Лабораторна робота

Тема: Динамічні структури даних.

Студента групи 141

Бужака Андрія

Варіант №4

Завдання: Описати функцію, яка підраховує кількість входжень елемента Е, значення якого введено із клавіатури, у списку Q.

Код програми:

**Файл UMain.cpp**

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

#include <vcl.h>

#pragma hdrstop

#include "UMain.h"

#include "UAvto.h"

#include "URandomContent.h"

#include "USearch.h"

#pragma package(smart\_init)

#pragma resource "\*.dfm"

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

TfrmMain \*frmMain;

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

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

: TForm(Owner)

{

}

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

void FEnabled1(bool f)

{

frmMain->edtNumber->Enabled=f;

frmMain->edtMark->Enabled=f;

frmMain->edtNameLast->Enabled=f;

frmMain->edtNameFirst->Enabled=f;

frmMain->edtAddressStreet->Enabled=f;

frmMain->edtAddressHouse->Enabled=f;

frmMain->edtAddressFlat->Enabled=f;

frmMain->edtReceipt->Enabled=f;

frmMain->btnFirst->Enabled=!f;

frmMain->btnPrev->Enabled=!f;

frmMain->btnNext->Enabled=!f;

frmMain->btnLast->Enabled=!f;

frmMain->btnNew->Enabled=!f;

frmMain->btnUpdate->Enabled=!f;

frmMain->btnDelete->Enabled=!f;

frmMain->btnSave->Enabled=f;

frmMain->btnRandomContent->Enabled=f;

}

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

void FShow(TAvto a)

{

frmMain->edtNumber->Text=a.number;

frmMain->edtMark->Text=a.mark;

frmMain->edtNameLast->Text=a.name.last;

frmMain->edtNameFirst->Text=a.name.first;

frmMain->edtAddressStreet->Text=a.address.street;

frmMain->edtAddressHouse->Text=a.address.house;

frmMain->edtAddressFlat->Text=a.address.flat;

frmMain->edtReceipt->Text=a.receipt;

}

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

void \_\_fastcall TfrmMain::btnSearchClick(TObject \*Sender)

{

frmSearch->Show();

}

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

void \_\_fastcall TfrmMain::btnFirstClick(TObject \*Sender)

{

FFirst();

FShow(FGet());

}

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

void \_\_fastcall TfrmMain::btnPrevClick(TObject \*Sender)

{

FPrev();

FShow(FGet());

}

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

void \_\_fastcall TfrmMain::btnNextClick(TObject \*Sender)

{

FNext();

FShow(FGet());

}

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

void \_\_fastcall TfrmMain::btnLastClick(TObject \*Sender)

{

FLast();

FShow(FGet());

}

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

void \_\_fastcall TfrmMain::btnNewClick(TObject \*Sender)

{

FNew();

FShow(FGet());

FEnabled1(true);

}

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

void \_\_fastcall TfrmMain::btnSaveClick(TObject \*Sender)

{

TAvto b;

b.number=frmMain->edtNumber->Text;

b.mark=frmMain->edtMark->Text;

b.name.last=frmMain->edtNameLast->Text;

b.name.first=frmMain->edtNameFirst->Text;

b.address.street=frmMain->edtAddressStreet->Text;

b.address.house=frmMain->edtAddressHouse->Text;

b.address.flat=frmMain->edtAddressFlat->Text;

b.receipt=frmMain->edtReceipt->Text;

FSave(b);

FEnabled1(false);

}

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

void \_\_fastcall TfrmMain::btnUpdateClick(TObject \*Sender)

{

FEnabled1(true);

}

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

void \_\_fastcall TfrmMain::btnDeleteClick(TObject \*Sender)

{

if(Application->MessageBox("Ви дійсно бажаєте видалити цей запис?","Виделання поточного запису",MB\_YESNO)==mrYes)

{

FDelete();

FShow(FGet());

}

}

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

void \_\_fastcall TfrmMain::btnRandomContentClick(TObject \*Sender)

{

char s[1000];

FRandom1(s);

frmMain->edtNumber->Text=s;

FRandom2(s);

frmMain->edtMark->Text=s;

FRandom3(s);

frmMain->edtNameLast->Text=s;

FRandom4(s);

frmMain->edtNameFirst->Text=s;

FRandom5(s);

frmMain->edtAddressStreet->Text=s;

FRandom6(s);

frmMain->edtAddressHouse->Text=s;

FRandom7(s);

frmMain->edtAddressFlat->Text=s;

FRandom8(s);

frmMain->edtReceipt->Text=s;

}

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

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

{

FNew();

FEnabled1(true);

}

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

**Файл UMain.h**

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

#ifndef UMainH

#define UMainH

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

#include <Classes.hpp>

#include <Controls.hpp>

#include <StdCtrls.hpp>

#include <Forms.hpp>

#include "UAvto.h"

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

class TfrmMain : public TForm

{

\_\_published: // IDE-managed Components

TEdit \*edtNumber;

TEdit \*edtMark;

TEdit \*edtNameLast;

TEdit \*edtNameFirst;

