|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

**ОТЧЕТ ОБ УЧЕБНОЙ ПРАКТИКЕ**

|  |  |
| --- | --- |
| Тип практики | Проектно-технологическая практика |

|  |  |
| --- | --- |
| Название  предприятия | НУК ИУ МГТУ им. Н.Э. Баумана |

|  |  |  |  |
| --- | --- | --- | --- |
| Студент группы ИУ6-21Б |  |  | Т.Е. Старжевский |
|  |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |
| Руководитель практики |  |  | О.А. Веселовская |
|  |  | (Подпись, дата) | (И.О. Фамилия) |

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*2022 г.*

**Оглавление:**

Техническое задание..................................................................................................3

Объектная декомпозиция программы……………………………………………..3

Диаграмма пользовательского интерфейса.............................................................4

Диаграмма классов интерфейсной и предметной области программы................5

Полученные формы интерфейса...............................................................................6

Диаграмма последовательности действий процедуры *TMain.Button1Click.........*7

Код программы, разделенный на модули (Unit).....................................................8

unit Main\_Form......................................................................................................................8

unit Data\_base\_form...............................................................................................................9

unit Form\_dedit.....................................................................................................................13

unit Pass\_form.......................................................................................................................20

unit graphic\_form..................................................................................................................21

Заключение................................................................................................................24

**Вариант 16.**

**Задание 1. Создание программной системы на Object Pascal.**

Выполнить объектную декомпозицию, разработать формы интерфейса, диаграмму состояний интерфейса, диаграммы классов интерфейсной и предметной областей, диаграмму последовательности действий одной из реализуемых операций. Разработать, протестировать и отладить программу.

База данных (файл) трамвайного депо содержит сведения о подвижном составе: инвентарный номер вагона, модель, год производства, срок службы (лет). Программа должна в интерактивном режиме формировать файл, добавлять и удалять данные, а также воспринимать каждый из перечисленных запросов и давать на него ответ.

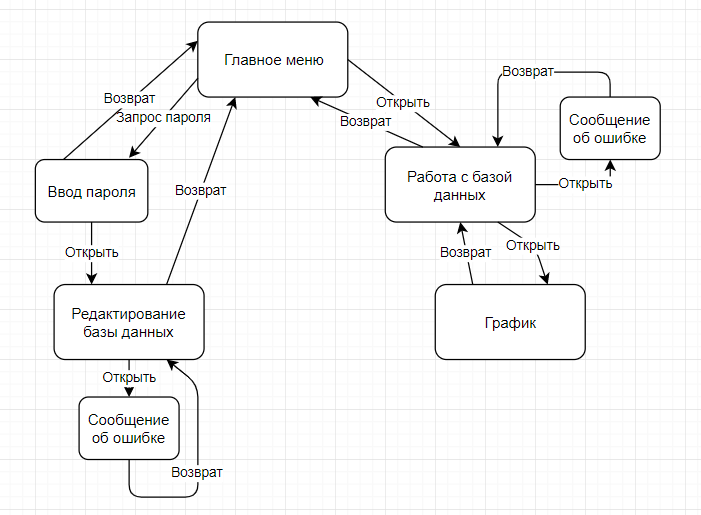
1. Найти модели трамваев, которые произведены начиная с заданного года.

2. Показать номера и модели трамваев, срок службы которых истек.

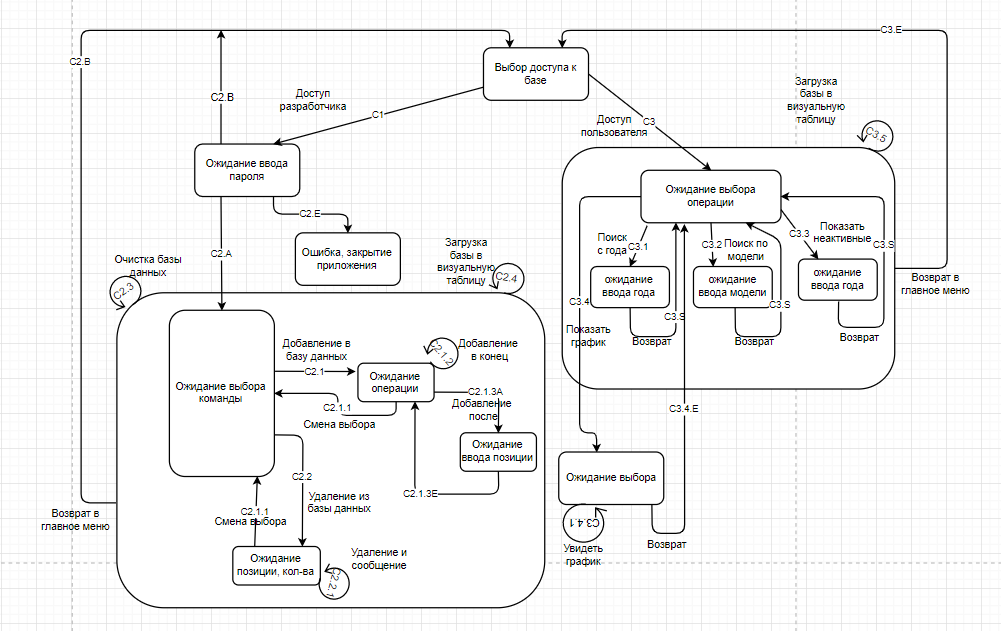
3. Определить количество трамваев заданной модели в депо.

