Учреждение образования

«БЕЛОРУССКИЙГОСУДАРСТВЕННЫЙТЕХНОЛОГИЧЕСКИЙУНИВЕРСИТЕТ»

Н. Н. Пустовалова, Н. В. Пацей

# ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

**в 2-х частях**

**Часть 2**

**Лабораторный практикум**

**Лабораторная работа № 6. Динамические структуры данных. Односвязные списки**

**Лабораторная работа № 6. Динамические структуры данных. Односвязные списки**

|  |  |
| --- | --- |
| **Задание** | **Программа** |
| 1. Изучить работу с *односвязным списком*, выполнив программу, представленную в правой части. Написать условие задачи. |  |
| 2. В правой части приведена программа, реализующая работу с односвязным списком с использованием *функций пользователя*.  Выполнив программу, изучить способы передачи параметров в функции. |  |

#include <iostream>

using namespace std;

struct list

{

float number;

list\* next;

};

void insert(list\*&, float); //функция добавления элемента, передается адрес списка и символ, который добавляется

float del(list\*&, float); //функция удаления, передается адрес списка и сим-вол, который удаляется

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

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

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

void sum7(list\*); //функция подсчета суммы чисел, больших 7

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:

sum7(first); // вычисление суммы

break;

default:

cout << "Неправильный выбор" << endl;

menu();

break;

}

cout << "Что дальше? ";

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 суммы чисел, больших 7" << 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 sum7(list\* p) // Подсчет суммы чисел, больших 7

{

float sm = 0;

if (p == NULL)

cout << "Список пуст" << endl;

else

{

while (p != NULL)

{

if (p->number > 7)

sm = sm + (p->number);

p = p->next;

}

cout << "Сумма = " << sm << endl;

}

}

|  |  |
| --- | --- |
| 3. В правой части приведена программа, реализующая работу с односвязным списком.  Выполнив программу, изучить способы *записи списка в файл* и *считывания* из файла. |  |

#include <iostream>

#include <fstream>

using namespace std;

struct list

{

char symbol;

list\* next;

};

void insert(list\*& p, char value); //Добавление символа в начало списка

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

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

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

void menu(void); //Вывод меню

int main()

{

setlocale(LC\_CTYPE, "Russian");

list\* first = nullptr;

int choice;

char value;

menu(); // вывести меню

cout << " ? ";

cin >> choice;

while (choice != 4)

{

switch (choice)

{

case 1:

cout << "Введите символ ";

cin >> value;

insert(first, value);

printList(first);

break;

case 2:

toFile(first);

break;

case 3:

fromFile(first);

break;

default:

cout << "Неправильный выбор" << endl;

menu();

break;

}

cout << "? ";

cin >> choice;

}

return 0;

}

void insert(list\*& p, char value) //Добавление символа в начало списка

{

list\* newP = new list;

if (newP != NULL) //есть ли место?

{

newP->symbol = value;

newP->next = p;

p = newP;

}

else

{

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

}

}

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

{

if (p == NULL)

cout << "Список пуст" << endl;

else

{

cout << "Список:" << endl;

while (p != NULL)

{

cout << "-->" << p->symbol;

p = p->next;

}

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

}

}

void toFile(list\*& p)

{

list\* temp = p;

list buf;

ofstream frm("mList.dat");

if (frm.fail())

{

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.dat\n";

}

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

{

list buf, \* first = nullptr;

ifstream frm("mList.dat");

if (frm.fail())

{

cout << "\n Ошибка открытия файла";

exit(1);

}

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

while (!frm.eof())

{

insert(first, buf.symbol);

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

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

}

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

frm.close();

p = first;

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

}

void menu(void) //Вывод меню

{

cout << "Сделайте выбор:" << endl;

cout << " 1 - Ввод символа" << endl;

cout << " 2 - Запись списка в файл" << endl;

cout << " 3 - Чтение списка из файла" << endl;

cout << " 4 - Выход" << endl;

}

|  |  |
| --- | --- |
| 4. 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.  **Вар.1:**  Создать список, содержащий элементы целого типа. Найти сумму положительных элементов или выдать сообщение, что положительных элементов нет. |  |

#include <iostream>

using namespace std;

struct list

{

int value;

list\* next;

};

void printline()