TEdit \*edtAddressStreet;

TEdit \*edtAddressHouse;

TEdit \*edtAddressFlat;

TEdit \*edtReceipt;

TButton \*btnSearch;

TButton \*btnFirst;

TButton \*btnPrev;

TButton \*btnNext;

TButton \*btnLast;

TButton \*btnNew;

TButton \*btnSave;

TButton \*btnUpdate;

TButton \*btnDelete;

TLabel \*Label1;

TLabel \*Label2;

TLabel \*Label3;

TLabel \*Label4;

TLabel \*Label5;

TLabel \*Label6;

TLabel \*Label7;

TLabel \*Label8;

TLabel \*Label9;

TButton \*btnRandomContent;

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

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

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

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

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

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

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

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

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

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

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

private: // User declarations

public: // User declarations

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

};

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

extern PACKAGE TfrmMain \*frmMain;

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

void FShow(TAvto);

#endif

**Файл UAvto.cpp**

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

#pragma hdrstop

#include "UAvto.h"

#include "UMain.h"

#pragma package(smart\_init)

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

TAvto \*Avto=NULL;

int N=1;

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

void FFirst()

{

if(Avto!=NULL)

while(Avto->prev!=NULL)

Avto=Avto->prev;

}

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

void FPrev()

{

if(Avto!=NULL && Avto->prev!=NULL)

Avto=Avto->prev;

}

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

void FNext()

{

if(Avto!=NULL && Avto->next!=NULL)

Avto=Avto->next;

}

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

void FLast()

{

if(Avto!=NULL)

while(Avto->next!=NULL)

Avto=Avto->next;

}

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

void FNew()

{

TAvto \*tmp;

if(Avto!=NULL)

{

FLast();

tmp=new TAvto;

tmp->prev=Avto;

Avto->next=tmp;

tmp->next=NULL;

tmp->id=N++;

Avto=tmp;

}

else

{

Avto=new TAvto;

Avto->prev=NULL;

Avto->next=NULL;

Avto->id=N++;

}

}

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

void FSave(TAvto a)

{

if(Avto!=NULL)

{

Avto->number=a.number;

Avto->mark=a.mark;

Avto->name.last=a.name.last;

Avto->name.first=a.name.first;

Avto->address.street=a.address.street;

Avto->address.house=a.address.house;

Avto->address.flat=a.address.flat;

Avto->receipt=a.receipt;

}

}

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

void FDelete()

{

TAvto \*d;

if(Avto!=NULL)

{

if(Avto->prev!=NULL && Avto->next!=NULL)

{

Avto->prev->next=Avto->next;

Avto->next->prev=Avto->prev;

d=Avto;

Avto=Avto->next;

delete d;

}

else

if(Avto->prev!=NULL && Avto->next==NULL)

{

d=Avto;

Avto->prev->next=NULL;

Avto=Avto->prev;

delete d;

}

else

if(Avto->prev==NULL && Avto->next!=NULL)

{

d=Avto;

Avto->next->prev=NULL;

Avto=Avto->next;

delete d;

}

else

if(Avto->prev==NULL && Avto->next==NULL)

{

delete Avto;

Avto=NULL;

}

}

}

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

TAvto FGet()

{

TAvto a;

if(Avto!=NULL)

{

a.id=Avto->id;

a.number=Avto->number;

a.mark=Avto->mark;

a.name.last=Avto->name.last;

a.name.first=Avto->name.first;

a.address.street=Avto->address.street;

a.address.house=Avto->address.house;

a.address.flat=Avto->address.flat;

a.receipt=Avto->receipt;

}

else

{

a.id=NULL;

a.number="";

a.mark="";

a.name.last="";

a.name.first="";

a.address.street="";

a.address.house="";

a.address.flat="";

a.receipt="";

}

return a;

}

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

bool FIsEnd()

{

if(Avto->next==NULL)

return true;

else

return false;

}

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

void FFindID(int i)

{

if(Avto!=NULL)

{

FFirst();

while(Avto->id!=i && Avto->next!=NULL)

FNext();

FShow(FGet());

}

}

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

**Файл UAvto.h**

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

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

#ifndef UAvtoH

#define UAvtoH

#include <String.h>

#include <Classes.hpp>

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

typedef struct

{

String last;

String first;

}TName;

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

typedef struct

{

String street;

String house;

String flat;

}TAddress;

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

typedef struct OAvto

{

int id;

String number;

String mark;

TName name;

TAddress address;

String receipt;

OAvto \*prev;

OAvto \*next;

}TAvto;

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

void FFirst();

void FPrev();

void FNext();

void FLast();

void FNew();

void FSave(TAvto);

void FDelete();

TAvto FGet();

bool FIsEnd();

void FFindID(int);

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

#endif

**Файл USearch.cpp**

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

#include <vcl.h>

#pragma hdrstop

#include "USearch.h"

#include "UAvto.h"

#include "UMain.h"

#pragma package(smart\_init)

#pragma resource "\*.dfm"

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

TfrmSearch \*frmSearch;

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

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

: TForm(Owner)

