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:

Студент 1 курса 10 группы

Мандрик Алексей Иванович

Преподаватель: асс. Андронова М.В.

Минск, 2024

В соответствии со своим вариантом разработать программу с использованием ***односвязного списка*** по данным, представленным 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 8**



**Код программы:**

#include <iostream>

#include <fstream>

using namespace std;

// Структура

struct list

{

int number;

list\* next;

};

void menu(void) //Вывод меню

{

cout << "Сделайте выбор:" << endl;

cout << " 1 - Ввод числа" << endl;

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

cout << " 3 - Вычисление суммы отрицательных чисел, кратных 2" << endl;

cout << " 4 - Запись в файл" << endl;

cout << " 5 - Чтение из файла" << endl;

cout << " 0 - Выход" << endl;

}

void insert(list\*& p, int 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;

}

int del(list\*& p, int 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 отрицательных чисел, кратных двум

{

int sm = 0;

if (p == NULL)

cout << "Список пуст" << endl;

else

{

while (p != NULL)

{

if (p->number < 0 && p->number % 2 == 0)

sm = sm + (p->number);

p = p->next;

}

cout << "Сумма = " << sm << endl;

}

}

void toFile(list\*& p) // запись в файл

{

list\* temp = p;

list buf;

ofstream frm("mList.txt"); // Открытие для записи

if (frm.fail()) // Если невозможно открыть

{

cout << "\n Ошибка открытия файла";

exit(1);

}

while (temp)

{

buf = \*temp;

frm.write((char\*)&buf, sizeof(list));

temp = temp->next;

}

frm.close();

cout << "Список записан в файл mList.txt\n";

}

void fromFile(list\*& p) //Считывание из файла

{

list buf, \* first = nullptr;

ifstream frm("mList.txt"); // Открытие для чтения

if (frm.fail()) // Если невозможно открыть

{

cout << "\n Ошибка открытия файла";

exit(1);

}

frm.read((char\*)&buf, sizeof(list));

while (!frm.eof())

{

insert(first, buf.number);

cout << "-->" << buf.number;

frm.read((char\*)&buf, sizeof(list));

}

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

frm.close();

p = first;

cout << "\nСписок считан из файла mList.txt\n";

}

int main()

{

setlocale(LC\_CTYPE, "Russian");

list\* first = NULL;

int choice;

int value;

menu(); // вывести меню

cout << " ? ";

cin >> choice;

while (choice != 0)

{

switch (choice)

{

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

cin >> value;

insert(first, value); // Ввод числа

printList(first); // Вывод списка

break;

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

{

cout << "Введите удаляемое число ";

cin >> value;

if (del(first, value)) // Поиск и удаление нужного числа

{

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

printList(first); // Вывод списка

}

else

cout << "Число не найдено" << endl;

}

else

cout << "Список пуст" << endl;

break;

case 3: sum(first); // вычисление суммы

break;

case 4: toFile(first); // запись в файл

break;

case 5: fromFile(first); // Чтение из файла

break;

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

menu();

break;

}

cout << "? ";

cin >> choice;

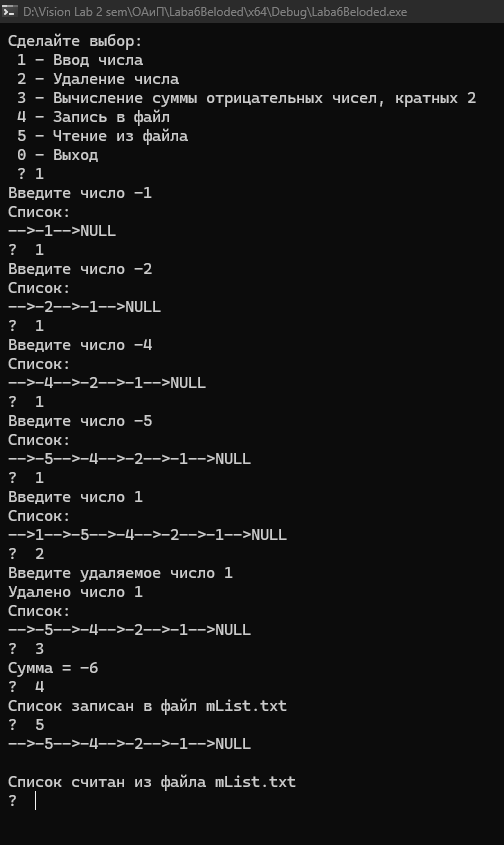
}

cout << "Конец" << endl;

return 0;

