Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа № 6

По дисциплине «Основы алгоритмизации и программирования»

На тему «Динамические структуры данных. Односвязные списки»

Выполнил:

Рауба Арсений

Студент 1 курса 8 группы

Преподаватель: асс. Андронова М.В.

Минск, 2024

В соответствии со своим вариантом разработать программу с использованием односвязного списка по данным, пред-ставленным в таблице ниже.

Программа должна содержать меню с пунктами: добавление элемента, удаление элемента, поиск элемента, вывод списка в консольное окно, запись списка в файл, считывание списка из файла.

Вариант №9



#include <iostream>

#include <fstream>

#include <string>

using namespace std;

struct list {

float num;

list\* next;

};

list\* arrlist = nullptr; // указатель на начало списка

list\* arr; // указатель на текущий элемент

void addElement();

void deleteElement();

void viewAll();

void average();

void write\_to\_file();

void read\_from\_file();

int main()

{

int choice;

do {

cout << "Menu " << endl;

cout << "1. Add element to list." << endl;

cout << "2. Delete element." << endl;

cout << "3. View all elements." << endl;

cout << "4. Average of negative elements" << endl;

cout << "5. Write list to file." << endl;

cout << "6. Read list from file." << endl;

cout << "7. Exit." << endl;

cin >> choice;

switch (choice) {

case 1:

addElement();

break;

case 2:

deleteElement();

break;

case 3:

viewAll();

break;

case 4:

average();

break;

case 5:

write\_to\_file();

break;

case 6:

read\_from\_file();

break;

case 7:

cout << "Exit." << endl;

break;

default:

cout << "Error." << endl;

break;

}

} while (choice != 7);

return 0;

}

//Функция для ввода элемента

void addElement()

{

arr = new list; //Выделение памяти

arr->next = arrlist;

arrlist = arr;

cout << "Enter number:";

cin >> arr->num;

}

//Функция для удаления элемента

void deleteElement()

{

int value;

cin >> value;

list\* temp = arrlist, \* temparrList = nullptr;

arr = arrlist;

while (arr)

{

if (arr->num != value)

{

temp = new list;

temp->next = temparrList;

temparrList = temp;

temp->num = arr->num;

}

arr = arr->next;

}

arrlist = temparrList;

while (temp)

{

arr = temp;

temp = temp->next;

}

delete temp;

}

//Функция для отображения всех элементов списка

void viewAll()

{

arr = arrlist;

while (arr)

{

cout << arr->num << ' ';

arr = arr->next;

}

cout << endl;

}

// Функция для среднего значения

void average()

{

arr = arrlist;

int counter=0;

float avg=0;

while (arr)

{

if (arr->num<0)

avg += arr->num;

arr = arr->next;

counter++;

}

if (avg == 0)

cout << "Element search error." << endl;

else cout << "Average = " << avg/counter << endl;

}

//Функция для записи в файл

void write\_to\_file()

{

ofstream file("file.txt");

arr = arrlist;

while (arr)

{

file << arr->num;

arr = arr->next;

file << ' ';

}

file << endl << endl;

file.close();

}

//Функция для чтения из файла

void read\_from\_file()

{

ifstream file("file.txt");

string a;

cout << endl;

getline(file, a);

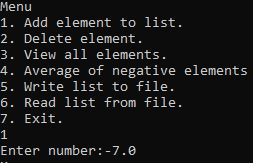
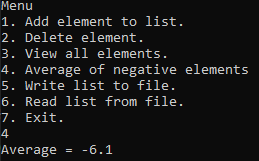
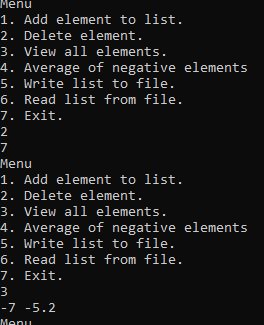
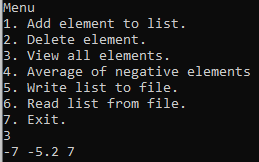
cout << endl << a;

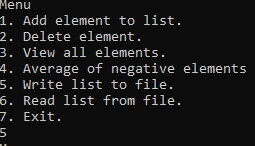
getline(file, a);

cout << endl << a << endl;

file.close();

