь

2021

**Цель работы**

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

**Постановка задачи**

Формирование, печать, добавление и удаление элементов оформить в виде функций. Предусмотреть сообщения об ошибках при открытии файла и выполнении операций ввода/вывода.

Структура "Фильм":

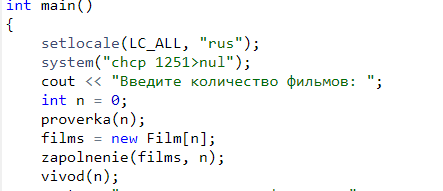
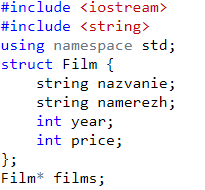
* Название фильма;
* Имя режиссера;
* Год выпуска;
* Стоимость

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

**Анализ задачи**

1.Необходимые действия

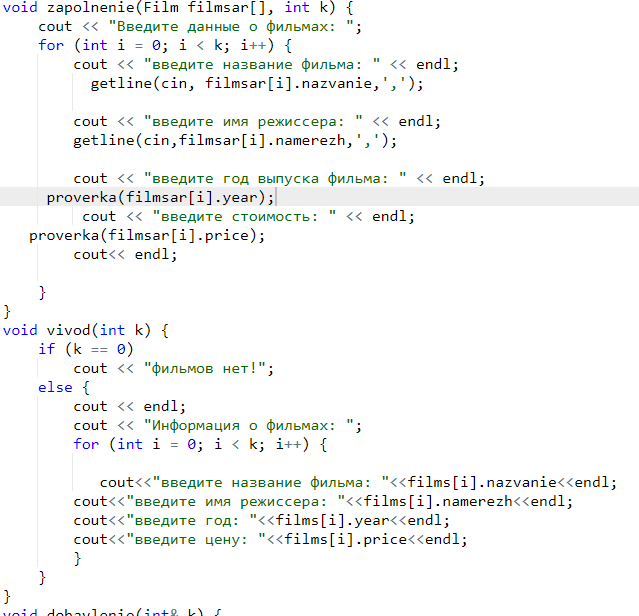
1) Ввести значение переменной n, которая будет равна количеству фильмов и в дальнейшем использовать эту переменную в качестве длины динамического массива.  
2) Создать динамический массив структур с полями для названия фильма, имени режиссёра, года выпуска и стоимости, потраченной на фильм.



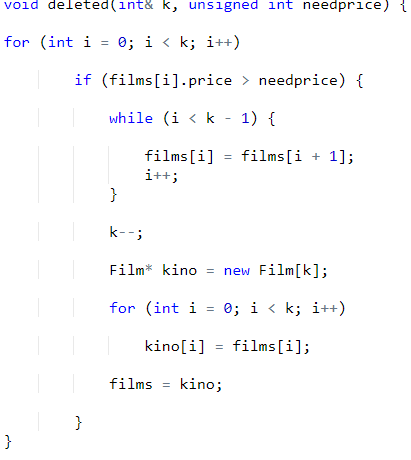
3) Заполнить массив структур данными о фильмах и вывести его на консоль.



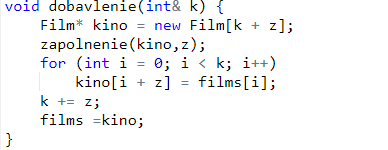
Функции в подробном виде:

  
5) Ввести определенную стоимость для отбора подходящих фильмов.

  
6) Далее проходимся по массиву структур, используя в теле цикла условный оператор, который сравнивает заданную нами стоимость и стоимость фильма. Если стоимость фильма больше заданной, то данный элемент удаляется и элементы массива сдвигаются влево.  
Затем длина массива уменьшается на единицу.  
Создаем снова динамический массив структур с новой длиной массива. Проходимся в цикле по новому массиву от начала до конца, присваивая текущему элементу нового массива текущий элемент старого массива.Затем присваиваем старому массиву новый массив с помощью оператора присваивания. Старый массив удалится, его длина будет меньше на единицу. Вместо него будет создан другой массив с копиями элементов, за исключением того, который программа стерла.