}

**Результат выполнения:**



**Дополнительные задания:**

**Вариант 1**

****

**Код программы:**

#include <iostream>

#include <fstream>

using namespace std;

// Структура

struct list

{

int number;

list\* next;

};

void menu(void) //Вывод меню

{

cout << "Сделайте выбор:" << endl;

cout << " 1 - Ввод числа" << endl;

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

cout << " 3 - Вычисление суммы положительных чисел" << endl;

cout << " 4 - Запись в файл" << endl;

cout << " 5 - Чтение из файла" << endl;

cout << " 0 - Выход" << endl;

}

void insert(list\*& p, int 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;

}

int del(list\*& p, int 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 положительных чисел

{

int sm = 0;

if (p == NULL)

cout << "Список пуст" << endl;

else

{

while (p != NULL)

{

if (p->number > 0)

sm = sm + (p->number);

p = p->next;

}

cout << "Сумма = " << sm << endl;

}

}

void toFile(list\*& p) // запись в файл

{

list\* temp = p;

list buf;

ofstream frm("mList.txt"); // Открытие для записи

if (frm.fail()) // Если невозможно открыть

{

cout << "\n Ошибка открытия файла";

exit(1);

}

while (temp)

{

buf = \*temp;

frm.write((char\*)&buf, sizeof(list));

temp = temp->next;

}

frm.close();

cout << "Список записан в файл mList.txt\n";

}

void fromFile(list\*& p) //Считывание из файла

{

list buf, \* first = nullptr;

ifstream frm("mList.txt"); // Открытие для чтения

if (frm.fail()) // Если невозможно открыть

{

cout << "\n Ошибка открытия файла";

exit(1);

}

frm.read((char\*)&buf, sizeof(list));

while (!frm.eof())

{

insert(first, buf.number);

cout << "-->" << buf.number;

frm.read((char\*)&buf, sizeof(list));

}

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

frm.close();

p = first;

cout << "\nСписок считан из файла mList.txt\n";

}

int main()

{

setlocale(LC\_CTYPE, "Russian");

list\* first = NULL;

int choice;

int value;

menu(); // вывести меню

cout << " ? ";

cin >> choice;

while (choice != 0)

{

switch (choice)

{

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

cin >> value;

insert(first, value); // Ввод числа

printList(first); // Вывод списка

break;

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

{

cout << "Введите удаляемое число ";

cin >> value;

if (del(first, value)) // Поиск и удаление нужного числа

{

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

printList(first); // Вывод списка

}

else

cout << "Число не найдено" << endl;

}

else

cout << "Список пуст" << endl;

break;

case 3: sum(first); // вычисление суммы

break;

case 4: toFile(first); // запись в файл

break;

case 5: fromFile(first); // Чтение из файла

break;

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

menu();

break;

}

cout << "? ";

cin >> choice;

