

Федеральное государственное автономное образовательное учреждение
высшего образования «Санкт-Петербургский государственный
электротехнический университет «ЛЭТИ»
им. В.И. Ульянова (Ленина)»

кафедра физики

ОТЧЕТ
по лабораторной работе № 2
«Структуры. Линейные списки»

Выполнил: _____ Прокофьев М.Д.

Группа № 9304

Преподаватель: _____ Чайка К.В.

Санкт-Петербург, 2020

ЦЕЛЬ РАБОТЫ:

Обучение работы с линейными списками, а конкретно с двунаправленными, с использованием структур данных.

ЗАДАНИЕ:

Создайте двунаправленный список музыкальных композиций MusicalComposition и api (application programming interface - в данном случае набор функций) для работы со списком.

Структура элемента списка (тип - MusicalComposition)

name - строка неизвестной длины (гарантируется, что длина не может быть больше 80 символов), название композиции.

author - строка неизвестной длины (гарантируется, что длина не может быть больше 80 символов), автор композиции/музыкальная группа.

year - целое число, год создания.

Функция для создания элемента списка (тип элемента MusicalComposition)

```
MusicalComposition* createMusicalComposition(char* name, char* author, int year)
```

Функции для работы со списком:

```
MusicalComposition* createMusicalCompositionList(char** array_names, char** array_authors, int* array_years, int n); // создает список музыкальных композиций MusicalCompositionList, в котором:
```

n - длина массивов array_names, array_authors, array_years.

поле name первого элемента списка соответствует первому элементу списка array_names (array_names[0]).

поле author первого элемента списка соответствует первому элементу списка array_authors (array_authors[0]).

поле year первого элемента списка соответствует первому элементу списка array_years (array_years[0]).

Аналогично для второго, третьего, ... n-1-го элемента массива.

! длина массивов array_names, array_authors, array_years одинаковая и равна n, это проверять не требуется.

Функция возвращает указатель на первый элемент списка.

```
void push(MusicalComposition* head, MusicalComposition* element); //
добавляет element в конец списка musical_composition_list
void removeEl (MusicalComposition* head, char* name_for_remove); // удаляет
элемент element списка, у которого значение name равно значению
name_for_remove
int count(MusicalComposition* head); //возвращает количество элементов
списка
void print_names(MusicalComposition* head); //Выводит названия композиций
```

В функции main написана некоторая последовательность вызова команд для проверки работы вашего списка.

Функцию main менять не нужно

ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

- Структура данных - это совокупность переменных, объединенных одним именем, предоставляющая общепринятый способ совместного хранения информации. Объявление структуры приводит к образованию шаблона, используемого для создания объектов структуры. Переменные, образующие структуру, называются членами структуры
- Линейный список – базовая динамическая структура данных в программировании, состоящая из узлов, каждый из которых содержит как собственно данные, так и одну или две ссылки («связки») на следующий и/или предыдущий узел списка. Принципиальным преимуществом перед массивом является структурная гибкость: порядок элементов связного списка может не совпадать с порядком расположения элементов данных в памяти компьютера, а порядок обхода списка всегда явно задаётся его внутренними связями.
- Однонаправленный список – это структура данных, представляющая собой последовательность элементов, в каждом из которых хранится значение и указатель на следующий элемент списка

- Двунаправленный список - это структура данных, состоящая из последовательности элементов, каждый из которых содержит информационную часть и два указателя на соседние элементы

ВЫВОДЫ

Получены навыки работы с структурами, а также их использование в линейных списках