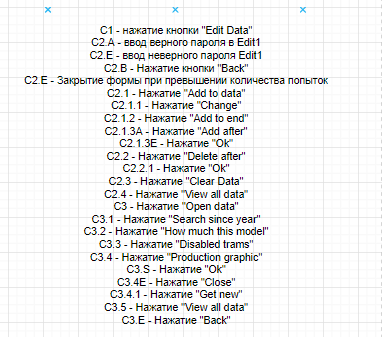
4. Построить график количества произведенных трамваев по годам.

**Объектная декомпозиция программы:**

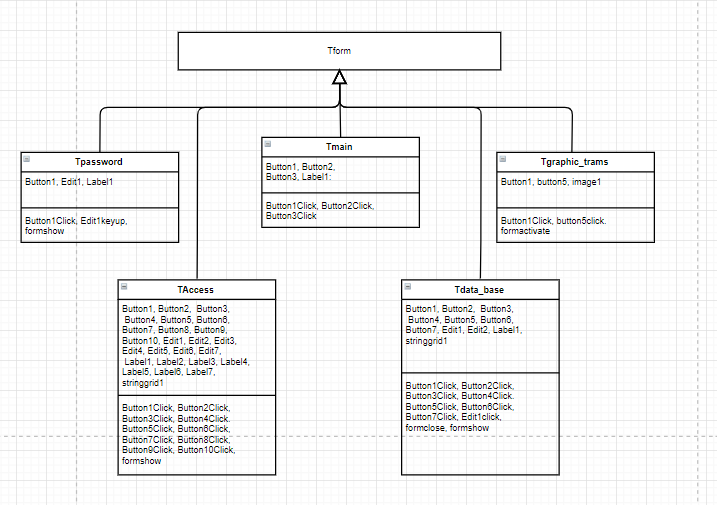
****

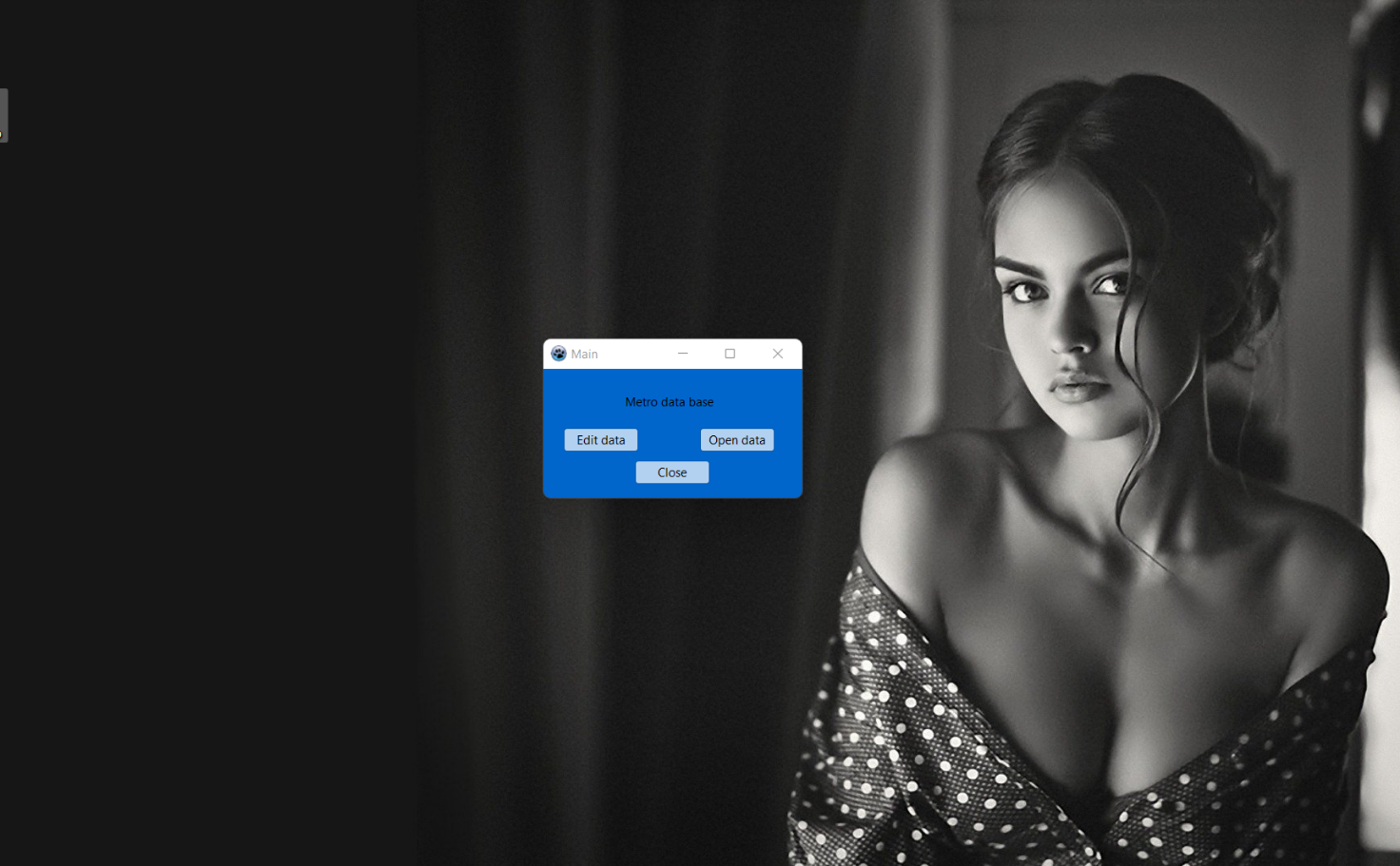
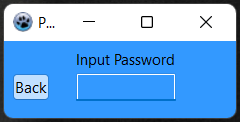
**Диаграмма пользовательского интерфейса:**

