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:

Федорович Вадим

Студент 1 курса 8 группы

Преподаватель: асс. Андронова М.В.

Минск, 2024

Основа:



#include <iostream>

#include <fstream>

using namespace std;

struct list

{ int number;

list \*next;

};

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

int del(list \*&, int); //функция удаления, передается адрес списка и символ, который удаляется

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

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

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

void sum5(list \*); //функция подсчета кратных 5

void sum\_neg(list \*); //функция подсчета отрицательных элементов

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

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

int main()

{ setlocale(LC\_CTYPE, "Russian");

list \*first = NULL;

int choice;

int value;

menu(); // вывести меню

cout<<" ? ";

cin>>choice;

while (choice != 8)

{ 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: sum5(first); // вычисление суммы кратных 5

break;

case 4: sum\_neg(first); // вычисление суммы отрицательных элементов

break;

case 5: toFile(first);

break;

case 6: fromFile(first);

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 суммы чисел кратных 5"<<endl;

cout<<" 4 - Вычисление суммы отрицательных элементов "<<endl;

cout<<" 5 - Запись в файл "<<endl;

cout<<" 6 - Чтение файла "<<endl;

cout<<" 5 - 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 sum5(list \*p) // Подсчет суммы чисел кратных 5

{

int sm = 0;

if (p == NULL)

cout<<"Список пуст"<<endl;

else

{ while (p!= NULL)

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

sm = sm + (p->number);

p = p->next;

}

cout<<"Сумма = "<<sm<<endl;

}

}

void sum\_neg(list\* p) // Подсчет суммы отрицательных элементов

{

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)

{

ofstream frm("mList.dat", ios::binary); // Открываем файл в двоичном режиме

if (!frm.is\_open())

{

cout << "\nОшибка открытия файла" << endl;

return;

}

while (p != nullptr)

{

frm.write(reinterpret\_cast<char\*>(p), sizeof(list));

p = p->next;

}

frm.close();

cout << "Список записан в файл mList.dat" << endl;

}

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

{

ifstream frm("mList.dat", ios::binary); // Открываем файл в двоичном режиме

if (!frm.is\_open())

{

cout << "\nОшибка открытия файла" << endl;

return;

}

list\* first = nullptr;

list\* last = nullptr;

list\* newNode = nullptr;

list buf;

while (frm.read(reinterpret\_cast<char\*>(&buf), sizeof(list)))

{

newNode = new list;

newNode->number = buf.number;

newNode->next = nullptr;

if (first == nullptr)

{

first = newNode;

last = newNode;

}

else

{

last->next = newNode;

last = newNode;

}

}

frm.close();

p = first;

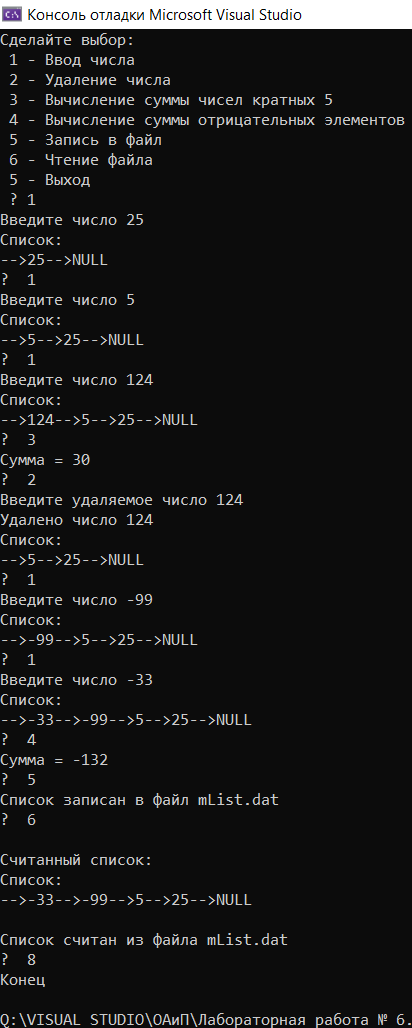
// Выводим считанный список

cout << "\nСчитанный список:" << endl;

printList(p);

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

}



Допы:



#include <iostream>

#include <fstream>

using namespace std;

struct list

{

int number;

list\* next;

};

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

int del(list\*&, int); //функция удаления, передается адрес списка и символ, который удаляется

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

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

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

void sum2(list\*); //функция подсчета кратных 2

void sum\_neg(list\*); //функция подсчета отрицательных элементов

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

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

int main()

{

setlocale(LC\_CTYPE, "Russian");

list\* first = NULL;

int choice;

int value;

menu(); // вывести меню

cout << " ? ";

cin >> choice;

while (choice != 8)

{

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: sum2(first); // вычисление суммы кратных 5

break;

case 4: sum\_neg(first); // вычисление суммы отрицательных элементов

break;

case 5: toFile(first);

break;

case 6: fromFile(first);

