САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра ВТ

КУРСОВАЯ РАБОТА

по дисциплине «Программирование»

Тема: Разработка электронной картотеки

Студент гр. 9305 _____ Салауров Е.М.

Преподаватель _____ Перязева Ю.В.

Санкт-Петербург
2020

ЗАДАНИЕ
НА КУРСОВУЮ РАБОТУ

Студент Салауров Е.М.

Группа 9305

Тема работы: Разработка электронной картотеки

Исходные данные:
В качестве исходных данных используется файл, содержащий внутри себя данные для карточек.

Содержание пояснительной записки:
«Аннотация», «Содержание», «Введение»,
«Электронная картотека», «Программная реализация», «Заключение», «Список использованных источников», «Приложение А. Схема вызова функций», «Приложение Б. Схемы функций», «Приложение В. Текст программы».

Предполагаемый объем пояснительной записки:
Не менее 24 страниц.

Дата выдачи задания: 01.04.2020

Дата сдачи реферата: 07.06.2020

Дата защиты реферата: 00.00.2020

Студент _____ Салауров Е.М.

Преподаватель _____ Перязева Ю.В.

АННОТАЦИЯ

В содержание данной курсовой работы входит: теоретическая информация о электронной картотеке, сама программа, реализующая работу, блок-схемы программы и функций, которые мы в ней используем. Кроме этого показаны контрольные примеры.

.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Цель:
законченное поэтапное решение содержательной задачи (постановка задачи, спецификация, выбор структур данных и разработка алгоритма, программная реализация, тестирование).

Основные задачи:

- занесение данных в электронную картотеку;
- внесение изменений (исключение, корректировка, добавление);
- поиск данных по различным признакам;
- сортировку по различным признакам;
- вывод результатов на экран и сохранение на диске.

Предметной областью в данной работе являются коллекционные монеты.

В процессе обработки картотека должна храниться в памяти компьютера в виде списков и массивов структур, связанных указателями.

Для реализации основных операций с электронной картотекой был написан двусвязный список и все необходимые функции.

1. ЭЛЕКТРОННАЯ КАРТОТЕКА

Электронная картотека - это один из способов взаимодействия с базой данных, в которой могут содержаться упорядоченные данные о тех или иных объектах.

Данная программа также позволяет пользователю взаимодействовать с данными, добавляя, изменяя и удаляя их, кроме того доступна сортировка и поиск по нужному параметру структур, поиск программы не чувствительный к регистру.

Кроме этого, в программе предусмотрен ввод некорректных значений, который вместо выхода из программы, позволяет пользователю заново ввести нужное ему значение. Также программа учитывает отсутствие данных в файле, выдавая об этом специальное сообщения пользователю.

2. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

2.1. Описание работы программы

В main находится всего лишь две функции – fill_list, command_selecting, которая, как видно из названия, выводит в консоль интерфейс программы, предлагая пользователю выбрать ту или иную функцию работы с картотекой.

После выбора функции программа выводит пользователю отдельное меню взаимодействия с картотекой, исходя из выбора пользователя.

Сама программа поделена на папку с библиотечными файлами которые отвечают за подключение функций и отдельные файлы формата .c. Т.е. в программе сохранена модульность, это сделано для более удобной координации программиста при работе с проектом, а также позволяет сохранить программе структурированность.

2.2. Описание структур данных

Структура ZNAK:

Название переменной	**Тип переменной**	**Назначение**
face_value	int	Поле с номиналом монеты
currency	char	Поле с наименованием валюты монеты
type_of_money	char	Поле с типом монеты
country	char	Поле с наименованием страны в котором была создана монета
NAME	char	Поле с именем
DATE	int	Массив содержащий дату чеканки монеты

Структура node:

Название переменной	**Тип переменной**	**Назначение**
data	group	Информационное поле
id	int	Переменная номер структуры
prev	node	Указатель на предыдущую структуру в списке(Этого поля нет в кольцевом списке)
next	node	Указатель на следующую структуру в списке

Структура head:

Название переменной	**Тип переменной**	**Назначение**
N	int	Переменная счетчик структур
first	node	Указатель на первую структуру в списке
last	node	Указатель на последнюю структуру в списке

2.3. Описание функций

Функция main():

Описание:

Точка входа в программу. Отвечает за открытие файла, содержащего данные для последующей работы.

Прототип:

int main()

Пример вызова:

main()

Описание переменных:

Название переменной	Тип переменной	Назначение
q	head	Указатель на голову списка

Возвращает значение: 0, если работа программы завершена успешно.