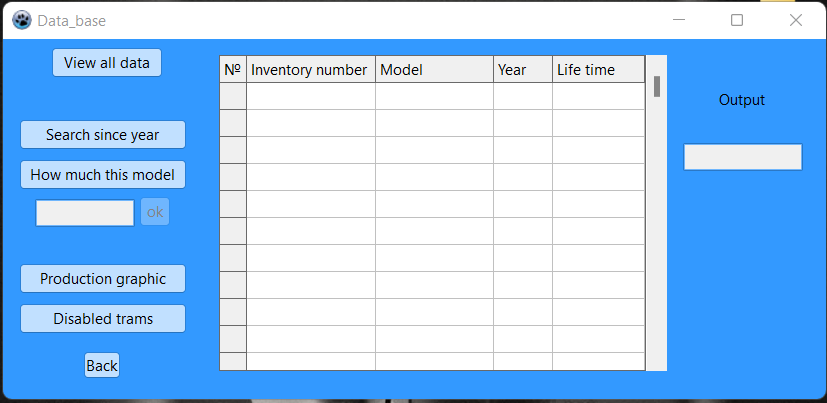
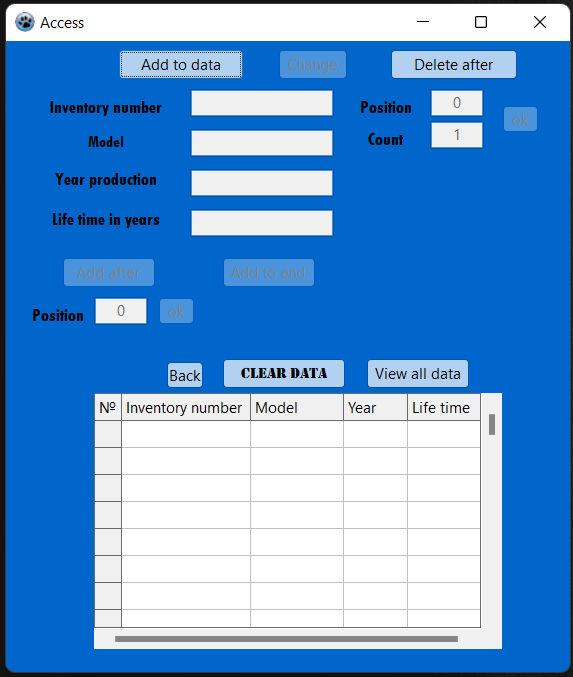
****