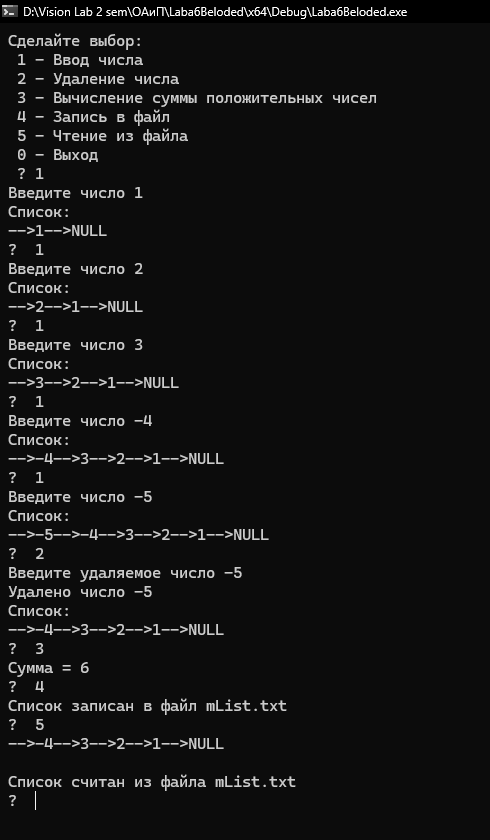
}

cout << "Конец" << endl;

return 0;

}

**Результат выполнения:**

****

**Вариант 3**

****

**Код программы:**

#include <iostream>

#include <fstream>

using namespace std;

// Структура

struct list

{

int number;

list\* next;

};

void menu(void) //Вывод меню

{

cout << "Сделайте выбор:" << endl;

cout << " 1 - Ввод числа" << endl;

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

cout << " 3 - Вычисление суммы отрицательных чисел, кратных 2" << endl;

cout << " 4 - Запись в файл" << endl;

cout << " 5 - Чтение из файла" << endl;

cout << " 0 - Выход" << endl;

}

void insert(list\*& p, int 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;

}

int del(list\*& p, int 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, кратных пяти

{

int sm = 0;

if (p == NULL)

cout << "Список пуст" << endl;

else

{

while (p != NULL)

{

if (p->number > 0 && p->number % 5 == 0)

sm = sm + (p->number);

p = p->next;

}

cout << "Сумма = " << sm << endl;

}

}

void toFile(list\*& p) // запись в файл

{

list\* temp = p;

list buf;

ofstream frm("mList.txt"); // Открытие для записи

if (frm.fail()) // Если невозможно открыть

{

cout << "\n Ошибка открытия файла";

exit(1);

}

while (temp)

{

buf = \*temp;

frm.write((char\*)&buf, sizeof(list));

temp = temp->next;

}

frm.close();

cout << "Список записан в файл mList.txt\n";

}

void fromFile(list\*& p) //Считывание из файла

{

list buf, \* first = nullptr;

ifstream frm("mList.txt"); // Открытие для чтения

if (frm.fail()) // Если невозможно открыть

{

cout << "\n Ошибка открытия файла";

exit(1);

}

frm.read((char\*)&buf, sizeof(list));

while (!frm.eof())

{

insert(first, buf.number);

cout << "-->" << buf.number;

frm.read((char\*)&buf, sizeof(list));

}

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

frm.close();

p = first;

cout << "\nСписок считан из файла mList.txt\n";

}

int main()

{

setlocale(LC\_CTYPE, "Russian");

list\* first = NULL;

int choice;

int value;

menu(); // вывести меню

cout << " ? ";

cin >> choice;

while (choice != 0)

{

switch (choice)

{

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

cin >> value;

insert(first, value); // Ввод числа

printList(first); // Вывод списка

break;

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

{

cout << "Введите удаляемое число ";

cin >> value;

if (del(first, value)) // Поиск и удаление нужного числа

{

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

printList(first); // Вывод списка

}

else

cout << "Число не найдено" << endl;

}

else

cout << "Список пуст" << endl;

break;

case 3: sum(first); // вычисление суммы

break;

case 4: toFile(first); // запись в файл

break;

case 5: fromFile(first); // Чтение из файла

break;

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

menu();

break;

}

cout << "? ";

cin >> choice;

