Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №9**

Дисциплина: «Информатика»

Тема: Структурное программирование. Файловые потоки

Вариант 17

Выполнил:

Студент группы Рис-20-1б

Томилов Владислав Алексеевич

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2021**

**Цель задачи**

Цель – написать программу с использованием структур данных и выполнение операций с ними.

Для достижения поставленной цели необходимо решить следующие задачи:

* Найти как создаются и объявляются структуры данных.
* Найти как работают структуры данных.
* Найти оптимальные методы ввода информации в структуру данных.
* Разработать программу по найденному материалу.

Постановка задачи

Вар.17. Задача состоит в написании программы, создать структура с названием “Фильм” с полями название, режиссёр, год выпуска, стоимость. Заполнить структуру данными. Удалить все элементы, у которых стоимость превышает заданную. Добавить элемент в начало файла. Для выполнения данной работы использован язык программирования C++ в программной среде Visual Studio 2019.

Анализ задачи

1. Определить какие операции должны быть выполнены по заданию:

* Поиск по стоимости фильма.
* Добавление элемента в начало файла.
* Для поля “Название” создано поле типа string.
* Для поля “Режиссёр” создано поле типа string.
* Для поля “Год выпуска” создано поле типа int.
* Для поля “Стоимость” создано поле типа int.
* Важный момент:

Для заполнения полей “Название” и “Режиссер” разработана функция getline().

1. Разработать структуру “Фильм” и объявить её:

Разработка структуры проведена для 100 элементов:

Film a[100];

1. Также создаем дополнительную структуру, которая будет являться отредактированным (конечным) списком фильмов.

Film b[100];

1. Инициализация переменной N, отвечающей за количество фильмов в списке. Значение N вводится пользователем с клавиатуры через <cin>.
2. Данные в структуру введены через цикл for, в этом же цикле введенные данные проверены на корректность ввода.

Название и ФИО режиссера каждого фильма вводится через функцию getline():

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

getline(cin, a[i].director);

1. Инициализация переменной p, отвечающей за заданную стоимость для сортировки списка фильмов. Значение p вводится пользователем с клавиатуры через <cin>.

Поскольку элементы, в которых содержится информация о фильмах, стоимость которых превышает заданную, нужно удалить, используем условный оператор <if (a[i].price <= p)> для проверки этого условия.

Если же элемент удовлетворяет условию, то производится его запись в дополнительную структуру сразу на 2 место, так как в дальнейшем потребуется записать элемент в начало файла (то есть на первое место дополнительной структуры).

1. Производится запись данных еще одного фильма в начало дополнительной структуры, то есть b[0].name, b[0].director, b[0].date, b[0].price, используя оператор <cin> и функцию getline().
2. Структура выводится в консоль через цикл <for>.

Код

#include <iostream>

#include <string>

using namespace std;

struct Film {

string name;

string director;

int date;

int price;

};

int main()

{

system("chcp 1251>nul");

Film a[100];

Film b[100];

int N;

cout << "Введите количество фильмов в списке - ";

while (!(cin >> N) || (cin.peek() != '\n') || N <= 0)

{

cin.clear();

system("cls");

while (cin.get() != '\n');

{

cout << "Введите количество фильмов в списке - ";

}

}

system("cls");

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

{

cout << "Название: " << endl; cin.ignore(); getline(cin, a[i].name);

cout << "Режиссёр: " << endl; getline(cin, a[i].director);

cout << "Год выпуска: " << endl;

while (!(cin >> a[i].date) || (cin.peek() != '\n') || a[i].date <= 0)

{

cin.clear();

system("cls");

while (cin.get() != '\n');

{

cout << "Название: " << endl << a[i].name << endl << "Режиссёр: " << endl << a[i].director << endl << "Год выпуска: " << endl;

}

}

cout << "Стоимость: " << endl;

while (!(cin >> a[i].price) || (cin.peek() != '\n') || a[i].price <= 0)

{

cin.clear();

system("cls");

while (cin.get() != '\n');

{

cout << endl << "Название: " << endl << a[i].name << endl << "Режиссёр: " << endl << a[i].director << endl << "Год выпуска: " << endl << a[i].date << endl << "Стоимость: " << endl;

}

}

system("cls");

}

int p;

cout << "Введите искомую стоимость фильма - ";

while (!(cin >> p) || (cin.peek() != '\n') || p <= 0)

{

cin.clear();

system("cls");

while (cin.get() != '\n');

{

cout << "Введите искомую стоимость фильма - ";

}

}

int j = 1;

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

{

if (a[i].price <= p)

{

b[j].name = a[i].name;

b[j].director = a[i].director;

b[j].date = a[i].date;

b[j].price = a[i].price;

j++;

}

}

system("cls");

N = j;

cout << "Введите данные фильма, который хотите добавить:";

cout << endl << "Название: " << endl; cin.ignore(); getline(cin, b[0].name);

cout << "Режиссёр: " << endl; getline(cin, b[0].director);

cout << "Год выпуска: " << endl; cin >> b[0].date;

cout << "Стоимость: " << endl; cin >> b[0].price;

system("cls");

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

{

cout << endl << i + 1 << ". " << '\t' << b[i].name << endl;

cout << '\t' << b[i].director << endl;

cout << '\t' << b[i].date << endl;

cout << '\t' << b[i].price << endl;

}

}

Работа кода