  
7) Выводим для наглядности массив структур на консоль  
8) Создаем новый динамический массив структур для хранения старой информации о фильмах и добавления информации о новом фильме в начало  
9) Вводим Новый элемент  
10) Проходимся в цикле от 0 до конца (длина старого массива сохранилась), присваивая элементам нового массива (начиная с индекса 1) элементы старого массива

11) Увеличить длину старого массива на 1 и присваиваем старому массиву новый массив

  
12) Выводим обновлённый массив структур на консоль



2. Программныйкод

#include <iostream>

#include <string>

using namespace std;

struct Film {

string nazvanie;

string namerezh;

int year;

int price;

};

Film\* films;

constint z = 1;

voidproverka(int&num) {

cin>>num;

while(true)

{

if (num<=0) {

cout<< "некорректный ввод, введите заново: ";

cin>>num;

} else break;}}

voidzapolnenie(Film filmsar[], int k) {

cout<< "Введите данные о фильмах: ";

for (inti = 0; i< k; i++) {

cout<< "введите название фильма: " <<endl;

getline(cin, filmsar[i].nazvanie,',');

cout<< "введите имя режиссера: " <<endl;

getline(cin,filmsar[i].namerezh,',');

cout<< "введите год выпуска фильма: " <<endl;

proverka(filmsar[i].year);

cout<< "введитестоимость: " <<endl;

proverka(filmsar[i].price);

cout<<endl;

}

}

voidvivod(int k) {

if (k == 0)

cout<< "фильмовнет!";

else {

cout<<endl;

cout<< "Информация о фильмах: ";

for (inti = 0; i< k; i++) {

cout<<"введитеназваниефильма: "<<films[i].nazvanie<<endl;

cout<<"введитеимярежиссера: "<<films[i].namerezh<<endl;

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

cout<<"введитецену: "<<films[i].price<<endl;

}

}

}

voiddobavlenie(int& k) {

Film\* kino = new Film[k + z];

zapolnenie(kino,z);

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

kino[i + z] = films[i];

k += z;

films =kino;

}

void deleted(int& k, unsigned intneedprice) {

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

if (films[i].price >needprice) {

while (i< k - 1) {

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

i++;

}

k--;

Film\* kino = new Film[k];

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

kino[i] = films[i];

films = kino;

}

}

int main()

{

setlocale(LC\_ALL, "rus");

system("chcp 1251>nul");

cout<< "Введите количество фильмов: ";

int n = 0;

proverka(n);

films = new Film[n];

zapolnenie(films, n);

vivod(n);

cout<< "сокращние списка фильмов: ";

cout<< "Введите определенную стоимость: ";

intneedprice;

proverka(needprice);

deleted(n,needprice);

vivod(n);

cout<< "добавление нового элемента";

dobavlenie(n);

cout<< "обновленный список фильмов: ";

vivod(n);

system("pause");

return 0;

}

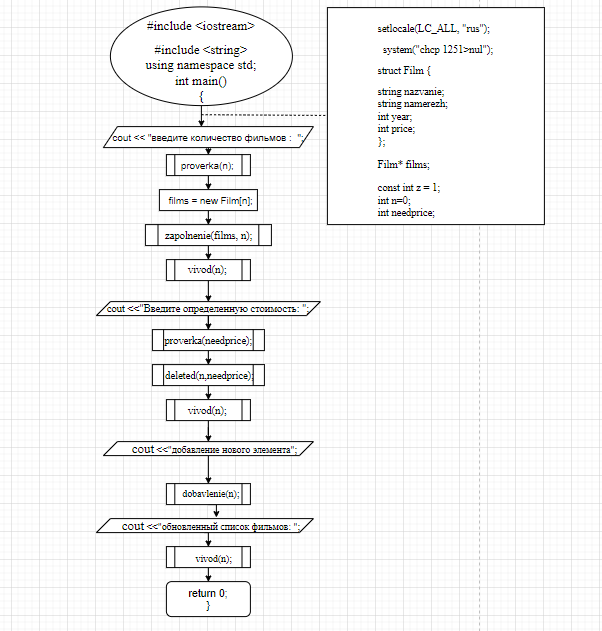
3.Типы переменных  
Struct, string, int, динамический массив структур  
4. Представление данных  
Struct: представление данных о фильмах  
String: Название фильма и имя режиссёра  
Int: год выпуска фильма, стоимость фильма,длина массива  
Динамический массив структур: хранение данных о фильмах  
5. Поля структуры  
nazvanie – хранение названия фильма;