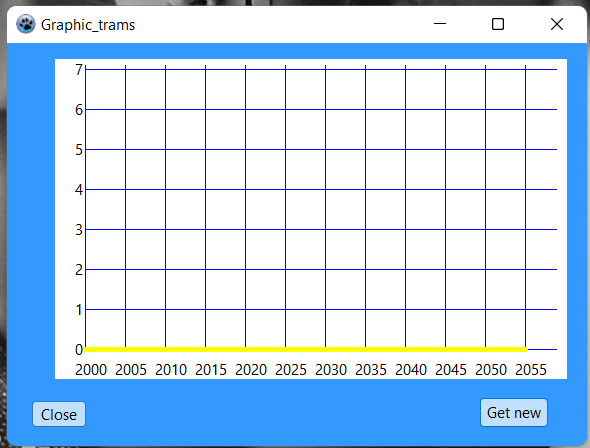
****

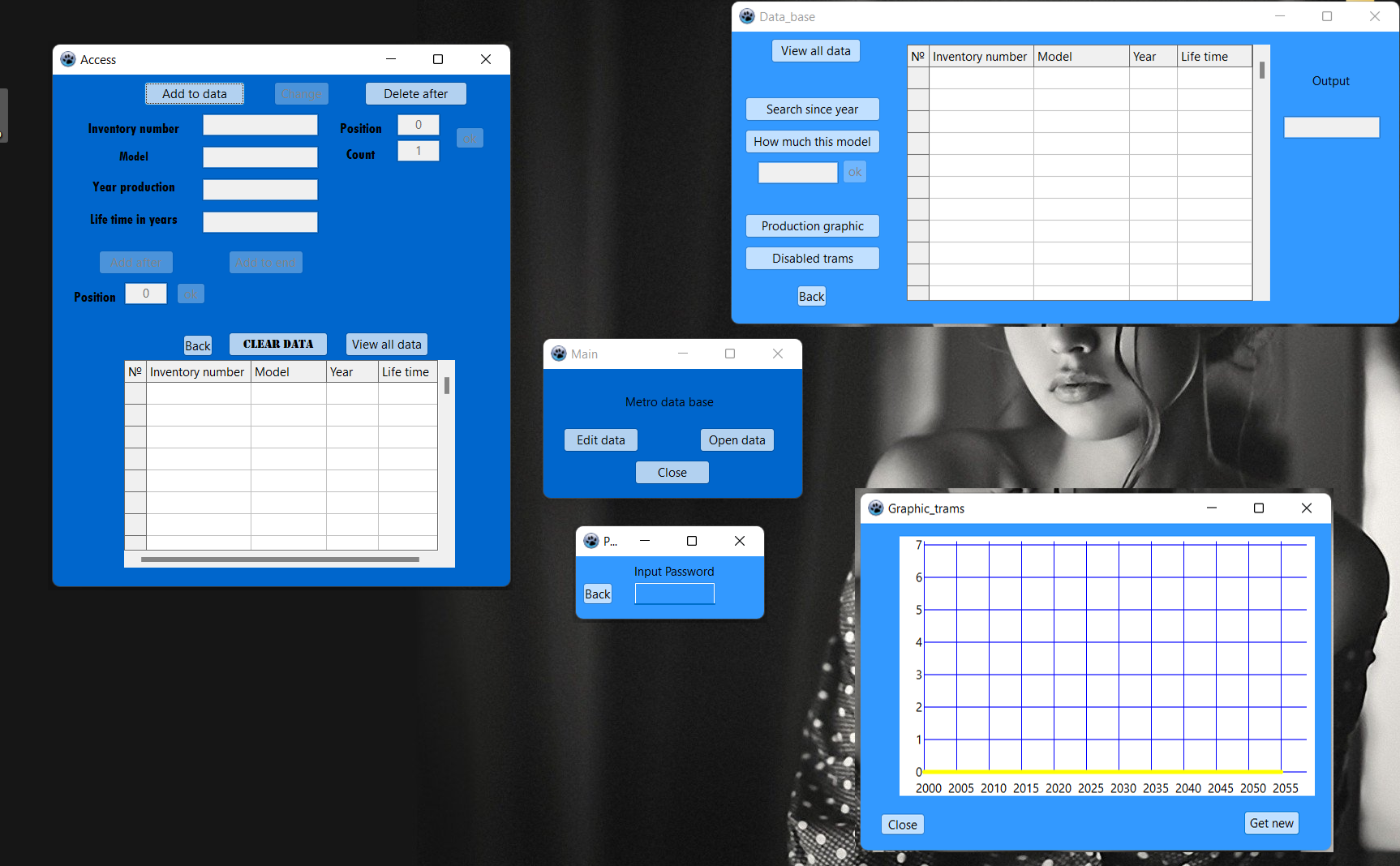
**Диаграмма классов интерфейсной и предметной области программы:**

****

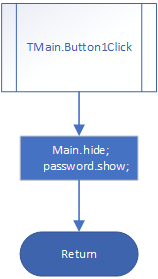
**Полученные формы интерфейса:**

****

****

В совокупности выглядит так:

**Диаграмма последовательности действий процедуры** *TMain.Button1Click:*



**Код программы, разделенный на модули (Unit):**

**unit Main\_Form;**

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, Forms, Controls, Graphics,

Dialogs, StdCtrls, pass\_form, data\_base\_form,graphic\_form;

type

metro\_member = record

Inv\_num,birthday,life\_time:word;

model:string[15];

end;

{ TMain }

TMain = class(TForm)

Button1: TButton;

Button2: TButton;

Button3: TButton;

Label1: TLabel;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

private

public

end;

var

Main: TMain;

pass:string[20] = '0957';

f:file of metro\_member;

implementation

{$R \*.lfm}

{procedure metro\_member.init (iInv\_num,ibirthday,ilife\_time:word;imodel:string);

begin

Inv\_num:=iInv\_num;

birthday := ibirthday;

life\_time := ilife\_time;

model := imodel;

end;

procedure metro\_member.print;

begin

writeln ('Inv number: ', inv\_num, ' | Model: ',model, ' | Year production: ',birthday, ' | Life time: ',life\_time);

end; }

{ TMain }

procedure TMain.Button1Click(Sender: TObject);

begin

Main.hide;

password.show;

end;

procedure TMain.Button2Click(Sender: TObject);

begin

main.hide;

data\_base.Show;

end;

procedure TMain.Button3Click(Sender: TObject);

begin

data\_base.Close;

password.Close;

main.Close;

end;

end.

