МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»

РУТ(МИИТ)

Кафедра “УТБиИС”

**Отчёт №8**

**по дисциплине «Программирование»**

Выполнил: Гнидин Станислав Евгеньевич

Группа: УИБ-111

Проверил: Никольская М.Н.

МОСКВА 2020

Задача 7.

Ввести и записать в файл информацию о репертуаре кинотеатров.

Таблица имён

Main – основная программа

|  |  |
| --- | --- |
| Исходные данные | |
| notes | Массив известных сеансов |
| Результат | |
| Рабочие переменные | |
| count | Количество сеансов |
| Name | Название фильма нового сеанса |
| Time | Время нового сеанса |
| t | Временное хранилище времени |

read – функция для чтения репертуара

|  |  |
| --- | --- |
| Исходные данные | |
| file | Название файла для чтения |
| Результат | |
| n | Количество считанных сеансов |
| notes | Массив считанных сеансов |
| Рабочие переменные | |
| index | Индекс считываемого сеанса |

Write – функция для для записи массива сеансов в файл

|  |  |
| --- | --- |
| Исходные данные | |
| notes | Массив сеансов |
| file | Название файла для записи |
| n | Количество сеансов в массиве |
| Результат | |
| Рабочие переменные | |
| i | Счётчик цикла |

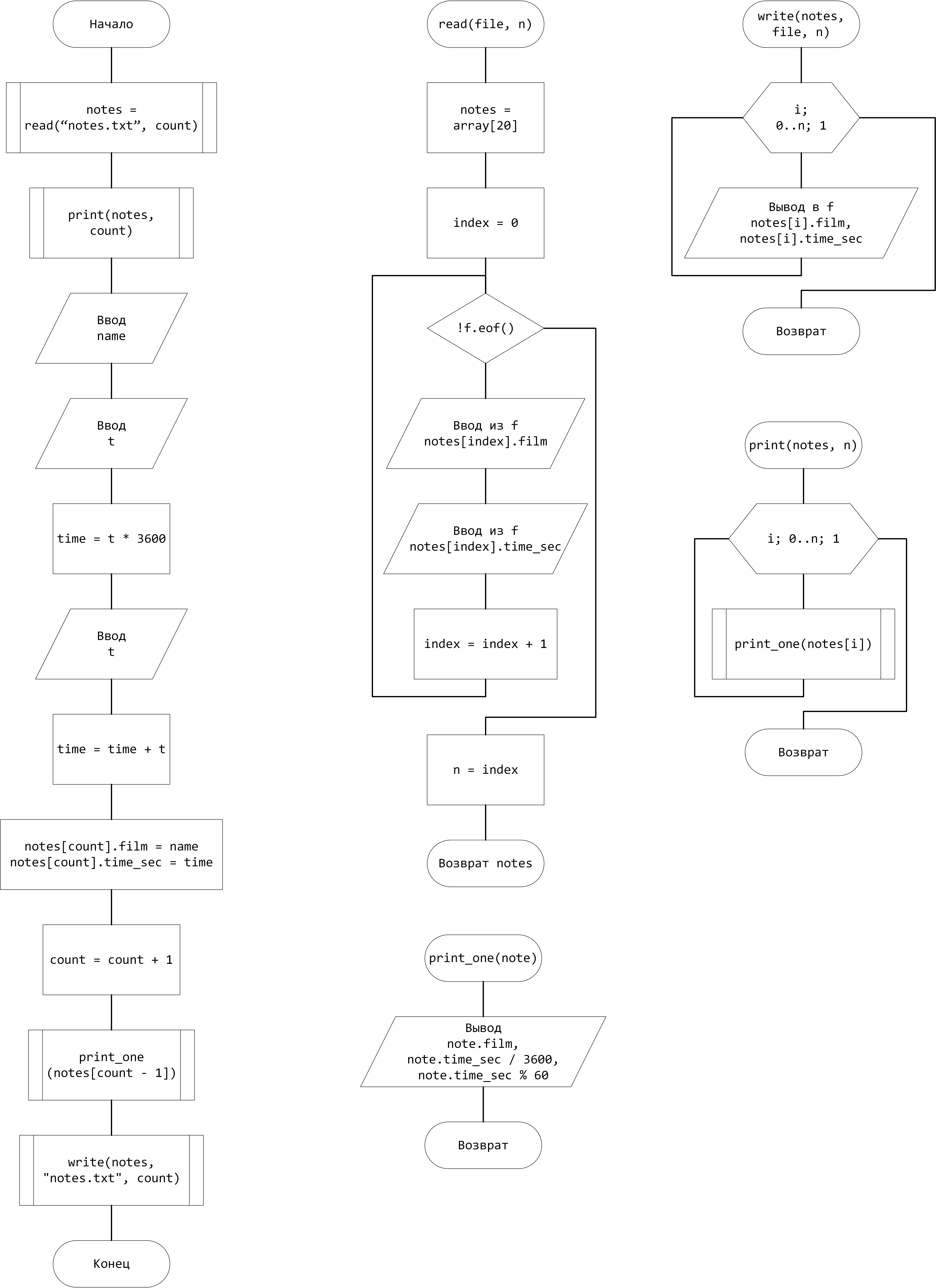
Print\_one – функция для записи одного сеанса в консоль

|  |  |
| --- | --- |
| Исходные данные | |
| note | Сеанс |
| Результат | |
| Рабочие переменные | |

print – функция для вывода всех сеансов в консоль

|  |  |
| --- | --- |
| Исходные данные | |
| notes | Массив сеансов |
| n | Количество сеансов |
| Результат | |
| Рабочие переменные | |
| i | Счётчик цикла |

Блок схема



Проверочные данные

|  |  |
| --- | --- |
| Ввод | Вывод |
| Avengers 50435  Spider-man 64820  Smeshariki 82850  Alladin 19 40 | Avengers 50435  Spider-man 64820  Smeshariki 82850  Введённый сеанс:  Alladin в 19:40 |

Код программы на языке C++.

#include <iostream>

#include <fstream>

using namespace std;

struct Note

{

string film;

int time\_sec;

};

Note\* read(const string& file, int& n)

{

Note\* notes = new Note[20];

int index = 0;

ifstream f(file);

while (!f.eof())

{

f >> notes[index].film;

f >> notes[index].time\_sec;

index++;

}

f.close();

n = index;

return notes;

}

void write(Note\* notes, const string& file, int n)

{

ofstream f(file);

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

{

f << notes[i].film << " " << notes[i].time\_sec << "\n";

}

f.close();

}

void print\_one(Note\* note)

{

cout << note->film << " в " << note->time\_sec / 3600 << ":" << note->time\_sec % 60 << "\n";

}

void print(Note\* notes, int n)

{

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

{

print\_one(&notes[i]);

}

}

int main()

{

setlocale(LC\_ALL, "russian");

int count;

Note\* notes = read("notes.txt", count);

print(notes, count);

string name;

int time;

cout << "Введите название нового фильма: ";

cin >> name;

int t;

cout << "Введите время сеанса (ч м): ";

cin >> t;

time = t \* 3600;

cin >> t;

time += t;

notes[count].film = name;

notes[count].time\_sec = time;

count++;

cout << "Введённый сеанс: \n";

print\_one(&notes[count - 1]);

write(notes, "notes.txt", count);

system("pause");

}

Листинг  