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 суммы чисел кратных 2" << endl;

cout << " 4 - Вычисление суммы отрицательных элементов " << endl;

cout << " 5 - Запись в файл " << endl;

cout << " 6 - Чтение файла " << endl;

cout << " 8 - 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 sum2(list\* p) // Подсчет суммы чисел кратных 5

{

int sm = 0;

if (p == NULL)

cout << "Список пуст" << endl;

else

{

while (p != NULL)

{

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

sm = sm + (p->number);

p = p->next;

}

cout << "Сумма = " << sm << endl;

}

}

void sum\_neg(list\* p) // Подсчет суммы отрицательных элементов

{

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)

{

ofstream frm("mList.dat", ios::binary); // Открываем файл в двоичном режиме

if (!frm.is\_open())

{

cout << "\nОшибка открытия файла" << endl;

return;

}

while (p != nullptr)

{

frm.write(reinterpret\_cast<char\*>(p), sizeof(list));

p = p->next;

}

frm.close();

cout << "Список записан в файл mList.dat" << endl;

}

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

{

ifstream frm("mList.dat", ios::binary); // Открываем файл в двоичном режиме

if (!frm.is\_open())

{

cout << "\nОшибка открытия файла" << endl;

return;

}

list\* first = nullptr;

list\* last = nullptr;

list\* newNode = nullptr;

list buf;

while (frm.read(reinterpret\_cast<char\*>(&buf), sizeof(list)))

{

newNode = new list;

newNode->number = buf.number;

newNode->next = nullptr;

if (first == nullptr)

{

first = newNode;

last = newNode;

}

else

{

last->next = newNode;

last = newNode;

}

}

frm.close();

p = first;

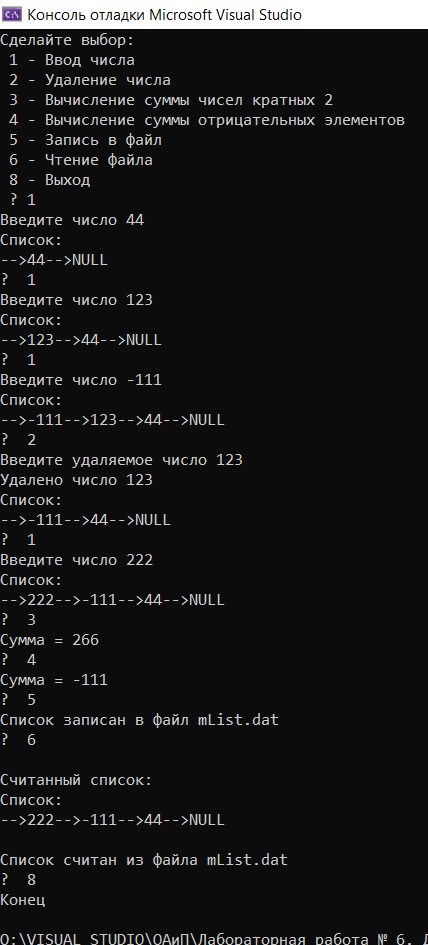
// Выводим считанный список

cout << "\nСчитанный список:" << endl;

printList(p);

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

}





#include <iostream>

#include <fstream>

using namespace std;

struct list

{

int number;

list\* next;

};

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

int del(list\*&, int); //функция удаления, передается адрес списка и символ, который удаляется

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

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

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

void sum2(list\*); //функция подсчета кратных 2

void sum(list\*); //функция подсчета отрицательных элементов

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

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

int main()

{

setlocale(LC\_CTYPE, "Russian");

list\* first = NULL;

int choice;

int value;

menu(); // вывести меню

cout << " ? ";

cin >> choice;

while (choice != 8)

{

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: sum2(first); // вычисление суммы кратных 5

break;

case 4: sum(first); // вычисление суммы отрицательных элементов

break;

case 5: toFile(first);

break;

case 6: fromFile(first);

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 суммы чисел кратных 5" << endl;

cout << " 4 - Вычисление суммы положительных элементов " << endl;

cout << " 5 - Запись в файл " << endl;

cout << " 6 - Чтение файла " << endl;

cout << " 8 - 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 sum2(list\* p) // Подсчет суммы чисел кратных 5

{

int sm = 0;

if (p == NULL)

cout << "Список пуст" << endl;

else

{

while (p != NULL)

{

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

sm = sm + (p->number);

p = p->next;

}

cout << "Сумма = " << sm << endl;

}

}

void sum(list\* p) // Подсчет суммы отрицательных элементов

{

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)

{

ofstream frm("mList.dat", ios::binary); // Открываем файл в двоичном режиме

if (!frm.is\_open())

{

cout << "\nОшибка открытия файла" << endl;

return;

}

while (p != nullptr)

{

frm.write(reinterpret\_cast<char\*>(p), sizeof(list));

p = p->next;

}

frm.close();

cout << "Список записан в файл mList.dat" << endl;

