Институт Космических и Информационных технологий Сибирского Федерального Университета  
**Факультет Информатики и вычислительной техникиКафедра Вычислительной техники**

**Отчёт по лабораторной работе №8**

**«Динамические двусвязные списки»**

Выполнил студент гр. КИ

Проверила:

Красноярск 2015

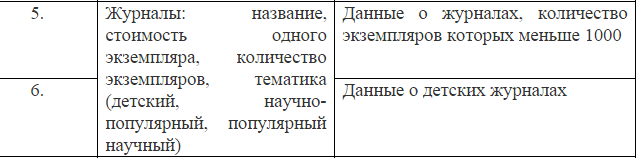
**Цели работы:**

Получить практические навыки решения задач с использованием динамических двусвязных списков.

**Задание:**

С помощью текстового редактора создать файл, в котором содержатся данные, указанные в таблице №1 (в соответствии вариантами). Данные о каждом объекте располагаются в отдельной строке и разделены пробелами.

Разработать, отладить, продемонстрировать и защитить преподавателю графическую схему алгоритма и программу для решения следующей задачи: прочитать из текстового файла данные в динамический двусвязный список, отфильтровать данные в соответствии с указаниями в таблице №1 и сохранить отфильтрованные данные в двоичном файле (имя файла вводится с клавиатуры). Вывести содержимое текстового и двоичного файлов на монитор в виде таблицы.



**Код программы:**

#include <iostream>

#include <fstream>

#include <string>

#include <clocale>

using namespace std;

struct journal

{

string

name,

theme;

int

kol,

coast;

journal \*rlink, \*llink;

};

//Функция записи двусвязного списка в бинарный файл слева напpаво

void llist (journal \*left , FILE \*fbw)

{

journal \*k = left;

while (k != NULL)

{

fwrite (&k -> name, sizeof(string), 1, fbw);

fwrite (&k -> coast, sizeof(int), 1, fbw);

fwrite (&k -> kol, sizeof(int), 1, fbw);

fwrite (&k -> theme, sizeof(string), 1, fbw);

k = k->rlink;

}

}

int main ()

{

setlocale(LC\_ALL, "russian");

ifstream ft("laba7.txt");

journal \*right1, \*left1, \*k1, \*q1, // right1(2) и left1(2) - правые и левые указатели

\*right2, \*left2, \*k2, \*q2; // k1(2) и q1(2) - рабочие указатели, q1(2) - второй рабочий указатель хранит адрес предпоследнего узла

string namen,themen;

int coastn, koln;

// Создание первых пустых узлов

k1 = new journal;

k1 -> rlink = NULL;

k1 -> llink = NULL;

right1 = k1;

q1 = k1;

k2 = new journal;

k2 -> rlink = NULL;

k2 -> llink = NULL;

right2 = k2;

q2 = k2;

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

while(ft >> namen >> coastn >> koln >> themen)

{

cout.setf(ios::left);

cout.width(15);

cout << namen;

cout.width(5);

cout << coastn;

cout.width(20);

cout << koln;

cout.width(20);

cout << themen << endl;

// Создание остальных узлов, заполнение и фильтрация

if (koln<1000)

{

k1 = new journal;

k1 -> name = namen;

k1 -> coast = coastn;

k1 -> kol = koln;

k1 -> theme = themen;

k1 -> rlink = q1;

q1 -> llink = k1;

q1 = k1;

}

else

{

k2 = new journal;

k2 -> name = namen;

k2 -> coast = coastn;

k2 -> kol = koln;

k2 -> theme = themen;

k2 -> rlink = q2;

q2 -> llink = k2;

q2 = k2;

}

}

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

q1 -> llink = NULL;

left1 = q1;

q2 -> llink = NULL;

left2 = q2;

//Удаление пустых правых узлов

k1 = right1;

q1 = k1;

right1 = k1->llink;

right1 -> rlink = NULL;

k1 = NULL;

delete q1;

k2 = right2;

q2 = k2;

right2 = k2->llink;

right2 -> rlink = NULL;

k2 = NULL;

delete q2;