{

}

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

int FFind(TAvto a,String s) //повертає кількість входжень елемента в запис

{

int k=0;

if(a.number.Pos(s)>0)

k++;

if(a.mark.Pos(s)>0)

k++;

if(a.name.last.Pos(s)>0)

k++;

if(a.name.first.Pos(s)>0)

k++;

if(a.address.street.Pos(s)>0)

k++;

if(a.address.house.Pos(s)>0)

k++;

if(a.address.flat.Pos(s)>0)

k++;

if(a.receipt.Pos(s)>0)

k++;

return k;

}

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

int FSearch(String s) //повертає кількість входжень шуканого елемента у всі поля всіх записів

{

int k=0,t=1;

TAvto a;

FFirst();

while(FIsEnd()==false)

{

a=FGet();

k+=FFind(a,s);

if(FFind(a,s)>0)

{

frmSearch->stg->Cells[0][t]=IntToStr(a.id);

frmSearch->stg->Cells[1][t]=a.number;

frmSearch->stg->Cells[2][t]=a.mark;

frmSearch->stg->Cells[3][t]=a.name.last;

frmSearch->stg->Cells[4][t]=a.name.first;

frmSearch->stg->Cells[5][t]=a.address.street;

frmSearch->stg->Cells[6][t]=a.address.house;

frmSearch->stg->Cells[7][t]=a.address.flat;

frmSearch->stg->Cells[8][t]=a.receipt;

t++;

}

FNext();

}

a=FGet();

k+=FFind(a,s);

if(FFind(a,s)>0)

{

frmSearch->stg->Cells[0][t]=IntToStr(a.id);

frmSearch->stg->Cells[1][t]=a.number;

frmSearch->stg->Cells[2][t]=a.mark;

frmSearch->stg->Cells[3][t]=a.name.last;

frmSearch->stg->Cells[4][t]=a.name.first;

frmSearch->stg->Cells[5][t]=a.address.street;

frmSearch->stg->Cells[6][t]=a.address.house;

frmSearch->stg->Cells[7][t]=a.address.flat;

frmSearch->stg->Cells[8][t]=a.receipt;

t++;

}

frmSearch->stg->RowCount=t;

return k;

}

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

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

{

frmSearch->stg->Cells[0][0]="ID";

frmSearch->stg->Cells[1][0]="Номер авто";

frmSearch->stg->Cells[2][0]="Марка";

frmSearch->stg->Cells[3][0]="Прізвище";

frmSearch->stg->Cells[4][0]="Ім'я";

frmSearch->stg->Cells[5][0]="Вулиця";

frmSearch->stg->Cells[6][0]="Будинок";

frmSearch->stg->Cells[7][0]="Квартира";

frmSearch->stg->Cells[8][0]="Штрафи";

}

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

void \_\_fastcall TfrmSearch::btnShawAllClick(TObject \*Sender)

{

TAvto a;

int t=1;

FFirst();

while(FIsEnd()==false)

{

a=FGet();

frmSearch->stg->Cells[0][t]=IntToStr(a.id);

frmSearch->stg->Cells[1][t]=a.number;

frmSearch->stg->Cells[2][t]=a.mark;

frmSearch->stg->Cells[3][t]=a.name.last;

frmSearch->stg->Cells[4][t]=a.name.first;

frmSearch->stg->Cells[5][t]=a.address.street;

frmSearch->stg->Cells[6][t]=a.address.house;

frmSearch->stg->Cells[7][t]=a.address.flat;

frmSearch->stg->Cells[8][t]=a.receipt;

t++;

FNext();

}

a=FGet();

frmSearch->stg->Cells[0][t]=IntToStr(a.id);

frmSearch->stg->Cells[1][t]=a.number;

frmSearch->stg->Cells[2][t]=a.mark;

frmSearch->stg->Cells[3][t]=a.name.last;

frmSearch->stg->Cells[4][t]=a.name.first;

frmSearch->stg->Cells[5][t]=a.address.street;

frmSearch->stg->Cells[6][t]=a.address.house;

frmSearch->stg->Cells[7][t]=a.address.flat;

frmSearch->stg->Cells[8][t]=a.receipt;

t++;

frmSearch->stg->RowCount=t;

}

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

void \_\_fastcall TfrmSearch::btnSearchClick(TObject \*Sender)

{

String s;

s=frmSearch->edtSearch->Text;

frmSearch->edtCount->Text=IntToStr(FSearch(s));

}

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

void \_\_fastcall TfrmSearch::stgSelectCell(TObject \*Sender, int ACol,

int ARow, bool &CanSelect)

{

FFindID(StrToInt(frmSearch->stg->Cells[0][ARow]));

}

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

**Файл URandomContent.h**

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

#ifndef URandomContentH

#define URandomContentH

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

void FRandom1(char content[]);

void FRandom2(char content[]);

void FRandom3(char content[]);

void FRandom4(char content[]);

void FRandom5(char content[]);

void FRandom6(char content[]);

void FRandom7(char content[]);

void FRandom8(char content[]);

#endif





