**ГЛАВА 17**

**1.**

Для определения типа данных нужно решить, как данные будут хранится и какие операции можно совершать над ними.

**2.**

Обход связного списка в листинге 17.2 может осуществляться только в одном направлении потому, что это односвязный список и каждый узел списка хранит только один указатель на следующий узел в списке.

Для создания двусвязного списка, который будет хранить два указателя, на предыдущий узел и на следующий нужно сделать так:

struct film

{

char title[TSIZE];

int rating;

struct film \* prev;

struct film \* next;

};

**3.**

ADT – это абстрактный тип данных. Он создается для того, чтобы упростить проектирование и изменение программы программистом. Абстрактный тип данных определяет свойства типа и операции, которые можно выполнять над этим типом.

**4.**

Если функцию QueueIsEmpty() переписать так, чтобы она в качестве аргумента принимала саму структуру, а не указатель, нужно было бы делать чтобы функция и возвращала эту же структуру. Ведь при передаче функции самой структуры, создается ее копия, проводятся операции над копией и после функция возвращает копию структуры, которую нужно присвоить той структуре, которую мы передавали, чтобы ее изменить. Плюсом при передаче самой структуры является то, что нельзя случайным образом изменить оригинал структуры, ведь операции проводятся над ее копией.

Минус заключается в том, что нужно выделять лишнюю память для копии структуры и это займет больше времени.

Плюсом при передаче адреса структуры является то, что не нужно создавать копию и можно работать напрямую с оригиналом.

Минус в передаче указателя на структуру это то, что при не осторожной работе можно изменить данные, которые нельзя было трогать.

**5.**

a.

Определение ADT для стека:

Имя типа: Стек

Свойства типа: Может содержать упорядоченную последовательность элементов

Операции типа: Инициализация стека пустым содержимым

Определение, является ли стек пустым

Определение, является ли стек полным

Определение количества элементов в стеке

Добавление элемента в конец стека

Удаление элемента из конца стека и инициализация предыдущего элемента пустым содержимым

Опустошение стека

**6.**

При последовательном поиске: 3 = 3, 1023 = 1023, 65535 = 65535.

При двоичном поиске: 3 = 2, 1023 = 10, 65535 = 16.