

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
Архитектура данных	Ошибка! Закладка не определена.
Структура вызова функций	5
Описание структур	3
Описание функций	4
Описание переменных	5
Схема алгоритма.....	7
Текст программы	7
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*	
2	g_id	int	
3	cl	int	
4	filename	char[]	
5	str	char[]	
6	text	char[]	
7	cur node	NodeOfList*	
8	f	FILE*	
Функция <i>int from_str_to_int(char *str)</i>			
1	str	char*	
2	ans	int	
Функция <i>float from_str_to_float(char *str)</i>			
1	str	char*	
2	ans	float	
3	a	float	

4	b	float	
Функция <i>void from str to int mas(char *str, int *mas)</i>			
1	str	char*	
2	mas	int*	
3	ind	int	
4	j	int	
Функция <i>Athlete *fill_struct(char *str)</i>			
1	str	char*	
2	user	Athlete*	
3	word	char*	
4	pole	char*[]	
5	ind	int	
6	tt	int	
Функция <i>ListOfAthlete *make_list()</i>			
1	ph	ListOfAthlete*	
Функция <i>NodeOfList *create_node(char *str, int g_id)</i>			
1	str	char*	
2	g_id	int	
3	new_node	NodeOfList*	
Функция <i>void print_node(NodeOfList *node)</i>			
1	node	NodeOfList*	
Функция <i>void print_one(NodeOfList *node)</i>			
1	node	NodeOfList*	
Функция <i>void print(ListOfAthlete *list)</i>			
1	list	ListOfAthlete*	
Функция <i>void sorted(int *mas, ListOfAthlete *list, int param)</i>			
1	mas	int*	
2	list	ListOfAthlete*	
3	param	int	
4	cur_node	NodeOfList*	
5	min_node	NodeOfList*	
6	ind	int	
Функция <i>void find(ListOfAthlete *list)</i>			
1	list	ListOfAthlete*	
2	cur_node	NodeOfList*	
3	str	char[]	
4	mas	int[]	
5	fl	int	
6	param	int	
Функция <i>NodeOfList **get_mas(ListOfAthlete *list)</i>			
1	list	ListOfAthlete*	
2	cur_node	NodeOfList*	
3	mas	NodeOfList**	

Функция <i>void my_swap(NodeOfList **mas, ListOfAthlete *list, int i, int j)</i>			
1	mas	NodeOfList**	
2	list	ListOfAthlete*	
3	i	int	
4	j	int	
5	q	NodeOfList*	
Функция <i>void sort(ListOfAthlete *list)</i>			
1	list	ListOfAthlete*	
2	mas	NodeOfList**	
3	str	char[]	
4	n	int	
5	param	int	
Функция <i>void add(ListOfAthlete *list, int g_id)</i>			
1	list	ListOfAthlete*	
2	g_id	int	
3	str	char[]	
4	cur_node	NodeOfList*	
Функция <i>void edit(ListOfAthlete *list)</i>			
1	list	ListOfAthlete*	
2	the_node	NodeOfList*	
3	id	int	
4	str	char[]	
5	s_id	char[]	
Функция <i>void delete(ListOfAthlete *list)</i>			
1	list	ListOfAthlete*	
2	cur_node	NodeOfList*	
3	prev_node	NodeOfList*	
4	str	char[]	
5	ch	char	
6	mas	int[]	
7	fl	int	
8	param	int	
9	cnt	int	
Функция <i>void save(ListOfAthlete *list)</i>			
1	list	ListOfAthlete*	
2	f	FILE*	
3	filename	char[]	

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.