**unit Data\_base\_form;**

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, Forms, Controls,

Graphics, Dialogs, StdCtrls, Grids, ComCtrls,graphic\_form;

type

{ TData\_base }

TData\_base = class(TForm)

Button1: TButton;

Button2: TButton;

Button3: TButton;

Button4: TButton;

Button5: TButton;

Button6: TButton;

Button7: TButton;

Edit1: TEdit;

Edit2: TEdit;

Label1: TLabel;

StringGrid1: TStringGrid;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button5Click(Sender: TObject);

procedure Button6Click(Sender: TObject);

procedure Button7Click(Sender: TObject);

procedure Edit1Click(Sender: TObject);

procedure FormClose(Sender: TObject; var CloseAction: TCloseAction);

procedure FormShow(Sender: TObject);

private

public

end;

var

Data\_base: TData\_base;

implementation

uses main\_form;

var kod:integer;

{$R \*.lfm}

{ TData\_base }

procedure TData\_base.Button6Click(Sender: TObject);

var rec : metro\_member;

size,i:integer;

begin

stringgrid1.Clean();

reset(f);

stringgrid1.Cells[0,0]:= '№';

stringgrid1.cells[1,0]:='Inventory number';

stringgrid1.cells[2,0]:='Model';

stringgrid1.cells[3,0]:='Year';

stringgrid1.cells[4,0]:='Life time';

size := filesize(f);

for i:= 1 to size do begin

read(f,rec);

stringgrid1.cells[0,i]:=inttostr(i);

stringgrid1.cells[1,i]:=inttostr(rec.Inv\_num);

stringgrid1.cells[2,i]:=rec.model;

stringgrid1.cells[3,i]:=inttostr(rec.birthday);

stringgrid1.cells[4,i]:=inttostr(rec.life\_time);

end;

end;

procedure TData\_base.Button7Click(Sender: TObject);

var rec:metro\_member;

i,vsp,j:integer;

s:string;

begin

reset(f);

stringgrid1.Clean(1,1,4,1000,[]);

case kod of

1:begin

val(edit1.Text,vsp,i);

j:=1;

for i:=1 to filesize(f) do begin

read(f,rec);

if rec.birthday >= vsp then begin

stringgrid1.cells[0,j]:=inttostr(j);

stringgrid1.cells[1,j]:=inttostr(rec.Inv\_num);

stringgrid1.cells[2,j]:=rec.model;

stringgrid1.cells[3,j]:=inttostr(rec.birthday);

stringgrid1.cells[4,j]:=inttostr(rec.life\_time);

j:=j+1;

end;

end;

s:=inttostr(j-1);

edit2.text:='Total: '+s;

end;

2:begin

s:=edit1.text;

j:=1;

for i:=1 to filesize(f) do begin

read(f,rec);

if rec.model = s then begin

stringgrid1.cells[0,j]:=inttostr(j);

stringgrid1.cells[1,j]:=inttostr(rec.Inv\_num);

stringgrid1.cells[2,j]:=rec.model;

stringgrid1.cells[3,j]:=inttostr(rec.birthday);

stringgrid1.cells[4,j]:=inttostr(rec.life\_time);

j:=j+ 1;

end;

end;

s:=inttostr(j-1);

edit2.text:='Total: '+s;

end;

3: begin

val(edit1.text,vsp,i);

j:=1;

for i:=1 to filesize(f) do begin

read(f,rec);

if (rec.birthday+rec.life\_time) < vsp then begin

stringgrid1.cells[0,j]:=inttostr(j);

stringgrid1.cells[1,j]:=inttostr(rec.Inv\_num);

stringgrid1.cells[2,j]:=rec.model;

stringgrid1.cells[3,j]:=inttostr(rec.birthday);

stringgrid1.cells[4,j]:=inttostr(rec.life\_time);

j:=j+ 1;

end;

end;

s:=inttostr(j-1);

edit2.text:='Total: '+s;

end;

end;

edit1.Enabled:=false;

button7.enabled:=false;

button1.Enabled:=True;

button2.Enabled:=true;

button3.Enabled:=true;

button4.Enabled:=true;

end;

procedure TData\_base.Edit1Click(Sender: TObject);

begin

edit1.Clear;

end;

procedure TData\_base.FormClose(Sender: TObject; var CloseAction: TCloseAction);

begin

graphic\_trams.Close;

end;

procedure TData\_base.FormShow(Sender: TObject);

begin

assignfile(f,'Data\_base.txt');