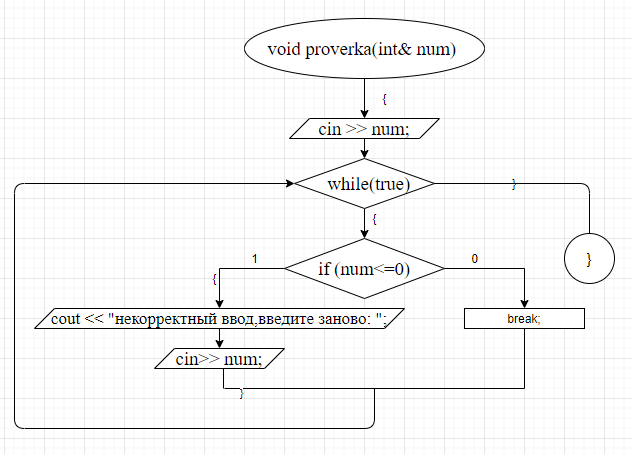
namerezch– хранение имени режиссёра;

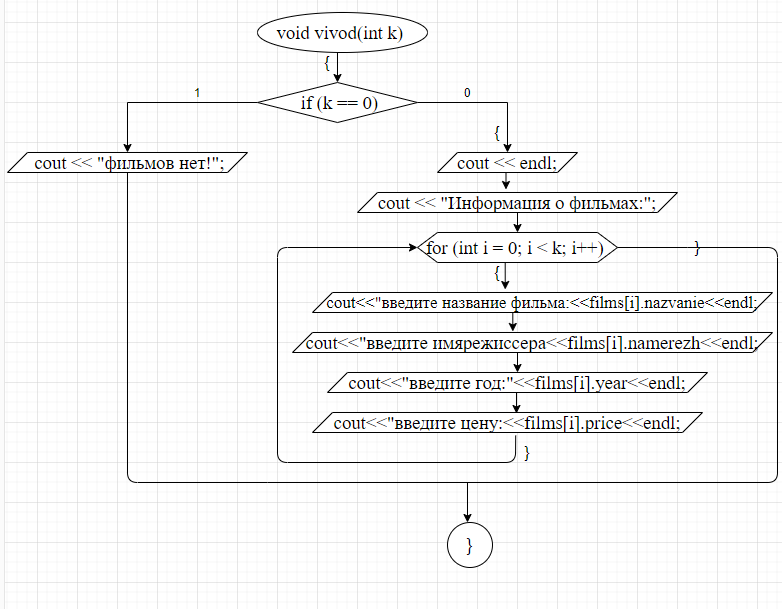
year– хранение числа, обозначающего год выпуска фильма;

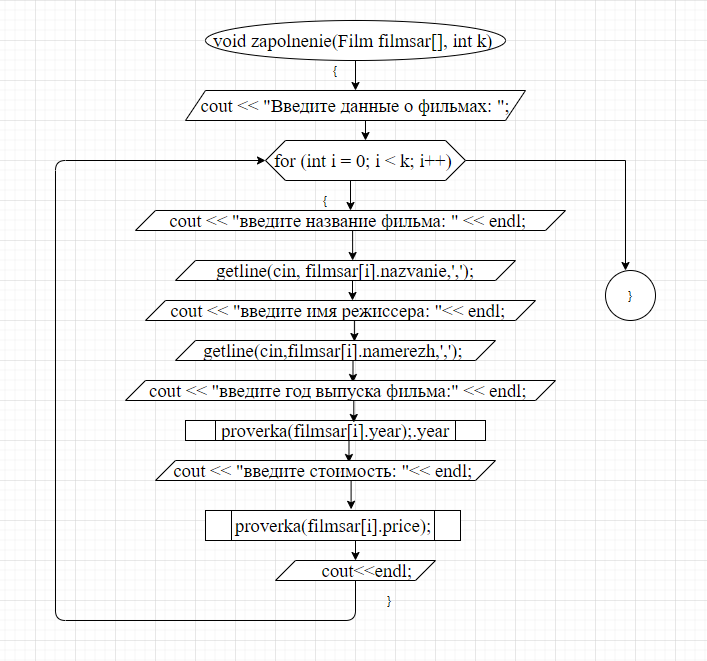
price– хранение числа, обозначающего стоимость фильма.  
6. Операторы ввода-вывода  
Cout – вывод данных на консоль  
Cin – ввод числовой информации в консоли  
Getline – ввод строковой информации для полей nazvanie и Namerezh в структуре  
7. Используемые операторные структуры  
В программе использовались следующие операторные структуры: цикл с предусловием, цикл со счетчиком for, оператор условного ветвления if-else и оператор возврата return.  
8. Проблемы и их решения  
Для проверки корректности ввода данных была создана отдельная функция. Она принимала переменную целого типа. Изначально она запрашивает ввести число, затем в цикле с предусловием происходит проверка числа на корректность. В теле цикла был использован условный оператор, в котором происходила проверка на корректность ввода числа. Если ввод некорректный, то выводится сообщение о том, что нужно ввести число заново. Если же пользователь ввел отрицательное число или 0, то также выводится сообщение о том, что нужно ввести положительное число. Выход из цикла происходит, когда пользователь вводит положительное число.