Функция fill_list

Описание:

Считывание информации из файла и забивания его в массив структуры. Пока строка не совпадет с предыдущей, программа ее разделяет и записывает в поля списка**.**

Прототип:

void fill_list(head *q)

Примеры вызова:

fill_list(q)

Описание переменных:

Название переменной	Тип переменной	Назначение
q	head	Указатель на голову списка
str	char	Массив для копирования
message	char	Массив для копирования

Функция enterFromKeyboard

Описание:
Ввод информации в базу данных с клавиатуры: вначало списка или в конец**.**
Прототип:
void enterFromKeyboard(head *q)
Примеры вызова:
enterFromKeyboard(q)
Описание переменных:

Название переменной	Тип переменной	Назначение
q	head	Указатель на голову списка
k	int	Переменная для определения действия
temp	node	Узел списка

Функция split():

Описание:
Функция разделения строки по заданному разделителю.
Каждая строка файла разделяется на элементы промежуточного массива строк s по разделителям с помощью функции и в зависимости от типа поля элемента массива структур выполняется преобразование элемента массива строк в поле отдельной структуры.
Прототип:
void split(char *mes, head *q)
Пример вызова:
split(char *mes, head *q);
Описание переменных:

Название переменной	Тип переменной	Назначение
temp	node	Указатель на структуру
s	char	Массив для копирования данных
n	int	Переменная для корректной записи данных в структуру
k	int	длина выделяемой подстроки

Возвращаемое значение: массив строк.

Функция str_len ():

Описание:
длина строки.
Прототип:
int str_len(char *s)
Пример вызова:
str_len(s)
Описание переменных:

Название переменной	Тип переменной	Назначение
s	char	строка
r	int	количество символов

Функция command_selecting ():

Описание:
работа с пользователем, тоесть программа не завершается пока пользователь не захочет этого (введет нужную команду), тоесть для доабвления элемента - 1, для удаления - 2, для вывода списка - 3, выхода из программы 4. Тоесть работает по не введено 4.
Прототип:
void command_selecting(head *q)
Пример вызова:
command_selecting(q)
Описание переменных:

Название переменной	Тип переменной	Назначение
q	head	голова списка
s	char	команда
f	int	индификатор команды

Функция free_head():

Описание:
очищение головы списка.
Прототип:
void free_head(head *q)
Примеры вызова:
free_head(q)
Описание переменных:

Название переменной	Тип переменной	Назначение
q	head	голова списка

Функция free_node ():

Описание:
очищение узла.
Прототип:
void free_node(node *temp)
Примеры вызова:
free_node(temp)
Описание переменных:

Название переменной	Тип переменной	Назначение
q	head	голова списка

Функция free_list ():

Описание:
очищение всего списка, пока у последнего узла не будет ссылка на NULL.
Прототип:
void free_list(head *q)
Примеры вызова:
free_list(q)
Описание переменных:

Название переменной	Тип переменной	Назначение
p	node	предыдущий узел
temp	node	узел
q	head	голова списка

Функция malloc_node ():

Описание:
выделяет память полям узла.
Прототип:
void malloc_node(node *temp)
Примеры вызова:
malloc_node(temp)
Описание переменных:

Название переменной	Тип переменной	Назначение
temp	node	узел

Функция create_head ():

Описание:
создание головы списка.
Прототип:
head *create_head()
Примеры вызова:
create_head()
Описание переменных:

Название переменной	Тип переменной	Назначение
p	head	голова списка

Функция add_last ():

Описание:
добавление узла в конец списка.
Прототип:
void add_last(head *q)
Примеры вызова:
add_last(q)
Описание переменных:

Название переменной	Тип переменной	Назначение
temp	node	узел
q	head	голова списка

Функция add_first ():

Описание:
добавление узла в начало списка
Прототип:
node *add_first(head *q)
Примеры вызова:
add_first(q)
Описание переменных:

Название переменной	Тип переменной	Назначение
temp	node	узел
q	head	голова списка

Функция delete_first ():

Описание:
удаление первого узла
Прототип:
void delete_first(head *q)
Примеры вызова:
delete_first(q)
Описание переменных:

Название переменной	Тип переменной	Назначение
temp	node	узел
q	head	голова списка

Функция delete_node ():

Описание:
удаление любого узла, кроме первого.
Прототип:
void delete_node(node *p, head *q)
Примеры вызова:
delete_node(p, q)
Описание переменных:

