МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего

профессионального образования

«Юго-Западный государственный университет»  
Кафедра программной инженерии

Лабораторная работа №4  
по дисциплине «Программирование на ЯВУ»

Изучение структур данных

Выполнил: Студент группы ПО-41б

Алябьев Сергей

Проверил: Доцент

Белова Т.М.

Курск, 2015

Задание 1:

Имеются данные о сотрудниках фирмы (фамилия, возраст, отношение к воинской службе). Найти фамилию сотрудника, младшего по возрасту и невоеннообязанного. Если таких сотрудником несколько, то вывести всех, упорядочив список фамилий по возрастанию.

Переменные:

struct data – структура для данных о сотруднике фирмы;

int n – число сотрудников;

\*head,\*p - указатели на структуру data;

int min\_age – переменная для хранения минимального возраста;

bool test\_army – проверка наличия хотя бы 1-го невоеннообязанного;

int i,j – счетчики;

Алгоритм

temp = head;

test\_army = false;

Нет

Да

n

temp!=p->next

ShowMessage("Список пуст")

Да

!temp->army

Да

test\_army=true; min\_age=temp->age; break;

temp=temp->next;

Да

!test\_army

ShowMessage("Все сотрудники военнообязанные"); return;

1

2

1

2

temp=head;

temp=temp->next;

min\_age = temp->age;

Да

temp!=p->next

!temp->army && temp->age < min\_age

Да

temp=temp->next;

Да

!temp->army && temp->age == min\_age

Да

temp!=p->next

temp=head;

Memo1->Lines->Add(temp->family)

3

4

3

4

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

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

Memo1->Lines->Strings[j] > Memo1->Lines->Strings[j+1]

Да

Нет

AnsiString temp\_str = Memo1->Lines->Strings[j]; Memo1->Lines->Strings[j] = Memo1->Lines->Strings[j+1]; Memo1->Lines->Strings[j+1] = temp\_str;

Нет

Да

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

Memo1->Lines->Strings[i] == Memo1->Lines->Strings[i+1]

Memo1->Lines->Delete(i); i-=2;

Текст программы

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

#include <vcl.h>

#pragma hdrstop

#include "Unit1.h"

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

#pragma package(smart\_init)

#pragma link "CSPIN"

#pragma resource "\*.dfm"

TForm1 \*Form1;

struct data

{

AnsiString family;

int age;

bool army;

data \*next;

};

int n(0);

data \*head,\*p;

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

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

: TForm(Owner)

{

}

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

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

{

data \*add\_collab = new data;

if(Edit1->Text!="")

add\_collab->family=Edit1->Text;

else

{

ShowMessage("Введите фамилию.");

return;

}

if(CSpinEdit1->Value)

add\_collab->age=CSpinEdit1->Value;

else

{

ShowMessage("Введите возраст");

return;

}

if(RadioGroup1->ItemIndex)

add\_collab->army=false;

else

add\_collab->army=true;

if(!head)

p=head=add\_collab;

else

{

p->next=add\_collab;

p=add\_collab;

}

n++;

}

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

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

{

Memo1->Clear();

}

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

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

{

Close();

}

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

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

{

Memo1->Clear();

}

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

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

{

data \*temp=head;

int min\_age;

bool test\_army(false);

if(n)

{

while(temp!=p->next)

{

if(!temp->army)

{

test\_army=true;

min\_age=temp->age;

break;

}

temp=temp->next;

}

if(!test\_army)

{

ShowMessage("Все сотрудники военнообязанные");

return;

}

temp=head;

while(temp!=p->next)

{

if(!temp->army && temp->age < min\_age)

min\_age = temp->age;

temp=temp->next;

}

temp=head;

while(temp!=p->next)

{

if(!temp->army && temp->age == min\_age)

Memo1->Lines->Add(temp->family);

temp=temp->next;

}

for(int i=0;i<Memo1->Lines->Count-1;i++)

for(int j=0;j<Memo1->Lines->Count-1-i;j++)

if(Memo1->Lines->Strings[j] > Memo1->Lines->Strings[j+1])

{

AnsiString temp\_str = Memo1->Lines->Strings[j];

Memo1->Lines->Strings[j] = Memo1->Lines->Strings[j+1];

Memo1->Lines->Strings[j+1] = temp\_str;

}

for(int i=0;i<Memo1->Lines->Count;i++)

if(Memo1->Lines->Strings[i] == Memo1->Lines->Strings[i+1])

{

Memo1->Lines->Delete(i);

i-=2;

}

}

else

ShowMessage("Список пуст");

}

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