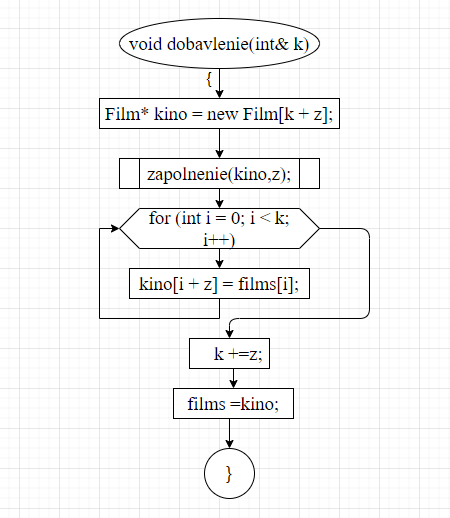
**Блок-схема**

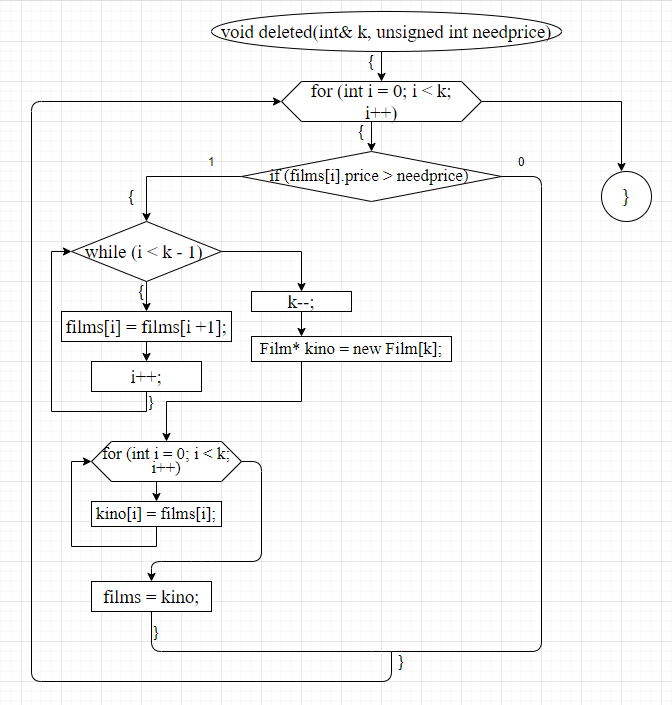
****



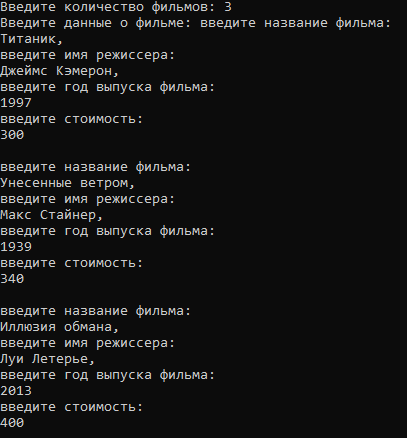


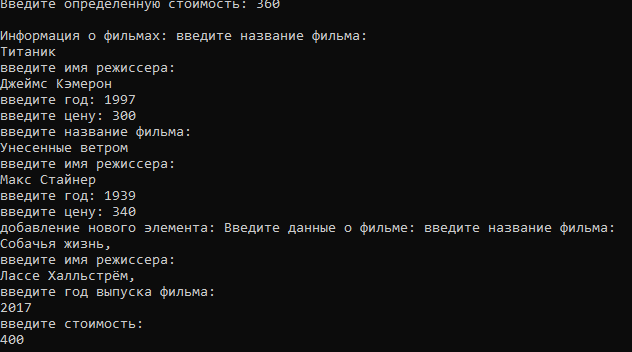


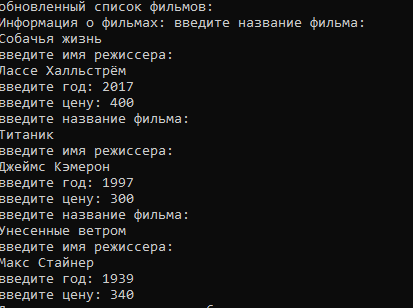




**Результат**

****

****

****