}

**Доп.Задачи**

**Вариант 1**

****

#include <iostream>

using namespace std;

struct list

{

float number;

list\* next;

};

void insert(list\*&, float);

float del(list\*&, float);

int IsEmpty(list\*); //функция, которая проверяет, пуст ли список

void printList(list\*); //функция вывода

void menu(void); //функция, показывающая меню

void sum(list\*); //функция подсчета суммы чисел

int main()

{

setlocale(LC\_CTYPE, "Russian");

list\* first = NULL;

int choice;

float value;

menu(); // вывести меню

cout << " ? ";

cin >> choice;

while (choice != 4)

{

switch (choice)

{

case 1: cout << "Введите число "; // добавить число в список

cin >> value;

insert(first, value);

printList(first);

break;

case 2: if (!IsEmpty(first)) // удалить число из списка

{

cout << "Введите удаляемое число ";

cin >> value;

if (del(first, value))

{

cout << "Удалено число " << value << endl;

printList(first);

}

else

cout << "Число не найдено" << endl;

}

else

cout << "Список пуст" << endl;

break;

case 3: sum(first); // вычисление среднего значения

break;

default: cout << "Неправильный выбор" << endl;

menu();

break;

}

cout << "? ";

cin >> choice;

}

cout << "Конец" << endl;

return 0;

}

void menu(void) //Вывод меню

{

cout << "Сделайте выбор:" << endl;

cout << " 1 - Ввод числа" << endl;

cout << " 2 - Удаление числа" << endl;

cout << " 3 - Вычисление среднего значения чисел, больших 0" << endl;

cout << " 4 - Выход" << endl;

}

void insert(list\*& p, float value) //Добавление числа value в список

{

list\* newP = new list;

if (newP != NULL) //есть ли место?

{

newP->number = value;

newP->next = p;

p = newP;

}

else

cout << "Операция добавления не выполнена" << endl;

}

float del(list \* &p, float value) // Удаление числа

{

list\* previous, \* current, \* temp;

if (value == p->number)

{

temp = p;

p = p->next; // отсоединить узел

delete temp; //освободить отсоединенный узел

return value;

}

else

{

previous = p;

current = p->next;

while (current != NULL && current->number != value)

{

previous = current;

current = current->next; // перейти к следующему

}

if (current != NULL)

{

temp = current;

previous->next = current->next;

delete(temp);

return value;

}

}

return 0;

}

int IsEmpty(list\* p) //Список пустой? (1-да, 0-нет)

{

return p == NULL;

}

void printList(list \* p) //Вывод списка

{

if (p == NULL)

cout << "Список пуст" << endl;

else

{

cout << "Список:" << endl;

while (p != NULL)

{

cout << "-->" << p->number;

p = p->next;

}

cout << "-->NULL" << endl;

}

}

void sum(list\* p) // Подсчет суммы чисел, больших 0

{

float sm = 0;

bool found = false;

if (p == NULL)

cout << "Список пуст" << endl;

else

{

while (p != NULL)

{

if (p->number > 0)

found = true;

sm = sm + (p->number);

p = p->next;

}

if (!found)

{

cout << "Положительных элементов не найдено" << endl;

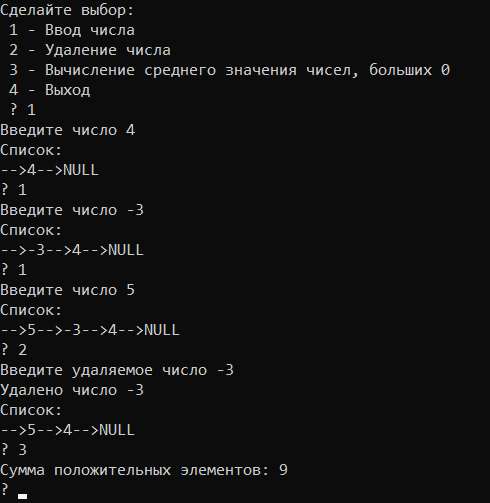
return;

}

cout << "Сумма положительных элементов: " << sm << endl;