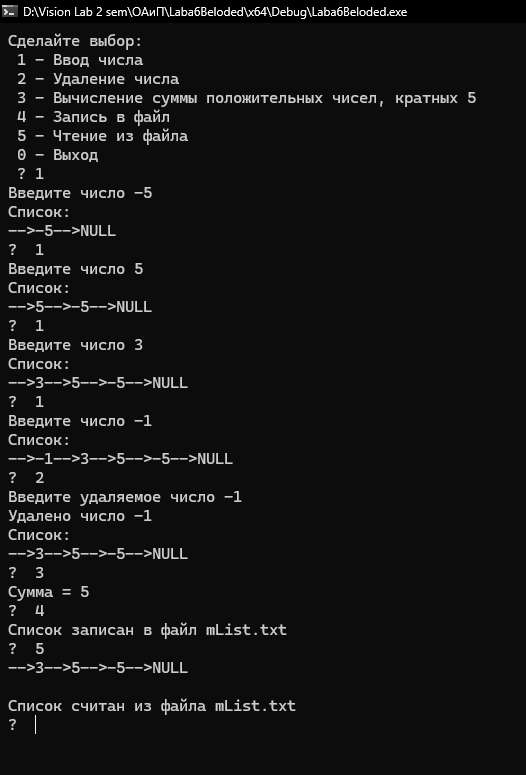
}

cout << "Конец" << endl;

return 0;

}

**Результат выполнения:**



**Вариант 14**

****

**Код программы:**

#include <iostream>

#include <fstream>

using namespace std;

// Структура

struct list

{

int number;

list\* next;

};

void menu(void) //Вывод меню

{

cout << "Сделайте выбор:" << endl;

cout << " 1 - Ввод числа" << endl;

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

cout << " 3 - Вычисление суммы отрицательных чисел, кратных 5" << endl;

cout << " 4 - Запись в файл" << endl;

cout << " 5 - Чтение из файла" << endl;

cout << " 0 - Выход" << endl;

}

void insert(list\*& p, int 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;

}

int del(list\*& p, int 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 отрицательных чисел, кратных пяти

{

int sm = 0;

if (p == NULL)

cout << "Список пуст" << endl;

else

{

while (p != NULL)

{

if (p->number < 0 && p->number % 5 == 0)

sm = sm + (p->number);

p = p->next;

}

cout << "Сумма = " << sm << endl;

}

}

void toFile(list\*& p) // запись в файл

{

list\* temp = p;

list buf;

ofstream frm("mList.txt"); // Открытие для записи

if (frm.fail()) // Если невозможно открыть

{

cout << "\n Ошибка открытия файла";

exit(1);

}

while (temp)

{

buf = \*temp;

frm.write((char\*)&buf, sizeof(list));

temp = temp->next;

}

frm.close();

cout << "Список записан в файл mList.txt\n";

}

void fromFile(list\*& p) //Считывание из файла

{

list buf, \* first = nullptr;

ifstream frm("mList.txt"); // Открытие для чтения

if (frm.fail()) // Если невозможно открыть

{

cout << "\n Ошибка открытия файла";

exit(1);

}

frm.read((char\*)&buf, sizeof(list));

while (!frm.eof())

{

insert(first, buf.number);

cout << "-->" << buf.number;

frm.read((char\*)&buf, sizeof(list));

}

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

frm.close();

p = first;

cout << "\nСписок считан из файла mList.txt\n";

}

int main()

{

setlocale(LC\_CTYPE, "Russian");

list\* first = NULL;

int choice;

int value;

menu(); // вывести меню

cout << " ? ";

cin >> choice;

while (choice != 0)

{

switch (choice)

{

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

cin >> value;

insert(first, value); // Ввод числа

printList(first); // Вывод списка

break;

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

{

cout << "Введите удаляемое число ";

cin >> value;

if (del(first, value)) // Поиск и удаление нужного числа

{

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

printList(first); // Вывод списка

}

else

cout << "Число не найдено" << endl;

}

else

cout << "Список пуст" << endl;

break;

case 3: sum(first); // вычисление суммы

break;

case 4: toFile(first); // запись в файл

break;

case 5: fromFile(first); // Чтение из файла

break;

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

menu();

break;

}

cout << "? ";

cin >> choice;

}

cout << "Конец" << endl;

return 0;

}

**Результат выполнения:**