}

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

{

ifstream frm("mList.dat", ios::binary); // Открываем файл в двоичном режиме

if (!frm.is\_open())

{

cout << "\nОшибка открытия файла" << endl;

return;

}

list\* first = nullptr;

list\* last = nullptr;

list\* newNode = nullptr;

list buf;

while (frm.read(reinterpret\_cast<char\*>(&buf), sizeof(list)))

{

newNode = new list;

newNode->number = buf.number;

newNode->next = nullptr;

if (first == nullptr)

{

first = newNode;

last = newNode;

}

else

{

last->next = newNode;

last = newNode;

}

}

frm.close();

p = first;

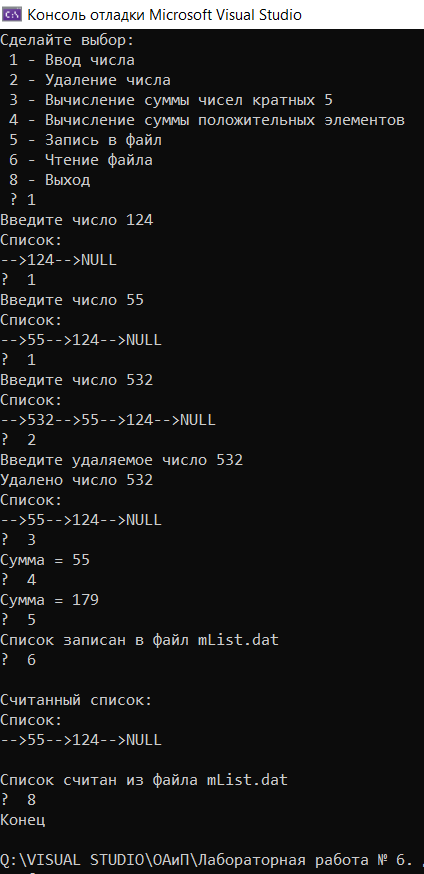
// Выводим считанный список

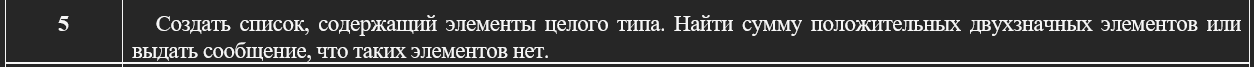
cout << "\nСчитанный список:" << endl;

printList(p);

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

}





#include <iostream>

#include <fstream>

using namespace std;

struct list

{

int number;

list\* next;

};

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

int del(list\*&, int); //функция удаления, передается адрес списка и символ, который удаляется

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

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

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

void sum2(list\*); //функция подсчета положительных двухзначных элементов

void sum(list\*);

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

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

int main()

{

setlocale(LC\_CTYPE, "Russian");

list\* first = NULL;

int choice;

int value;

menu(); // вывести меню

cout << " ? ";

cin >> choice;

while (choice != 8)

{

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: sum2(first); // вычисление суммы

break;

case 4: sum(first); // вычисление суммы отрицательных элементов

break;

case 5: toFile(first);

break;

case 6: fromFile(first);

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 суммы положительных двухзначных элементов " << endl;

cout << " 4 - Вычисление суммы положительных элементов" << endl;

cout << " 5 - Запись в файл " << endl;

cout << " 6 - Чтение файла " << endl;

cout << " 8 - 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 sum2(list\* p) // Подсчет суммы

{

int sm = 0;

if (p == NULL)

cout << "Список пуст" << endl;

else

{

while (p != NULL)

{

if (p->number >= 10 && p->number <= 100)

sm = sm + (p->number);

p = p->next;

}

cout << "Сумма = " << sm << endl;

}

}

void sum(list\* p) // Подсчет суммы

{

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)

{

ofstream frm("mList.dat", ios::binary); // Открываем файл в двоичном режиме

if (!frm.is\_open())

{

cout << "\nОшибка открытия файла" << endl;

return;

}

while (p != nullptr)

{

frm.write(reinterpret\_cast<char\*>(p), sizeof(list));

p = p->next;

}

frm.close();

cout << "Список записан в файл mList.dat" << endl;

}

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

{

ifstream frm("mList.dat", ios::binary); // Открываем файл в двоичном режиме

if (!frm.is\_open())

{

cout << "\nОшибка открытия файла" << endl;

return;

}

list\* first = nullptr;

list\* last = nullptr;

list\* newNode = nullptr;

list buf;

while (frm.read(reinterpret\_cast<char\*>(&buf), sizeof(list)))

{

newNode = new list;

newNode->number = buf.number;

newNode->next = nullptr;

if (first == nullptr)

{

first = newNode;

last = newNode;

}

else

{

last->next = newNode;

last = newNode;

}

}

frm.close();

p = first;

// Выводим считанный список

cout << "\nСчитанный список:" << endl;

printList(p);

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

} 