{$I-} reset(f); {$I+}

if Ioresult<>0 then begin

messagedlg('Data is empty ',mtconfirmation,[],0);

data\_base.Close;

main.Show;

end

else begin

edit2.Clear;

edit2.ReadOnly:=true;

edit1.Enabled:=false;

button7.enabled:=false;

button1.Enabled:=True;

button2.Enabled:=true;

button3.Enabled:=true;

button4.Enabled:=true;

end;

end;

procedure TData\_base.Button5Click(Sender: TObject);

begin

data\_base.close;

main.Show;

closefile(f);

end;

procedure TData\_base.Button1Click(Sender: TObject);

begin

edit2.Clear;

edit1.text:='Year:';

button1.Enabled:=false;

button2.Enabled:=false;

button3.Enabled:=false;

button4.Enabled:=false;

edit1.Enabled:=True;

button7.enabled:=true;

kod :=1;

edit1.NumbersOnly:=true;

end;

procedure TData\_base.Button2Click(Sender: TObject);

begin

edit2.Clear;

edit1.text:='Model:';

button1.Enabled:=false;

button2.Enabled:=false;

button3.Enabled:=false;

button4.Enabled:=false;

edit1.Enabled:=true;

button7.enabled:=true;

kod :=2;

edit1.NumbersOnly:=false;

end;

procedure TData\_base.Button3Click(Sender: TObject);

begin

button3.Enabled:=false;

graphic\_trams.Show;

end;

procedure TData\_base.Button4Click(Sender: TObject);

begin

edit2.Clear;

edit1.text:='Year now:';

button1.Enabled:=false;

button2.Enabled:=false;

button3.Enabled:=false;

button4.Enabled:=false;

edit1.Enabled:=true;

button7.enabled:=true;

kod :=3;

edit1.NumbersOnly:=false;

end;

end.

**unit Form\_dedit;**

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, Forms, Controls, Graphics,

Dialogs, StdCtrls, Grids, ComCtrls;

type

{ TAccess }

TAccess = class(TForm)

Button1: TButton;

Button10: TButton;

Button2: TButton;

Button3: TButton;

Button4: TButton;

Button5: TButton;

Button6: TButton;

Button7: TButton;

Button8: TButton;

Button9: TButton;

Edit1: TEdit;

Edit2: TEdit;

Edit3: TEdit;

Edit4: TEdit;

Edit5: TEdit;

Edit6: TEdit;

Edit7: TEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Label7: TLabel;

StringGrid1: TStringGrid;

procedure Button10Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button5Click(Sender: TObject);

procedure Button6Click(Sender: TObject);

procedure Button7Click(Sender: TObject);

procedure Button8Click(Sender: TObject);

procedure Button9Click(Sender: TObject);

procedure FormShow(Sender: TObject);

private

public

end;

var

Access: TAccess;

implementation

uses main\_form;

{$R \*.lfm}

{ TAccess }

procedure TAccess.Button4Click(Sender: TObject);

begin

closefile(f);

access.Close;

main.show;

main.SetFocus;

end;

procedure TAccess.Button5Click(Sender: TObject);

var rec:metro\_member;

code:integer=0;

s:string;

begin

val(edit1.Text,rec.Inv\_num,code);

rec.model:=edit2.Text;

val(edit3.Text,rec.birthday);

val(edit4.text,rec.life\_time);

write(f,rec);

edit1.clear;

edit2.clear;

edit3.clear;

edit4.clear;

str(filesize(f),s); edit6.text:=s; edit5.text:=s;

end;

procedure TAccess.Button6Click(Sender: TObject);

begin

button5.Enabled:=false;

button6.enabled:=false;

edit5.Enabled:=true;

button7.Enabled:=true;

end;

procedure TAccess.Button7Click(Sender: TObject);

var rec,rec\_main:metro\_member;

code,vsp,pos,size,i:integer;

Add\_to\_end:boolean = false;

s:string;

Vrem:file of metro\_member;

begin

size := filesize(f);

val (edit5.text,pos,code);

if pos > size then begin

Add\_to\_end:=Messagedlg('Position more than size of file. Add to end?', mtConfirmation,mbYesNo,0)=mrYes;

if add\_to\_end then button5.Click;

end

else begin

val(edit1.Text,rec\_main.Inv\_num,code);

rec\_main.model:=edit2.Text;

val(edit3.Text,rec\_main.birthday);

val(edit4.text,rec\_main.life\_time);

//Rewrite after

closefile(f);

rename(f,'Data\_base\_edit.txt');

reset(f);

Assignfile(Vrem,'Data\_base.txt');

rewrite(Vrem);

vsp := pos;

for i:=1 to vsp do begin

read(f,rec);

write(Vrem,rec);

end;

write(Vrem,rec\_main);

vsp := size - pos;

for i:=1 to vsp do begin

read(f,rec);

write(Vrem,rec);

end;

closefile(Vrem);

closefile(f);

erase(f);

Assignfile(f,'Data\_base.txt');

Reset(f);

seek(f,filesize(f));

str(filesize(f),s); edit6.text:=s; edit5.text:=s;

//that

button7.Enabled:=false;

button5.Enabled:=true;

button6.enabled:=true;

edit5.Enabled:=false;

str(filesize(f),s); edit6.text:=s; edit5.text:=s;

edit1.clear;

edit2.clear;

edit3.clear;

edit4.clear;

end;

end;

procedure TAccess.Button8Click(Sender: TObject);

var s:string;

count,pos,vsp,size,i:integer;

rec:metro\_member;

uncorrect:boolean=false;