Название переменной	Тип переменной	Назначение
temp	node	узел
q	head	голова списка
p	node	предыдущий узел

Функция output_list_reverse ():

Описание:
Печать базы данных от последней записи к первой.
Прототип:
void output_list_reverse(head *q)
Пример вызова:
output_list_reverse(q)
Описание переменных:

Название переменной	Тип переменной	Назначение
temp	node	узел
q	head	голова списка

Функция create_node():

Описание:
Создание узла в списке
Прототип:
node *create_node (head *q)
Пример вызова:
create_node(head *q);
Описание переменных:

Название переменной	Тип переменной	Назначение
temp	node	Структура для для создания списка

Функция struct_fill():

Описание:
Функция заполнения структуры данными из файла. В массив строк вводятся полученные из simple_split данные.
Прототип:
cars *struct_fill(char **str)
Пример вызова:
ch[i]=struct_fill(s2);
Описание переменных:

Название переменной	Тип переменной	Назначение
str0	group	Структура для распределения в ней данных из массива

Функция print_list():

Описание:
Функция вывода списка. Пока указатель на голову не NULL, выводит данные одной строки и переходит к следующей
Прототип:
void print_list(node *head)
Пример вызова:
print_list(head);
Описание переменных:

Название переменной	Тип переменной	Назначение
p	node	Указатель на голову списка

Функция insert_after():

Описание:

Вставляет новый узел в список при этом учитывает то, что пользователь вводит несуществующий индекс в списке.

Прототип:

void insert_after(node *value, int index)

Пример вызова:

insert_after_DLL(node *lst, int N)

Описание переменных:

Название переменной	**Тип переменной**	**Назначение**
N	int	Переменная для ориентирования в списке
p	node	Указатель на данные из головы массива
temp	node	Структура которую вставляют в список

2.4. Контрольные примеры

Меню программы

```
| | Menu:          |
+-+----------------+
|1| - Print struct |
|2| - Add stuct    |
|3| - Edit stuct   |
|4| - Search struct|
|5| - Sort struct  |
|6| - Swap struct  |
|7| - Delete struct|
|8| - Save struct  |
|0| - Exit         |
Your choice:
```

Список

N	Value	Currency	Type	Country	Name	DATE
1	21	eu	Coin	Eng	-	1098
2	50	rub	coin	USSR	-	1961
3	1	1	1	1	1	1
4	2	2	2	2	2	2
5	10	rub	coin	Russia	Msk	2011
6	11	rub	coin	Russia	Msk	2012
7	12	rub	coin	Russia	Msk	2013
8	13	rub	coin	Russia	Msk	2014
9	14	rub	coin	Russia	Msk	2015
10	15	rub	coin	Russia	Msk	2016
11	17	rub	coin	Russia	Msk	2018
12	18	rub	coin	Russia	Msk	2019
13	19	rub	coin	Russia	Msk	2020
14	20	rub	coin	Russia	Msk	2021
15	21	rub	coin	Russia	Msk	2022
16	22	rub	coin	Russia	Msk	2023

Список после сортировки по номиналу

N	Value	Currency	Type	Country	Name	DATE
1	1	1	1	1	1	1
2	2	2	2	2	2	2
3	10	rub	coin	Russia	Msk	2011
4	11	rub	coin	Russia	Msk	2012
5	12	rub	coin	Russia	Msk	2013
6	13	rub	coin	Russia	Msk	2014
7	14	rub	coin	Russia	Msk	2015
8	15	rub	coin	Russia	Msk	2016
9	17	rub	coin	Russia	Msk	2018
10	18	rub	coin	Russia	Msk	2019
11	19	rub	coin	Russia	Msk	2020
12	20	rub	coin	Russia	Msk	2021
13	21	eu	Coin	Eng	-	1098
14	21	rub	coin	Russia	Msk	2022
15	22	rub	coin	Russia	Msk	2023
16	50	rub	coin	USSR	-	1961

Добавление нового узла

N	Value	Currency	Type	Country	Name	DATE
1	6	6	6	6	6	6
2	1	1	1	1	1	1
3	2	2	2	2	2	2
4	10	rub	coin	Russia	Msk	2011
5	11	rub	coin	Russia	Msk	2012
6	12	rub	coin	Russia	Msk	2013
7	13	rub	coin	Russia	Msk	2014
8	14	rub	coin	Russia	Msk	2015
9	15	rub	coin	Russia	Msk	2016
10	17	rub	coin	Russia	Msk	2018
11	18	rub	coin	Russia	Msk	2019
12	19	rub	coin	Russia	Msk	2020
13	20	rub	coin	Russia	Msk	2021
14	21	eu	Coin	Eng	-	1098
15	21	rub	coin	Russia	Msk	2022
16	22	rub	coin	Russia	Msk	2023
17	50	rub	coin	USSR	-	1961

