

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра Вычислительной техники

Курсовая работа
по дисциплине «Программирование»
Тема: Разработка электронной картотеки

Студент гр. 3312

Шарапов И. Д.

Преподаватель

Аббас С. А.

Санкт-Петербург
2024

Содержание

Цель работы	3
Задание	3
Постановка задачи и описание решения.....	3
Описание структур.....	3
Описание функций	4
Структура вызова функций	5
Описание переменных	5
Схема алгоритма.....	8
Текст программы	8
Контрольные примеры.....	8
Содержимое файлов.....	8
Примеры выполнения программы.....	8
Выводы	8

Цель работы

Введение: цель + краткая формулировка задачи (с учетом предметной области). Пример краткой формулировки: «Создание электронной картотеки суперкаров».

Задание

Задание (общая формулировка) с конкретизацией пунктов меню, без раздела "Дополнительно".

Постановка задачи и описание решения

Ну, аббас просил ещё в разделе «Постановка задачи и описание решения» сразу прописывать свою предметную область.

Описание структур

1. Структура Athlete

№	Имя переменной	Тип	Назначение
1	name	char*	Имя спортсмена
2	university	char*	Название университета
3	age	int	Возраст
4	weight	float	Вес в килограммах
5	height	int	Рост в сантиметрах
6	result	int[3]	Результаты выступления
7	index	float	Отношение результата к весу

2. Структура NodeOfList

№	Имя переменной	Тип	Назначение
1	id	int	Уникальный ID
2	data	Athlete*	Указатель на данные о спортсмене
3	next	struct NodeOfList*	Указатель на следующую вершину списка
4	prev	struct NodeOfList*	Указатель на предыдущую вершину списка

3. Структура ListOfAthlete

№	Имя переменной	Тип	Назначение
1	length	int	Уникальный ID

2	first	struct NodeOfList*	Указатель на первую вершину списка
3	last	struct NodeOfList*	Указатель на последнюю вершину списка

Описание функций

№	Название	Назначение
1	main	Основная функция программы. Открывает файл, инициализирует список, отвечает за взаимодействие с пользователем через меню. В конце очищает память.
2	from_str_to_int	Конвертирует строку в целочисленное значение.
3	from_str_to_float	Конвертирует строку в число с плавающей точкой.
4	from_str_to_int_mas	Конвертирует строку в массив целочисленных значений длины 3.
5	fill_struct	Извлекает данные из строки и заполняет поля структуры Athlete.
6	make_list	Создаёт список (выделяет память и возвращает указатель на него).
7	create_node	Выделяет память под новую вершину списка, заполняет его данными и возвращает указатель на него.
8	help	Отображает доступные команды пользователю в консоли.
9	wait	Приостанавливает выполнение программы до нажатия клавиши Enter.
10	print_line	Выводит линию для разделения таблицы данных в консоли.
11	print_head	Выводит поля заголовка таблицы в консоль.
12	print_node	Выводит данные одного элемента списка в консоль.
13	print_one	Выводит данные одного элемента списка с заголовком в консоль.
14	print	Выводит все элементы списка в консоль.
15	sorted	Сортировка выбранных элементов списка и вывод отсортированных данных в консоль. (не влияет на порядок основного списка)
16	find	Поиск элементов списка и вывод найденных данных в консоль с возможностью последующей сортировки этих данных.

17	get_mas	Функция проходит по всему списку и заполняет массив указателями на вершины списка. Возвращает массив указателей.
18	my_swap	Меняет два значения в списке местами.
19	sort	Сортирует элементы списка в соответствии с заданным параметром и выводит отсортированные данные в консоль.
20	add	Пользователь вводит данные нового элемента, функция создаёт новую вершину и добавляет её в конец списка.
21	edit	Пользователь вводит ID элемента, который хочет отредактировать, и новые данные. Функция находит элемент по ID и предлагает изменить его данные по отдельным полям.
22	delete	Пользователь выбирает поле для удаления и вводит критерии поиска. Функция удаляет элементы, соответствующие заданным критериям.
23	save	Пользователь выбирает имя файла, в который будут сохранены данные списка. Функция записывает все данные в файл в соответствии с форматом.

Структура вызова функций

Описание переменных

Функция <i>int main()</i>			
№	Имя переменной	Тип	Назначение
1	list	ListOfAthlete*	Список спортсменов
2	g_id	int	Глобальный ID
3	cl	int	Флаг для отображения help
4	filename	char[]	Буфер для хранения имени файла
5	str	char[]	Буфер для команд пользователя
6	text	char[]	Буфер для содержимого файла
7	cur_node	NodeOfList*	Текущий элемент в списке
8	f	FILE*	Указатель на файл
Функция <i>int from_str_to_int(char *str)</i>			
1	str	char*	Стока, которую нужно конвертировать в число
2	ans	int	Результирующее число
Функция <i>float from_str_to_float(char *str)</i>			
1	str	char*	Стока, которую нужно конвертировать в число с плавающей точкой