Vrem:file of metro\_member;

begin

size := filesize(f);

if size <> 0 then begin

val(edit6.text,pos,vsp);

val(edit7.text,count,vsp);

//Proverki

if (count = 0) or ( pos = 0) then

messagedlg('Uncorrect position or count',mtconfirmation,[],0)

else begin

if pos>size then begin

pos := size;

uncorrect := true;

end;

while ((pos + count -1) > size) do

count := count - 1;

//

closefile(f);

rename(f,'Data\_base\_edit.txt');

reset(f);

Assignfile(Vrem,'Data\_base.txt');

rewrite(Vrem);

vsp := pos -1;

for i:=1 to vsp do begin

read(f,rec);

write(Vrem,rec);

end;

for i:=1 to count do read(f,rec);

vsp := size - pos - count +1;

for i:=1 to vsp do begin

read(f,rec);

write(Vrem,rec);

end;

closefile(Vrem);

closefile(f);

erase(f);

Assignfile(f,'Data\_base.txt');

Reset(f);

seek(f,filesize(f));

str(filesize(f),s); edit6.text:=s; edit5.text:=s;

str(count,s); edit7.text:=s;

if uncorrect then

s:='Position was > then filesize, was delete last el'

else

s := 'Successfull!';

messagedlg(s,mtconfirmation,[],0);

end;

end

else messagedlg('File was empty ',mtconfirmation,[],0)

end;

procedure TAccess.Button9Click(Sender: TObject);

var s:string;

begin

edit1.Enabled:=false;

edit2.Enabled:=false;

edit3.Enabled:=false;

edit4.Enabled:=false;

edit5.Enabled:=false;

edit6.Enabled:=false;

edit7.Enabled:=false;

Button5.Enabled:=false;

Button6.Enabled:=false;

Button7.Enabled:=false;

Button8.Enabled:=false;

Button9.Enabled:=false;

button1.enabled:=true;

button2.enabled:=true;

if IOresult=0 then begin

str(filesize(f),s); edit6.text:=s; edit5.text:=s;

end

else begin

edit6.text:='0'; edit5.text:='0';

end;

end;

procedure TAccess.FormShow(Sender: TObject);

var s:string;

begin

edit1.Enabled:=false;

edit2.Enabled:=false;

edit3.Enabled:=false;

edit4.Enabled:=false;

edit5.Enabled:=false;

edit6.Enabled:=false;

edit7.Enabled:=false;

button1.enabled:=true;

button2.enabled:=true;

Button5.Enabled:=false;

Button6.Enabled:=false;

Button7.Enabled:=false;

Button8.Enabled:=false;

Button9.Enabled:=false;

assignfile(f,'Data\_base.txt');

{$I-} reset(f); {$I+}

if Ioresult=0 then begin

seek(f,filesize(f));

str(filesize(f),s); edit6.text:=s; edit5.text:=s;

end

else begin

rewrite(f);

edit6.text:='0'; edit5.text:='0';

end;

end;

procedure TAccess.Button2Click(Sender: TObject);

begin

button1.enabled :=false;

button2.enabled:=false;

edit6.enabled:=true;

edit7.enabled:=true;

Button8.Enabled:=true;

Button9.Enabled:=true;

end;

procedure TAccess.Button3Click(Sender: TObject);

var key:boolean=false;

begin

if (filesize(f)=0)then messagedlg('File was empty ',mtconfirmation,[],0)

else begin

key:=Messagedlg('WARNING! Will be deleted ALL data-base, Are you sure?', mtConfirmation,mbYesNo,0)=mrYes;

if key then begin

rewrite(f);

edit6.text:='0'; edit5.text:='0';

end;

end;

end;

procedure TAccess.Button1Click(Sender: TObject);

begin

button1.enabled :=false;

button2.enabled:=false;

edit1.Enabled:=true;

edit2.Enabled:=true;

edit3.Enabled:=true;

edit4.Enabled:=true;

Button5.Enabled:=true;

Button6.Enabled:=true;

Button9.Enabled:=true;

end;

procedure TAccess.Button10Click(Sender: TObject);

var rec : metro\_member;

size,i:integer;

begin

stringgrid1.Clean();

reset(f);

stringgrid1.Cells[0,0]:= '№';

stringgrid1.cells[1,0]:='Inventory number';

stringgrid1.cells[2,0]:='Model';

stringgrid1.cells[3,0]:='Year';

stringgrid1.cells[4,0]:='Life time';

size := filesize(f);

for i:= 1 to size do begin

read(f,rec);

stringgrid1.cells[0,i]:=inttostr(i);

stringgrid1.cells[1,i]:=inttostr(rec.Inv\_num);

stringgrid1.cells[2,i]:=rec.model;

stringgrid1.cells[3,i]:=inttostr(rec.birthday);

stringgrid1.cells[4,i]:=inttostr(rec.life\_time);

end;

end;

end.

