МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего

профессионального образования

«Юго-Западный государственный университет»  
Кафедра программной инженерии

Лабораторная работа №3  
по дисциплине «Алгоритмы и структуры данных»

Программирование с использование графов

Выполнил: Студент группы ПО-41б

Алябьев Сергей

Проверил: Доцент

Белова Т.М.

Курск, 2015

Задание :

Написать функции создания, удаления и вывода графа.

Переменные:

struct unit – структура узла графа;

int n – число сотрудников;

unit \*head,\*p,\*tem\_unit,\*del\_unit,\*del\_unit2, \*del\_all\_unit,\*out\_unit, - указатели на структуру unit;

int n – число узлов в графе;

AnsiString unit\_str – переменная для ввода с какими узлами связан создаваемый узел;

bool check – переменная для проверки наличия удаляемого узла в массиве;

int \*mas\_new – динамический массив;

int i,j – счетчики;

Алгоритм

Функция ввода:

!CSpinEdit1->Value && Edit1->Text != ""

Нет

Да

ShowMessage("Пустой узел не может иметь связей.");

return;

Нет

Да

CSpinEdit1->Value && Edit1->Text == ""

ShowMessage("Введите с какими узлами связан."); return;

unit \*temp\_unit = new unit; temp\_unit->amount = CSpinEdit1->Value;

temp\_unit->mas = new int[temp\_unit->amount];

AnsiString unit\_str = Edit1->Text;

Нет

Да

!unit\_str.Pos(";") && CSpinEdit1->Value

ShowMessage("Узлы необходимо вводить через ';'");

return;

1

1

int i=0;i<temp\_unit->amount;i++

Нет

Да

!unit\_str.Pos(";")

return;

temp\_unit->mas[i]=StrToInt(unit\_str.SubString(1,unit\_str.Pos(";")-1));

unit\_str.Delete(1,unit\_str.Pos(";"));

temp\_unit->num = n++;

p=head=temp\_unit;

Нет

Да

!head

p->next=temp\_unit;

p=temp\_unit;

Функция вывода графа:

Да

Нет

n

StringGrid1->ColCount=n+1; StringGrid1->RowCount=n+1;

4

int i=0;i<n;i++

StringGrid1->Cells[0][i+1]=i; StringGrid1->Cells[i+1][0]=i;

unit \*out\_unit = head;

;out\_unit!=p->next;out\_unit=out\_unit->next

int i=0;i<out\_unit->amount;i++

StringGrid1->Cells[out\_unit->num+1][out\_unit->mas[i]+1]="+"; StringGrid1->Cells[out\_unit->mas[i]+1][out\_unit->num+1]="+";

2

3

1

3

2

1

int j=0;j<n+1;j++

int i=0;i<n+1;i++

Да

Нет

StringGrid1->Cells[j][i]==""

StringGrid1->Cells[j][i]="-";

4

j=0;j<Memo1->Lines->Count-1-i;j++

i=0;i<Memo1->Lines->Count-1;i++

StringGrid1->ColCount=0; StringGrid1->RowCount=0;

StringGrid1->Cells[j][i]=""

StringGrid1->Cells[j][i]=""

Функция удаления узла:

unit \*del\_unit;

Да

Нет

CSpinEdit2->Value

unit \*del\_unit2=head;

del\_unit=head;

head=del\_unit->next;

delete del\_unit->mas;

del\_unit;

del\_unit2=del\_unit2->next;

int i=0;i<CSpinEdit2->Value-1;i++

del\_unit=del\_unit2->next;

del\_unit2->next=del\_unit->next; delete del\_unit->mas; delete del\_unit;

n--;

del\_unit = head;

1

1

del\_unit!=p->next

Да

bool check(false);

int i=0;i<del\_unit->amount;i++

Нет

Да

del\_unit->mas[i] == CSpinEdit2->Value

check = true; break;

5

Да

check

Да

Нет

del\_unit->amount

delete del\_unit->mas;

3

int \*mas\_new = new int[del\_unit->amount-1];

4

2

4

3

2

del\_unit->mas[i] != CSpinEdit2->Value

int i=0;i<del\_unit->amount-1;i++

Да

delete del\_unit->mas; del\_unit->mas = mas\_new;

Нет

int j=i;j<del\_unit->amount-1;j++

del\_unit->mas[j]=del\_unit->mas[j+1];

mas\_new[i] = del\_unit->mas[i];

i--;

del\_unit->amount--;

Да

del\_unit->num > CSpinEdit2->Value

del\_unit->num--;

4

5

4

int i=0;i<del\_unit->amount;i++

del\_unit->mas[i] > CSpinEdit2->Value

Да

del\_unit->mas[i]--;

del\_unit=del\_unit->next;

Текст программы

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

#include <vcl.h>

#pragma hdrstop

#include "Unit1.h"

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

#pragma package(smart\_init)

#pragma link "CSPIN"