FILE \*fbw1, \*fbw2;

char namef1[11], namef2[11];

cout << "Имя для журналов меньше 1000:" << endl;

cin >> namef1;

cout << "Для остальных:" << endl;

cin >> namef2;

fbw1 = fopen(namef1, "wb");

fbw2 = fopen(namef2, "wb");

llist(left1, fbw1);

llist(left2, fbw2);

fclose(fbw1);

fclose(fbw2);

FILE \*fbr1, \*fbr2;

fbr1 = fopen(namef1, "rb");

fbr2 = fopen(namef2, "rb");

rewind(fbr1);

rewind(fbr2);

cout <<endl << "Журналы меньше 1000:" << endl;

k1 = new journal;

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

// Чтение двусвязного списка из бинарного файла

while (fread (&k1 ->name, sizeof(string), 1, fbr1) && fread (&k1 ->coast, sizeof(int), 1, fbr1) && fread (&k1 ->kol, sizeof(int), 1, fbr1) && fread (&k1 ->theme, sizeof(string), 1, fbr1))

{

k1 -> llink = right1;

cout.setf(ios::left);

cout.width(15);

cout << k1 -> name;

cout.width(5);

cout << k1 -> coast;

cout.width(20);

cout << k1 -> kol;

cout.width(20);

cout << k1 -> theme << endl;

}

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

cout << endl << "Другие журналы:" << endl;

k2 = new journal;

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

// Чтение двусвязного списка из бинарного файла

while (fread (&k2 ->name, sizeof(string), 1, fbr2) && fread (&k2 ->coast, sizeof(int), 1, fbr2) && fread (&k2 ->kol, sizeof(int), 1, fbr2) && fread (&k2 ->theme, sizeof(string), 1, fbr2))

{

k2 -> llink = right2;

cout.setf(ios::left);

cout.width(15);

cout << k2 -> name;

cout.width(5);

cout << k2 -> coast;

cout.width(20);

cout << k2 -> kol;

cout.width(20);

cout << k2 -> theme << endl;

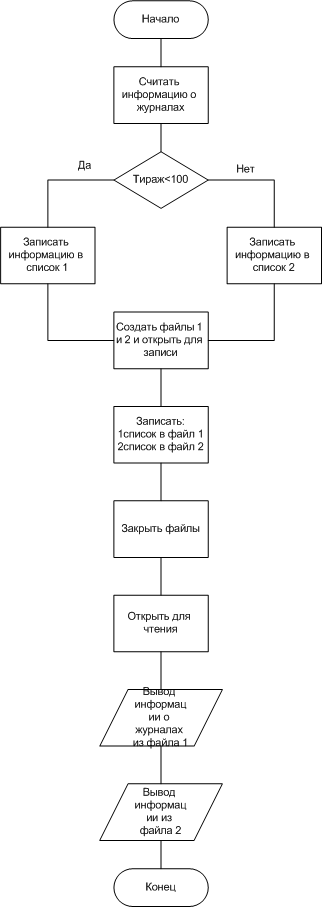
}

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

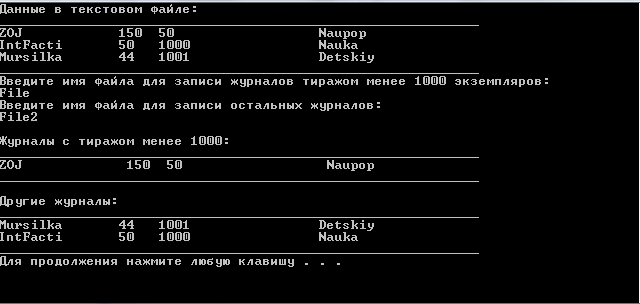
system ("pause");

return 0;

}

****

**Протокол отладки:**



**Вывод:** Были освоены приемы работы с динамическими двусвязными списками в c++.