ГУАП

КАФЕДРА № 34

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | Жиданов К.А. |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2 |
|  |
| по курсу: |
| ЯЗЫКИ ПРОГРАММИРОВАНИЯ |
|  |

РАБОТУ ВЫПОЛНИЛА

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТКА ГР. | 3145 |  |  |  | А.С. Дмитриева |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

**Вариант №1**

Двусвязный список (поиск, удаление, вставка)

**Цель работы**

Реализовать АТД (абстрактный тип данных) в виде пользовательского типа данных и набора функций, реализующих заданные операции. Помимо стандартных интерфейсов (чтение/добавление/поиск/удаление), требуется реализовать чтение/выгрузку данных из файла.

**Ход работы**

1. Двусвязный список – это структура данных, которая состоит из узлов, которые хранят данные и указатели на предыдущий и следующий узел



1. Создаем заголовочный файл для программы

#pragma once

#ifndef \_\_DLL\_\_

#define \_\_DLL\_\_

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

typedef struct linked\_list {

int key;

struct linked\_list\* prev, \* next;

}list\_t;

list\_t\* init();

list\_t\* delel(list\_t\*);

list\_t\* delroot(list\_t\*);

list\_t\* addel(list\_t\*);

list\_t\* findel(list\_t\*, int);

list\_t\* add\_from\_file(list\_t\*, char);

#endif

1. Создаем Си-файл и реализуем в нем следующие действия:
   1. Инициализация списка

list\_t\* init() {

list\_t\* lst;

lst = (list\_t\*)malloc(sizeof(list\_t));

lst->next = NULL;

lst->prev = NULL;

return lst;

};

* 1. Добавление элемента

list\_t\* addel(list\_t\* lst, int n) {

list\_t\* new\_item;

new\_item = (list\_t\*)malloc(sizeof(list\_t));

if (new\_item == NULL) {

return NULL;

}

new\_item->key = n;

new\_item->next = NULL;

while (lst->next != NULL) {

lst = lst->next;

}

lst->next = new\_item;

new\_item->prev = lst;

return new\_item;

};

* 1. Удаление элемента

list\_t\* delel(list\_t\* lst) {

list\_t\* prev, \* next;

prev = lst->prev;

next = lst->next;

if (prev != NULL) {

prev->next = lst->next;

}

if (next != NULL) {

next->prev = lst->prev;

}

free(lst);

return prev;

};

* 1. Поиск элемента

list\_t\* findel(list\_t\* lst, int n) {

if (lst == NULL) {

return NULL;

}

if (n == lst->key) {

return lst;

}

else {

return findel(lst->next, n);

}

};

* 1. Добавление элемента из файла

list\_t\* add\_from\_file(list\_t\* lst, char filename[]) {

FILE\* f;

char buffer[128];

f = fopen(filename, "r");

if (f == NULL) {

printf("Cannot open file '%s'", filename);

return 1;

}

while (!feof(f)) {

fgets(buffer, sizeof(buffer), f);

printf(buffer);

}

return 0;

}

1. Создаем основной Си-файл

int main(int argc, char\*\* argv) {

if (argv[1] != NULL) {

char filename = argv[1];

list\_t\* list = NULL;

list = init();

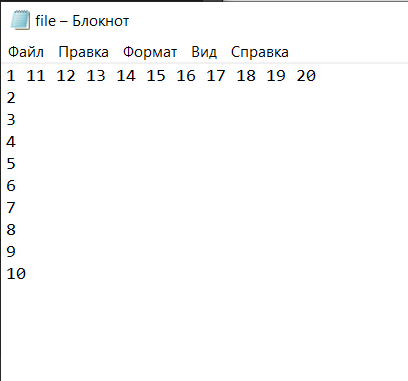
add\_from\_file(list, filename);

}

return 0;

}

1. Создаём файл с данными



**Вывод**

Реализовали двусвязный список на Си