#pragma resource "\*.dfm"

TForm1 \*Form1;

int n(0);

struct unit

{

int amount,num,\*mas;

unit \*next;

};

unit \*head,\*p;

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

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

: TForm(Owner)

{

}

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

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

{

//----Проверка введенных данных------------------------

if(!CSpinEdit1->Value && Edit1->Text != "")

{

ShowMessage("Пустой узел не может иметь связей.");

return;

}

if(CSpinEdit1->Value && Edit1->Text == "")

{

ShowMessage("Введите с какими узлами связан.");

return;

}

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

//----Заполнение узла----------------------------------------------------------

unit \*temp\_unit = new unit;

temp\_unit->amount = CSpinEdit1->Value;

temp\_unit->mas = new int[temp\_unit->amount];

AnsiString unit\_str = Edit1->Text;

if(!unit\_str.Pos(";") && CSpinEdit1->Value)

{

ShowMessage("Узлы необходимо вводить через ';'");

return;

}

for(int i=0;i<temp\_unit->amount;i++)

{

if(!unit\_str.Pos(";"))

return;

temp\_unit->mas[i]=StrToInt(unit\_str.SubString(1,unit\_str.Pos(";")-1));

unit\_str.Delete(1,unit\_str.Pos(";"));

}

temp\_unit->num = n++;

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

//----Добавление узла в граф--

if(!head)

p=head=temp\_unit;

else

{

p->next=temp\_unit;

p=temp\_unit;

}

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

}

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

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

{

Close();

}

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

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

{

for(int i=0;i<n+1;i++)

for(int j=0;j<n+1;j++)

StringGrid1->Cells[j][i]="";

StringGrid1->ColCount=0;

StringGrid1->RowCount=0;

CSpinEdit1->Value=0;

Edit1->Text="";

}

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

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

{

if(n)

{

StringGrid1->ColCount=n+1;

StringGrid1->RowCount=n+1;

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

{

StringGrid1->Cells[0][i+1]=i;

StringGrid1->Cells[i+1][0]=i;

}

unit \*out\_unit = head;

for(;out\_unit!=p->next;out\_unit=out\_unit->next)

for(int i=0;i<out\_unit->amount;i++)

{

StringGrid1->Cells[out\_unit->num+1][out\_unit->mas[i]+1]="+";

StringGrid1->Cells[out\_unit->mas[i]+1][out\_unit->num+1]="+";

}

for(int i=0;i<n+1;i++)

for(int j=0;j<n+1;j++)

if(StringGrid1->Cells[j][i]=="")

StringGrid1->Cells[j][i]="-";

for(int i=0;i<n+1;i++)

StringGrid1->Cells[i][i]="";

}

else

{

for(int i=0;i<n+1;i++)

for(int j=0;j<n+1;j++)

StringGrid1->Cells[j][i]="";

StringGrid1->ColCount=0;

StringGrid1->RowCount=0;

}

}

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

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

{

if(n)

{

unit \*del\_unit;

//----Удаление узла из списка-------------------

if(CSpinEdit2->Value)

{

unit \*del\_unit2=head;

for(int i=0;i<CSpinEdit2->Value-1;i++)

del\_unit2=del\_unit2->next;

del\_unit=del\_unit2->next;

del\_unit2->next=del\_unit->next;

delete del\_unit->mas;

delete del\_unit;

}

else

{

del\_unit=head;

head=del\_unit->next;

delete del\_unit->mas;

delete del\_unit;

}

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

n--;

del\_unit = head;

while(del\_unit!=p->next)

{

bool check(false);

for(int i=0;i<del\_unit->amount;i++)

if(del\_unit->mas[i] == CSpinEdit2->Value)

{

check = true;

break;

}

if(check)

{

if(del\_unit->amount)

{

int \*mas\_new = new int[del\_unit->amount-1];

for(int i=0;i<del\_unit->amount-1;i++)

if(del\_unit->mas[i] != CSpinEdit2->Value)

mas\_new[i] = del\_unit->mas[i];

else

{

for(int j=i;j<del\_unit->amount-1;j++)

del\_unit->mas[j]=del\_unit->mas[j+1];

i--;

}

delete del\_unit->mas;

del\_unit->mas = mas\_new;

}

else

delete del\_unit->mas;

del\_unit->amount--;

}

for(int i=0;i<del\_unit->amount;i++)

if(del\_unit->mas[i] > CSpinEdit2->Value)

del\_unit->mas[i]--;

if(del\_unit->num > CSpinEdit2->Value)

del\_unit->num--;

del\_unit=del\_unit->next;

}

}

}

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

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

{

if(n)

{

unit \*del\_all\_unit = head;

while(del\_all\_unit!=p->next)

{

delete del\_all\_unit->mas;

head=del\_all\_unit->next;

delete del\_all\_unit;

del\_all\_unit=head;

n--;

}

}

}

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