}

}

****

**Вариант 2**

****

#include <iostream>

using namespace std;

struct list

{

float number;

list\* next;

};

void insert(list\*&, float);

float del(list\*&, float);

int IsEmpty(list\*); //функция, которая проверяет, пуст ли список

void printList(list\*); //функция вывода

void menu(void); //функция, показывающая меню

void sum(list\*); //функция подсчета среднего значения чисел

int countch = 0;

int main()

{

setlocale(LC\_CTYPE, "Russian");

list\* first = NULL;

int choice;

float value;

menu(); // вывести меню

cout << " ? ";

cin >> choice;

while (choice != 4)

{

switch (choice)

{

case 1: cout << "Введите число "; // добавить число в список

cin >> value;

insert(first, value);

printList(first);

break;

case 2: if (!IsEmpty(first)) // удалить число из списка

{

cout << "Введите удаляемое число ";

cin >> value;

if (del(first, value))

{

cout << "Удалено число " << value << endl;

printList(first);

}

else

cout << "Число не найдено" << endl;

}

else

cout << "Список пуст" << endl;

break;

case 3: sum(first); // вычисление среднего значения

break;

default: cout << "Неправильный выбор" << endl;

menu();

break;

}

cout << "? ";

cin >> choice;

}

cout << "Конец" << endl;

return 0;

}

void menu(void) //Вывод меню

{

cout << "Сделайте выбор:" << endl;

cout << " 1 - Ввод числа" << endl;

cout << " 2 - Удаление числа" << endl;

cout << " 3 - Вычисление среднего значения чисел, больших 0" << endl;

cout << " 4 - Выход" << endl;

}

void insert(list\*& p, float value) //Добавление числа value в список

{

list\* newP = new list;

if (newP != NULL) //есть ли место?

{

newP->number = value;

newP->next = p;

p = newP;

}

else

cout << "Операция добавления не выполнена" << endl;

}

float del(list \* &p, float value) // Удаление числа

{

list\* previous, \* current, \* temp;

if (value == p->number)

{

temp = p;

p = p->next; // отсоединить узел

delete temp; //освободить отсоединенный узел

return value;

}

else

{

previous = p;

current = p->next;

while (current != NULL && current->number != value)

{

previous = current;

current = current->next; // перейти к следующему

}

if (current != NULL)

{

temp = current;

previous->next = current->next;

delete(temp);

return value;

}

countch--;

}

return 0;

}

int IsEmpty(list\* p) //Список пустой? (1-да, 0-нет)

{

return p == NULL;

}

void printList(list \* p) //Вывод списка

{

if (p == NULL)

cout << "Список пуст" << endl;

else

{

cout << "Список:" << endl;

while (p != NULL)

{

cout << "-->" << p->number;

p = p->next;

}

cout << "-->NULL" << endl;

}

}

void sum(list\* p) // Подсчет среднего значения чисел, больших 0

{

float sm = 0;

if (p == NULL)

cout << "Список пуст" << endl;

else

{

while (p != NULL)

{

if (p->number > 0)

countch++;

sm = sm + (p->number);

p = p->next;

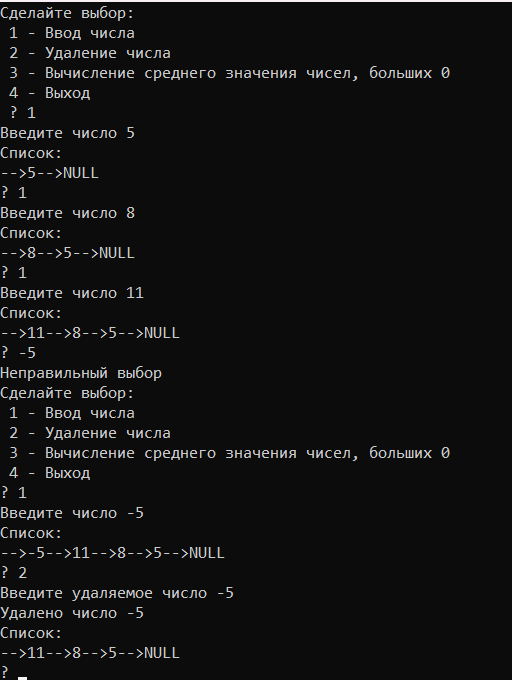
}

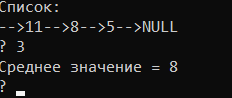
sm = sm / countch;

cout << "Среднее значение = " << sm << endl;

