МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего

профессионального образования

«Юго-Западный государственный университет»  
Кафедра программной инженерии

Лабораторная работа №2  
по дисциплине «Алгоритмы и структуры данных»

Программирование динамических списков

Выполнил: Студент группы ПО-41б

Алябьев Сергей

Проверил: Доцент

Белова Т.М.

Курск, 2015

Задание :

Реализовать логическую операцию И на односвязных линейных списках.

Переменные:

FIFO – структура;

\*head1,\*head2,\*head3 – указатель на первый элемент списка;

\*p1,\*p2,\*p3,\*p1\_d,\*p2\_d – указатели на структуру FIFO;

\*spisok,\*spisok1,\*spisok2,\*spisok3 – указатель на элемент списка;

n1,n2,n3 – количество элементов в списке;

Алгоритм

Создание элемента списка:

spisok->family=Edit1->Text;

spisok->code=Edit2->Text;

n++;

Да

head

Нет

p=head=spisok;

p->next=spisok;

p=spisok;

Вывод списка:

FIFO \*spisok=head;

Да

n

Нет

Да

spisok!=p->next

Memo1->Lines->Add(spisok->family); Memo1->Lines->Add(spisok->code); Memo1->Lines->Add(" "); spisok=spisok->next;

Удаление элемента списка:

p1\_d=head1;

Да

p1!=p1\_d

Нет

int i=1;i<n-1;i++

Head1=NULL; delete p1\_d

p1\_d=p1\_d->next;

delete p1;

p1=p1\_d;

p1\_d->next=NULL;

n1--;

Объединение списков:

FIFO \*spisok1=head1;

FIFO \*spisok3;

FIFO \*spisok2=head2;

int i=0;i<n1;i++

int j=0;j<n2;j++

spisok1->family == spisok2->family && spisok1->code == spisok2->code

Нет

Да

spisok3 = new FIFO; spisok3->family=spisok1->family; spisok3->code=spisok1->code;

head3

Нет

Да

p3->next=spisok3; p3=spisok3;

p3=head3=spisok3;

n3++;

1

4

2

3

11

4

2

3

spisok1=spisok1->next;

Текст программы

Unit1.cpp:

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit1.h"

#include "Unit2.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm1 \*Form1;

int n3(0);

FIFO \*head1,\*head2,\*head3,\*p1,\*p2,\*p3,\*p1\_d,\*p2\_d;

//---------------------------------------------------------------------------

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button3Click(TObject \*Sender)

{

Close();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button4Click(TObject \*Sender) //Вывод

{

FIFO \*spisok=head1;

if(n1)

while (spisok!=p1->next){

Memo1->Lines->Add(spisok->family + " " + spisok->code);

Memo1->Lines->Add(" ");

spisok=spisok->next;

}

spisok=head2;

if(n2)

while (spisok!=p2->next){

Memo2->Lines->Add(spisok->family + " " + spisok->code);

Memo2->Lines->Add(" ");

spisok=spisok->next;

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::FormCreate(TObject \*Sender)

{

Memo1->Clear();

Memo1->Lines->Add("Список 1:");

Memo1->Lines->Add("");

Memo2->Clear();

Memo2->Lines->Add("Список 2:");

Memo2->Lines->Add("");

Memo3->Clear();

Memo3->Lines->Add("Список 3:");

Memo3->Lines->Add("");

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button2Click(TObject \*Sender)

{

p1\_d=head1;

if(p1!=p1\_d) {

for(int i=1;i<n1-1;i++)

p1\_d=p1\_d->next;

delete p1;

p1=p1\_d;

p1\_d->next=NULL;

}

else {

head1=NULL;

delete p1\_d;

}

n1--;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button5Click(TObject \*Sender)

{

Form2->Show();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button1Click(TObject \*Sender)

{

Memo1->Clear();

Memo2->Clear();

Memo3->Clear();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button6Click(TObject \*Sender)

{

p2\_d=head2;

if(p2!=p2\_d) {

for(int i=1;i<n2-1;i++)

p2\_d=p2\_d->next;

delete p2;

p2=p2\_d;

p2\_d->next=NULL;

}

else {

head2=NULL;

delete p2\_d;

}

n2--;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button7Click(TObject \*Sender)

{

FIFO \*spisok1=head1;

FIFO \*spisok3;

for(int i=0;i<n1;i++)

{

FIFO \*spisok2=head2;

for(int j=0;j<n2;j++)

{

if(spisok1->family == spisok2->family && spisok1->code == spisok2->code)

{

spisok3 = new FIFO;

spisok3->family=spisok1->family;

spisok3->code=spisok1->code;

if(head3){

p3->next=spisok3;

p3=spisok3;

}

else

p3=head3=spisok3;

n3++;

}

spisok2=spisok2->next;

}

spisok1=spisok1->next;

}

spisok3=head3;

if(n3)

while (spisok3!=p3->next){

Memo3->Lines->Add(spisok3->family + " " + spisok3->code);

Memo3->Lines->Add(" ");

spisok3=spisok3->next;

}

}

//---------------------------------------------------------------------------

Unit2.cpp:

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit2.h"

#include "Unit1.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm2 \*Form2;

int n1(0),n2(0);

//---------------------------------------------------------------------------

\_\_fastcall TForm2::TForm2(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::Button2Click(TObject \*Sender)

{

Form2->Hide();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::Button1Click(TObject \*Sender)

{

n1++;

FIFO \*spisok = new FIFO;

spisok->family=Edit1->Text;

spisok->code=Edit2->Text;

if(head1){

p1->next=spisok;

p1=spisok;

}

else

p1=head1=spisok;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::Button3Click(TObject \*Sender)

{

n2++;

FIFO \*spisok = new FIFO;

spisok->family=Edit3->Text;

spisok->code=Edit4->Text;

if(head2){

p2->next=spisok;

p2=spisok;

}

else

p2=head2=spisok;

}

//---------------------------------------------------------------------------

Unit1.h:

//---------------------------------------------------------------------------

#ifndef Unit1H

#define Unit1H

//---------------------------------------------------------------------------

#include <Classes.hpp>

#include <Controls.hpp>

#include <StdCtrls.hpp>

#include <Forms.hpp>

#include "Unit2.h"

extern int n1,n2;

struct FIFO

{

AnsiString family;

AnsiString code;

FIFO \*next;

};

extern FIFO \*head1,\*head2,\*head3,\*p1,\*p2,\*p3,\*p1\_d,\*p2\_d;

//---------------------------------------------------------------------------

class TForm1 : public TForm

{

\_\_published: // IDE-managed Components

TButton \*Button2;

TButton \*Button3;

TButton \*Button4;

TMemo \*Memo1;

TButton \*Button5;

TButton \*Button1;

TMemo \*Memo2;

TMemo \*Memo3;

TButton \*Button6;

TButton \*Button7;

void \_\_fastcall Button3Click(TObject \*Sender);

void \_\_fastcall Button4Click(TObject \*Sender);

void \_\_fastcall FormCreate(TObject \*Sender);

void \_\_fastcall Button2Click(TObject \*Sender);

void \_\_fastcall Button5Click(TObject \*Sender);

void \_\_fastcall Button1Click(TObject \*Sender);

void \_\_fastcall Button6Click(TObject \*Sender);

void \_\_fastcall Button7Click(TObject \*Sender);

private: // User declarations

public: // User declarations

\_\_fastcall TForm1(TComponent\* Owner);

};

//---------------------------------------------------------------------------

extern PACKAGE TForm1 \*Form1;

//---------------------------------------------------------------------------

#endif