Перемещение местами узлов списка

N	Value	Currency	Type	Country	Name	DATE
1	6	6	6	6	6	6
2	50	rub	coin	USSR	-	1961
3	2	2	2	2	2	2
4	10	rub	coin	Russia	Msk	2011
5	11	rub	coin	Russia	Msk	2012
6	12	rub	coin	Russia	Msk	2013
7	13	rub	coin	Russia	Msk	2014
8	14	rub	coin	Russia	Msk	2015
9	15	rub	coin	Russia	Msk	2016
10	17	rub	coin	Russia	Msk	2018
11	18	rub	coin	Russia	Msk	2019
12	19	rub	coin	Russia	Msk	2020
13	20	rub	coin	Russia	Msk	2021
14	21	eu	Coin	Eng	-	1098
15	21	rub	coin	Russia	Msk	2022
16	22	rub	coin	Russia	Msk	2023
17	1	1	1	1	1	1

поиск данных в списке

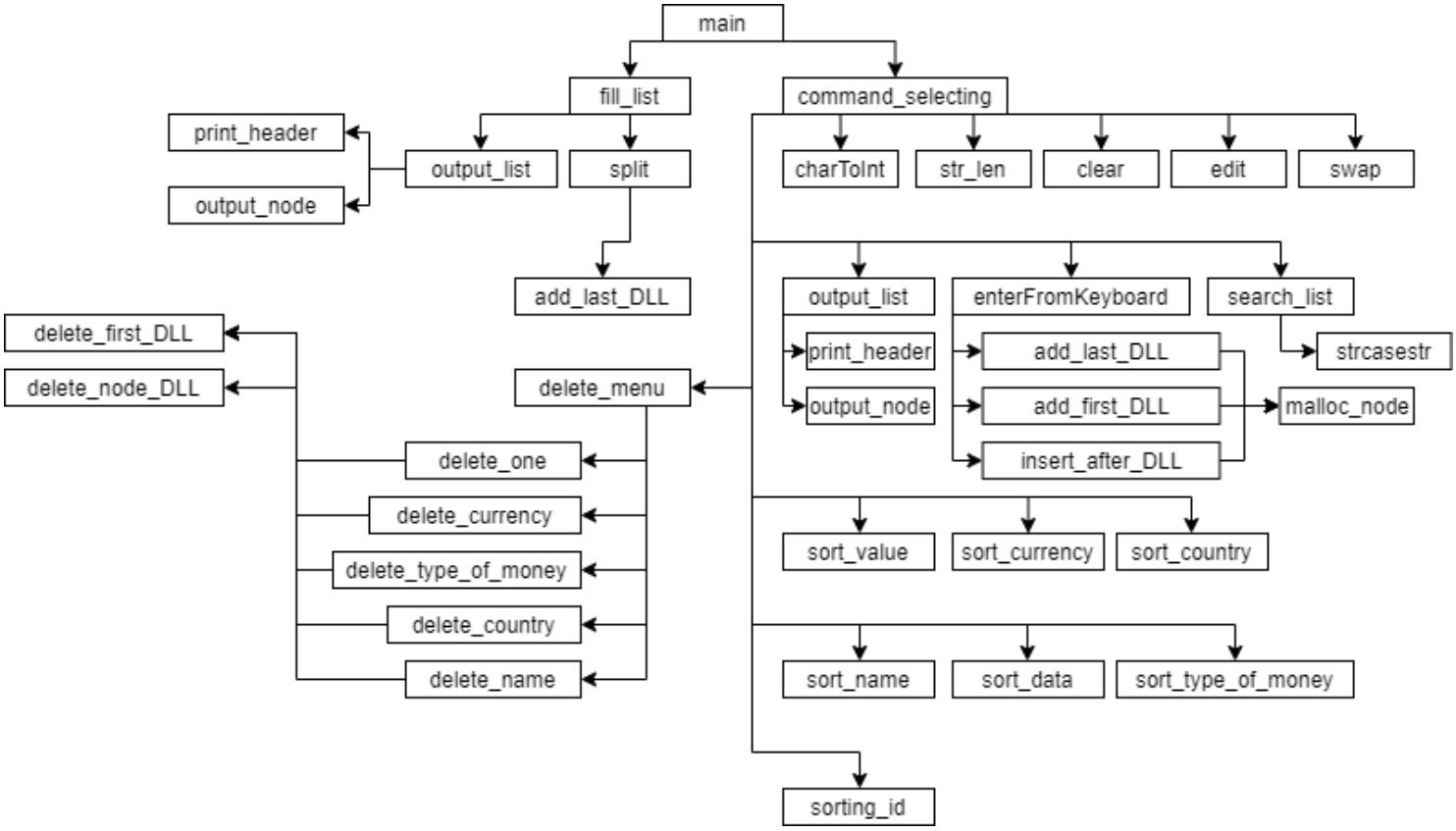
```
Search Data:ussr
| 2|   50|       rub|      coin|      USSR|         -|1961|
```