2	ans	float	Результирующее число с плавающей точкой
3	a	float	Целая часть числа
4	b	float	Дробная часть числа
Функция <i>void from_str_to_int_mas(char *str, int *mas)</i>			
1	str	char*	Строка, которую нужно конвертировать в массив чисел длины 3
2	mas	int*	Указатель на массив, в который будут записаны числа
3	ind	int	Текущий индекс в массиве
4	j	int	Текущий индекс в строке
Функция <i>Athlete *fill_struct(char *str)</i>			
1	str	char*	Строка, которую нужно конвертировать в сущность Athlete
2	user	Athlete*	Сущность Athlete
3	word	char*	Текущая подстрока в строке
4	pole	char*[]	Массив указателей на подстроки
5	ind	int	Текущий индекс в массиве подстрок
6	tt	int	Текущий индекс в строке
Функция <i>ListOfAthlete *make_list()</i>			
1	ph	ListOfAthlete*	Указатель на создаваемый список
Функция <i>NodeOfList *create_node(char *str, int g_id)</i>			
1	str	char*	Строка, которую нужно конвертировать в сущность Athlete
2	g_id	int	Глобальный ID
3	new_node	NodeOfList*	Указатель на создаваемую вершину
Функция <i>void print_node(NodeOfList *node)</i>			
1	node	NodeOfList*	Вершина списка, которую нужно вывести
Функция <i>void print_one(NodeOfList *node)</i>			
1	node	NodeOfList*	Вершина списка, которую нужно вывести
Функция <i>void print(ListOfAthlete *list)</i>			
1	list	ListOfAthlete*	Список, который нужно вывести
Функция <i>void sorted(int *mas, ListOfAthlete *list, int param)</i>			
1	mas	int*	Массив флагов для сортировки
2	list	ListOfAthlete*	Список спортсменов
3	param	int	Параметр, по которому сортируем
4	cur_node	NodeOfList*	Текущая вершина списка
5	min_node	NodeOfList*	Минимальная вершина списка
6	ind	int	Индекс минимальной вершины
Функция <i>void find(ListOfAthlete *list)</i>			
1	list	ListOfAthlete*	Список спортсменов
2	cur_node	NodeOfList*	Текущая вершина списка
3	str	char[]	Строка, которую ищет пользователь
4	mas	int[]	Массив флагов для поиска

5	fl	int	Флаг на то, что хоть один элемент найден
6	param	int	Параметр, по которому ищем
Функция <i>NodeOfList **get_mas(ListOfAthlete *list)</i>			
1	list	ListOfAthlete*	Список спортсменов
2	cur node	NodeOfList*	Текущая вершина списка
3	mas	NodeOfList**	Массив указателей на вершины списка
Функция <i>void my_swap(NodeOfList **mas, ListOfAthlete *list, int i, int j)</i>			
1	mas	NodeOfList**	Массив указателей на вершины списка
2	list	ListOfAthlete*	Список спортсменов
3	i	int	Индекс первого узла
4	j	int	Индекс второго узла
5	q	NodeOfList*	Вспомогательная вершина
Функция <i>void sort(ListOfAthlete *list)</i>			
1	list	ListOfAthlete*	Список спортсменов
2	mas	NodeOfList**	Массив указателей на вершины списка
3	str	char[]	Параметр в виде строки
4	n	int	Количество вершин в списке
5	param	int	Параметр, по которому сортируем
Функция <i>void add(ListOfAthlete *list, int g_id)</i>			
1	list	ListOfAthlete*	Список спортсменов
2	g_id	int	Глобальный ID
3	str	char[]	Буфер для данных о спортсмене
4	cur node	NodeOfList*	Текущая вершина списка
Функция <i>void edit(ListOfAthlete *list)</i>			
1	list	ListOfAthlete*	Список спортсменов
2	the_node	NodeOfList*	Вершина, которую редактируем
3	id	int	ID спортсмена для редактирования
4	str	char[]	Буфер для пользовательского ввода
5	s_id	char[]	Буфер ID спортсмена в строковом виде
Функция <i>void delete(ListOfAthlete *list)</i>			
1	list	ListOfAthlete*	Список спортсменов
2	cur_node	NodeOfList*	Текущая вершина списка
3	prev_node	NodeOfList*	Предыдущая вершина списка
4	str	char[]	Строка, которую ищет пользователь
5	ch	char	Символ для подтверждения
6	mas	int[]	Массив флагов для удаления
7	fl	int	Флаг на то, что хоть один элемент найден
8	param	int	Параметр, по которому ищем
9	cnt	int	Количество элементов для удаления
Функция <i>void save(ListOfAthlete *list)</i>			
1	list	ListOfAthlete*	Список спортсменов
2	f	FILE*	Указатель на файл
3	filename	char[]	Буфер для имени файла

Схема алгоритма

Текст программы

--

Контрольные примеры

№	Исходные данные	Результаты
1		
2		
3		

Содержимое файлов

Примеры выполнения программы

Выводы

В «Заключении» указывается, какие заголовочные файлы стандартной библиотеки использованы и для чего (какие функции из этих заголовочных файлов использованы), делаются выводы о достижении цели работы.