**unit Pass\_form;**

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls;

type

{ TPassword }

TPassword = class(TForm)

Button1: TButton;

Edit1: TEdit;

Label1: TLabel;

procedure Button1Click(Sender: TObject);

procedure Edit1KeyUp(Sender: TObject; var Key: Word; Shift: TShiftState);

procedure FormShow(Sender: TObject);

private

public

end;

var

Password: TPassword;

var tries:integer=3;

implementation

uses main\_form,form\_dedit;

{$R \*.lfm}

{ TPassword }

procedure TPassword.Edit1KeyUp(Sender: TObject; var Key: Word;

Shift: TShiftState);

var s:string;

begin

if (key = 13) then begin

if (edit1.Text = pass) then begin

password.close;

edit1.Text := '';

Access.show;

access.setfocus;

end

else begin

tries:=tries-1;

if tries < 1 then begin

messagedlg('Please, think about your behavior... Hacker', mterror, [],0);

password.Close;

end

else begin

s:='Are u have Access?! '+inttostr(tries)+' try left';

messagedlg(s, mterror, [],0);

end;

end;

end;

end;

procedure TPassword.FormShow(Sender: TObject);

begin

edit1.SetFocus;

end;

procedure TPassword.Button1Click(Sender: TObject);

begin

password.close;

access.close;

edit1.Text := '';

main.Show;

main.SetFocus;

end;

end.

**unit graphic\_form;**

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, Forms, Controls,

Graphics, Dialogs, StdCtrls, ExtCtrls;

type

{ TGraphic\_trams }

TGraphic\_trams = class(TForm)

Button1: TButton;

Button5: TButton;

Image1: TImage;

procedure Button1Click(Sender: TObject);

procedure Button5Click(Sender: TObject);

procedure FormActivate(Sender: TObject);

private

public

end;

var

Graphic\_trams: TGraphic\_trams;

implementation

uses data\_base\_form,main\_form;

{$R \*.lfm}

{ TGraphic\_trams }

procedure TGraphic\_trams.Button5Click(Sender: TObject);

begin

graphic\_trams.close;

data\_base.setfocus;

data\_base.button3.Enabled:=true;

end;

procedure TGraphic\_trams.FormActivate(Sender: TObject);

begin

image1.canvas.pen.width:=1;

image1.Canvas.Brush.Color:=clWhite;

image1.canvas.Pen.Color:=clblue;

image1.Canvas.Rectangle(-1,-1,1000,1000);

end;

procedure TGraphic\_trams.Button1Click(Sender: TObject);

var

i,j,dx,dy,size,count,help:integer;

left\_down,right\_top:tpoint;

s:string;

rec:metro\_member;

const n = 10; //n ~ Smesh text

const d = 5; // d ~ ++ about x

const first = 2000; // first ~ 1st x

begin

left\_down.Create(30,290);

right\_top.Create(502,5);

dx:=40; dy:=40;

size := filesize(f);

image1.canvas.Pen.Color:=clblue;

image1.canvas.pen.width:=1;

image1.Canvas.MoveTo(left\_down);

//Draw scheme

for i:=0 to trunc((right\_top.x-left\_down.X)/dx) do begin

s:=inttostr(i\*d+first);

image1.canvas.TextOut(left\_down.X+dx\*i-n,left\_down.y+n,s);

//horizontal

s:=inttostr(i);

image1.canvas.TextOut(left\_down.x-n,left\_down.y-dy\*i-n,s);

image1.canvas.moveto(left\_down.x,left\_down.y-dy\*i);

image1.canvas.lineto(right\_top.X,left\_down.y-dy\*i);

//vertical

image1.canvas.moveto(left\_down.x+dx\*i,left\_down.y);

image1.canvas.lineto(left\_down.X+dx\*i,right\_top.Y);

end;

image1.canvas.MoveTo(left\_down);

image1.canvas.Pen.Color:=clyellow;

image1.canvas.pen.width:=5;

//Podc4et < min

reset(f);

count := 0;

for i:=0 to size-1 do begin

read(f,rec);

if rec.birthday < first then count := count +1;

end;

image1.canvas.moveto(left\_down.x,left\_down.y-count\*dy);

// end. Pods4et norm:

help:=0;

for i:=0 to trunc((right\_top.x-left\_down.X)/dx-2) do begin

reset(f);

count :=0;

for j:=0 to size-1 do begin

read(f,rec);

if (rec.birthday < (i+1)\*d+first) and (rec.birthday >= i\*d+first) then

count := count + 1;

end;

help:=help+1;

image1.canvas.lineto(left\_down.x+(i+1)\*dx,left\_down.Y-count\*dy);

end;

//end. Pods4et > max

reset(f);

count := 0;

j:=first + d \* trunc((right\_top.x-left\_down.X)/dx);

for i:=0 to size-1 do begin

read(f,rec);

if rec.birthday > j then count := count +1;

end;

j:=trunc((right\_top.x-left\_down.X)/dx);

image1.canvas.lineto(left\_down.x+j\*dx,left\_down.y-count\*dy);

end;

end.

**Заключение:** По заданному техническому заданию было создано многооконное приложение для работы с базой данных. Объекты базы представлены в виде «записей». Программа протестирована, работает корректнj. Разработка осуществлялась в среде «Lazarus».