}

}

****

****

**Вариант 3**

****

#include <iostream>

using namespace std;

struct list

{

float number;

list\* next;

};

void insert(list\*&, float);

float del(list\*&, float);

int IsEmpty(list\*);

void printList(list\*);

void menu(void);

void sum(list\*);

int main()

{

setlocale(LC\_CTYPE, "Russian");

list\* first = NULL;

int choice;

float value;

menu();

cout << " ? ";

cin >> choice;

while (choice != 4)

{

switch (choice)

{

case 1:

cout << "Введите число: ";

cin >> value;

insert(first, value);

printList(first);

break;

case 2:

if (!IsEmpty(first))

{

cout << "Введите удаляемое число: ";

cin >> value;

if (del(first, value))

{

cout << "Удалено число " << value << endl;

printList(first);

}

else

cout << "Число не найдено" << endl;

}

else

cout << "Список пуст" << endl;

break;

case 3:

sum(first);

break;

default:

cout << "Неправильный выбор" << endl;

menu();

break;

}

cout << "? ";

cin >> choice;

}

cout << "Конец" << endl;

return 0;

}

void menu(void)

{

cout << "Сделайте выбор:" << endl;

cout << "1 - Ввод числа" << endl;

cout << "2 - Удаление числа" << endl;

cout << "3 - Найти сумму положительных элементов, кратных 5" << endl;

cout << "4 - Выход" << endl;

}

void insert(list\*& p, float value)

{

list\* newP = new list;

if (newP != NULL)

{

newP->number = value;

newP->next = p;

p = newP;

}

else

cout << "Операция добавления не выполнена" << endl;

}

float del(list\*& p, float value)

{

list\* previous, \* current, \* temp;

if (value == p->number)

{

temp = p;

p = p->next;

delete temp;

return value;

}

else

{

previous = p;

current = p->next;

while (current != NULL && current->number != value)

{

previous = current;

current = current->next;

}

if (current != NULL)

{

temp = current;

previous->next = current->next;

delete(temp);

return value;

}

}

return 0;

}

int IsEmpty(list\* p)

{

return p == NULL;

}

void printList(list\* p)

{

if (p == NULL)

cout << "Список пуст" << endl;

else

{

cout << "Список:" << endl;

while (p != NULL)

{

cout << "-->" << p->number;

p = p->next;

}

cout << "-->NULL" << endl;

}

}

void sum(list\* p)

{

float sm = 0;

if (p == NULL)

{

cout << "Список пуст" << endl;

return;

}

bool found = false;

while (p != NULL)

{

if (p->number > 0 && fmod(p->number, 5) == 0)

{

found = true;

sm += p->number;

}

p = p->next;

}

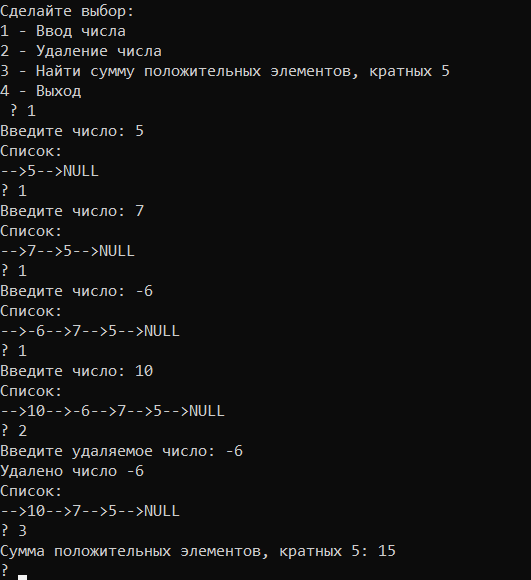
if (!found)

cout << "Положительных элементов, кратных 5, не найдено" << endl;

else

cout << "Сумма положительных элементов, кратных 5: " << sm << endl;

}

****