{

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

}

void showmenu()

{

printline();

cout << "Меню:\n" << "1 - Добавить новый элемент в список\n";

cout << "2 - Вывод списка\n" << "3 - Удалить элемент из списка\n";

cout << "4 - Поиск элемента\n" << "5 - Записать список в файл\n";

cout << "6 - Считать список из файла\n" << "7 - Сумма положительных элементов списка\n";

cout << "8 - Вывод меню\n" << "0 - Выход из программы\n";

printline();

}

void inputel(list\*& first)

{

int el;

list\* newel = new list;

cout << "Введите элемент который хотите добавить: ";

cin >> el;

newel->value = el;

newel->next = first;

first = newel;

}

void showlist(list\* first)

{

if (first == nullptr)

{

cout << "Список пуст\n";

}

while (first != nullptr)

{

cout << first->value << "-->";

first = first->next;

}

cout << "NULL" << endl;

}

bool delel(list\*& first)

{

bool isElDel = false;

list\* temp = first;

list\* previous = first, \* following = first;

int el;

cout << "Введите значение элемента который хотите удалить: ";

cin >> el;

if (first->value == el)

{

first = first->next;

delete(temp);

}

else

{

while (following->value != el && following != NULL)

{

previous = following;

following = following->next;

}

if (following == NULL)

{

cout << "Заданного значения нет в списке\n";

}

else

{

temp = following;

previous->next = following->next;

delete(temp);

isElDel = true;

return isElDel;

}

}

return isElDel;

}

void searchel(list\* first)

{

int i = 1;

int searchnum;

bool resultOfSearch = false;

cout << "Введите число которое хотите найти в списке: ";

cin >> searchnum;

while (first != nullptr)

{

if (first->value == searchnum)

{

cout << "Ваше число находится на " << i << " позиции с конца, ";

if (searchnum > 0)

{

cout << "и является положительным\n";

}

else if (searchnum < 0)

{

cout << "и является отрицательным\n";

}

else

{

cout << "И является нулем\n";

}

resultOfSearch = true;

}

i++;

first = first->next;

}

if (!resultOfSearch)

{

cout << "Заданного чиcла нет в списке\n";

}

}

void fileinput(list\* first)

{

FILE\* fdata;

int writedata;

if (!fopen\_s(&fdata, "data.bin", "ab"))

{

while (first != nullptr)

{

writedata = first->value;

fwrite(&writedata, sizeof(int), 1, fdata);

first = first->next;

}

fclose(fdata);

}

else

{

cout << "Не удалось открыть файл для записи\n";

}

}

void outputFromFile(list\*& first,int el)

{

list\* newel = new list;

newel->value = el;

newel->next = first;

first = newel;

}

void fileoutput(list\*& first)

{

FILE\* fdata;

int el;

if (!fopen\_s(&fdata, "data.bin", "rb"))

{

while (fread(&el, sizeof(int), 1, fdata))

{

outputFromFile(first, el);

}

fclose(fdata);

}

}

bool sumpositive(list\* first, int& sum)

{

while (first != nullptr)

{

if (first->value > 0)

{

sum += first->value;

}

first = first->next;

}

if (sum != 0)

{

return true;

}

else

{

return false;

}

}

int main()

{

int userInput;

list\* first = nullptr;

setlocale(LC\_CTYPE, "Russian");

showmenu();

do

{

cin >> userInput;

switch (userInput)

{

case 1:

inputel(first);

cout << "Элемент успешно добавлен:\n";

showlist(first);

break;

case 2:

showlist(first);

break;

case 3:

if (delel(first))

{

cout << "Элемент успешно удален:\n";

showlist(first);

}

else

{

cout << "Ошибка удаления элемента из списка!\n";

}

break;

case 4:

searchel(first);

break;

case 5:

fileinput(first);

break;

case 6:

fileoutput(first);

showlist(first);

break;

case 7:

{

int sum = 0;

if (sumpositive(first, sum))

{

cout << "Сумма положительных элементов = " << sum << endl;

}

else

{

cout << "В списке нет положительных элементов\n";

}

break;

}

case 8:

showmenu();

break;

case 0:

break;

default:

cout << "Ошибка ввода\n";

break;

}

cout << "Что дальше?\n";

} while (userInput != 0);

}