ЗАКЛЮЧЕНИЕ

В процессе курсовой работы была написана электронная картотека, основывающаяся на двусвязном списке, узлы которой содержат информацию, считанную из csv файла. Сами узлы можно изменять, добавлять, удалять в процессе работы, а также сортировать в порядке возрастания по разным параметрам.

ПРИЛОЖЕНИЕ А

СХЕМА ВЫЗОВА ФУНКЦИЙ

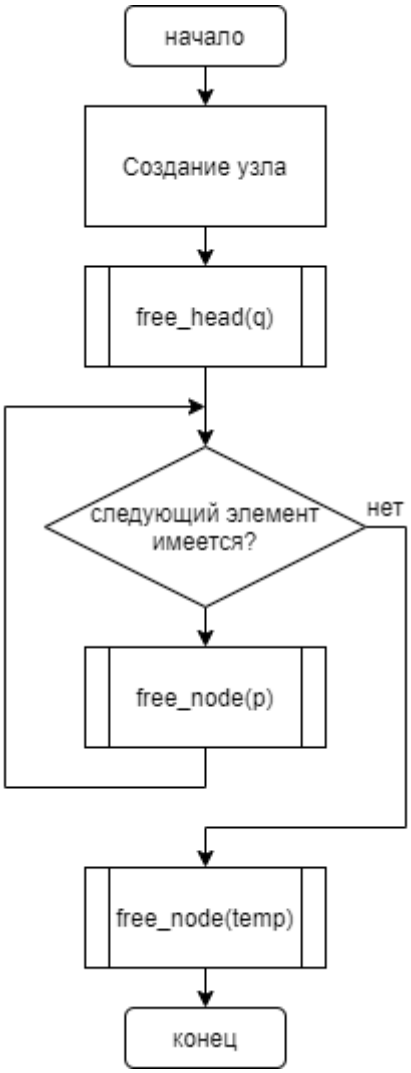


ПРИЛОЖЕНИЕ Б

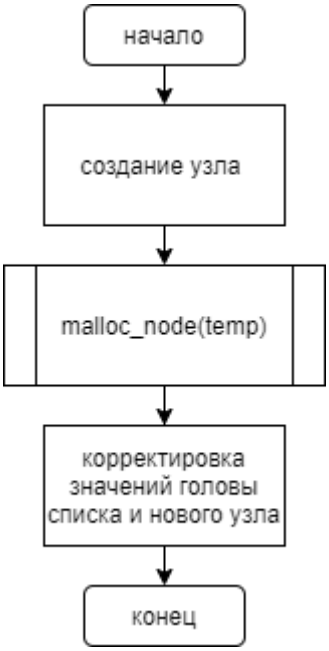
СХЕМЫ ФУНКЦИЙ

add_first_DLL

free_list


начало
Создание узла
free_head(q)
следующий элемент имеется?
нет
free_node(p)
free_node(temp)
конец

create_node_DLL



ПРИЛОЖЕНИЕ С
ТЕКСТ ПРОГРАММЫ

Ссылка на текст программы:
https://github.com/ZZANZZAN/LAB_PROG/tree/master/Kursovaia_sem2

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Двусвязный линейный список // prog-cpp. URL: https://prog-cpp.ru/data-dls/ (дата обращения: 21.05.2020)

2. Связный список. //Википедия. URL: https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D1%8F%D0%B7%D0%BD%D1%8B%D0%B9_%D1%81%D0%BF%D0%B8%D1%81%D0%BE%D0%BA (дата обращения: 20.05.2020)

3. Двусвязный список // learnc https://learnc.info/adt/double_linked_list.html (дата обращения: 21.05.2020)