Оглавление

Введение…………………………………………………………………2

Глава 1……………………………………………………………………3

Анализ задания 1…………………………………………………3

Реализация задания 1. Чертежи…………………………………3

Реализация задания 1. Текст программы………………………4

Результаты тестирования приложения по заданию 1…………20

Вывод по заданию 1………………………………………………21

Глава 2……………………………………………………………………21

Анализ задания 2…………………………………………………21

Реализация задания 2. Чертежи…………………………………22

Реализация задания 2. Текст программы………………………23

Результаты тестирования программы по заданию 2………….25

Вывод по заданию 2……………………………………………..25

Глава 3……………………………………………………………………26

Анализ задания 3…………………………………………………26

Реализация задания 3. Чертежи…………………………………26

Реализация задания 3. Текст программы………………………27

Результаты тестирования приложения по заданию 3…………42

Вывод по заданию 3……………………………………………..43

Заключение……………………………………………………………..43

Список литературы……………………………………………………..43

Введение

**Целью** практикума является получение навыков создания небольших программных систем с оконными и консольными интерфейсами.

**Задачами** практикума являются:

• более глубокое изучение средств реализации проектов программ на одном из изучаемых универсальных языках программирования высокого уровня;

• овладение методикой и получение практических навыков проектирования небольших программных систем при структурном и объектном подходах;

• воспитание внимания, аккуратности, систематичности, а также формирование интереса к изучаемой профессиональной деятельности.

Выполнение практикума должно способствовать формированию и развитию навыков и умений, обеспечивающих следующие компетенции:

• выделение объектов предметной области, обобщение их в классы, определение связей между классами;

• проектирование эргономичного обеспечения информационных систем;

• разработка и отладка компонентов аппаратно-программных комплексов с помощью современных автоматизированных средств проектирования;

• разработка проектной и эксплуатационной документации на программную и техническую продукцию;

• выполнение контроля разрабатываемых проектов и технической документации на соответствие стандартам и техническим требованиям;

• разработка интерфейсов «человек - ЭВМ».

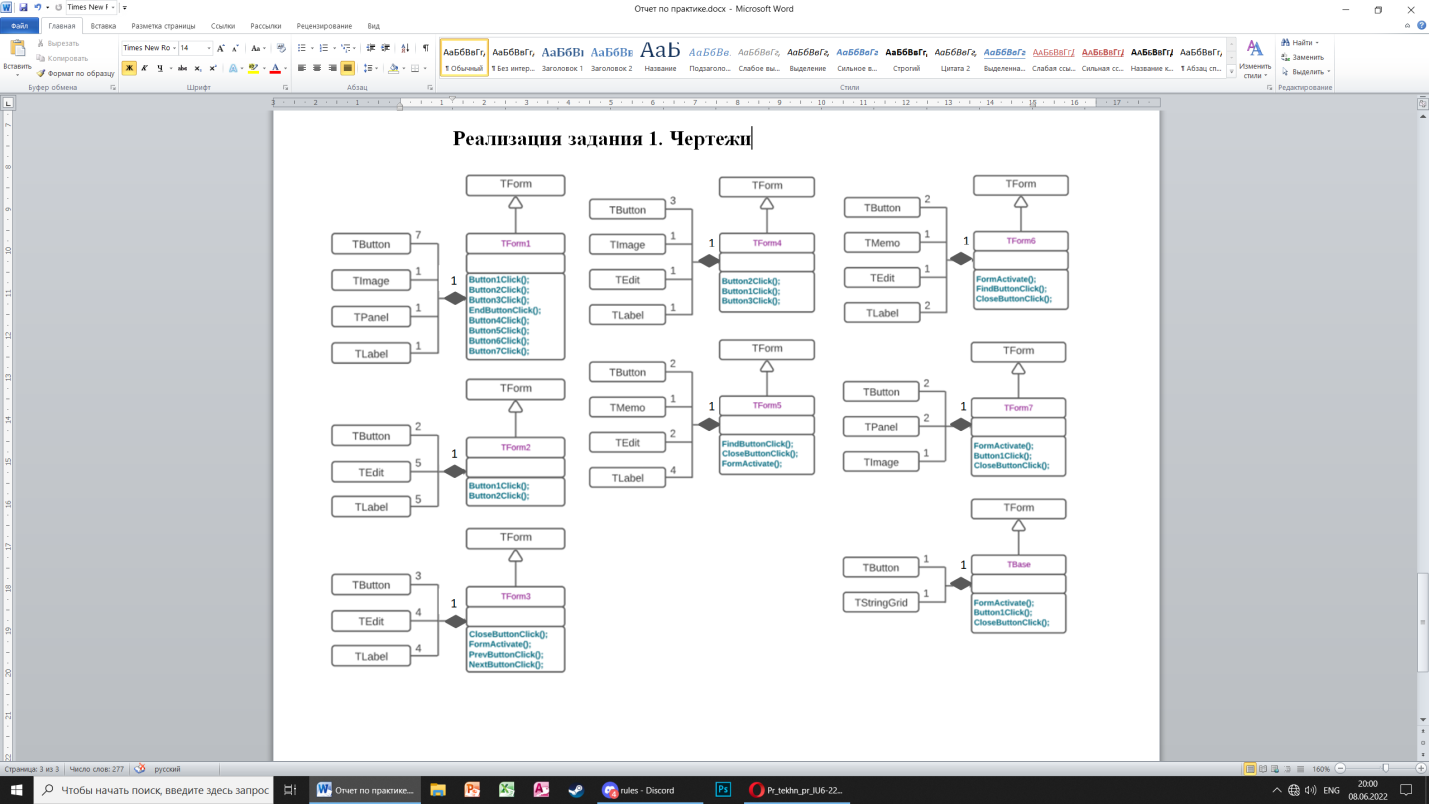
Глава 1

**Анализ задания 1**

В задании 1 необходимо было разработать программную систему на Object Pascal, позволяющую реализовывать добавление и удаление записей, а также поиск записей по определенным параметрам или построения графика. Для реализации данной системы создано приложение, включающее формы главного меню, добавления и удаления записей, поиска, рисования графика и вывода созданной библиотеки, осуществляющее все предполагаемые действия.

**Реализация задания 1. Чертежи**

Диаграмма классов:



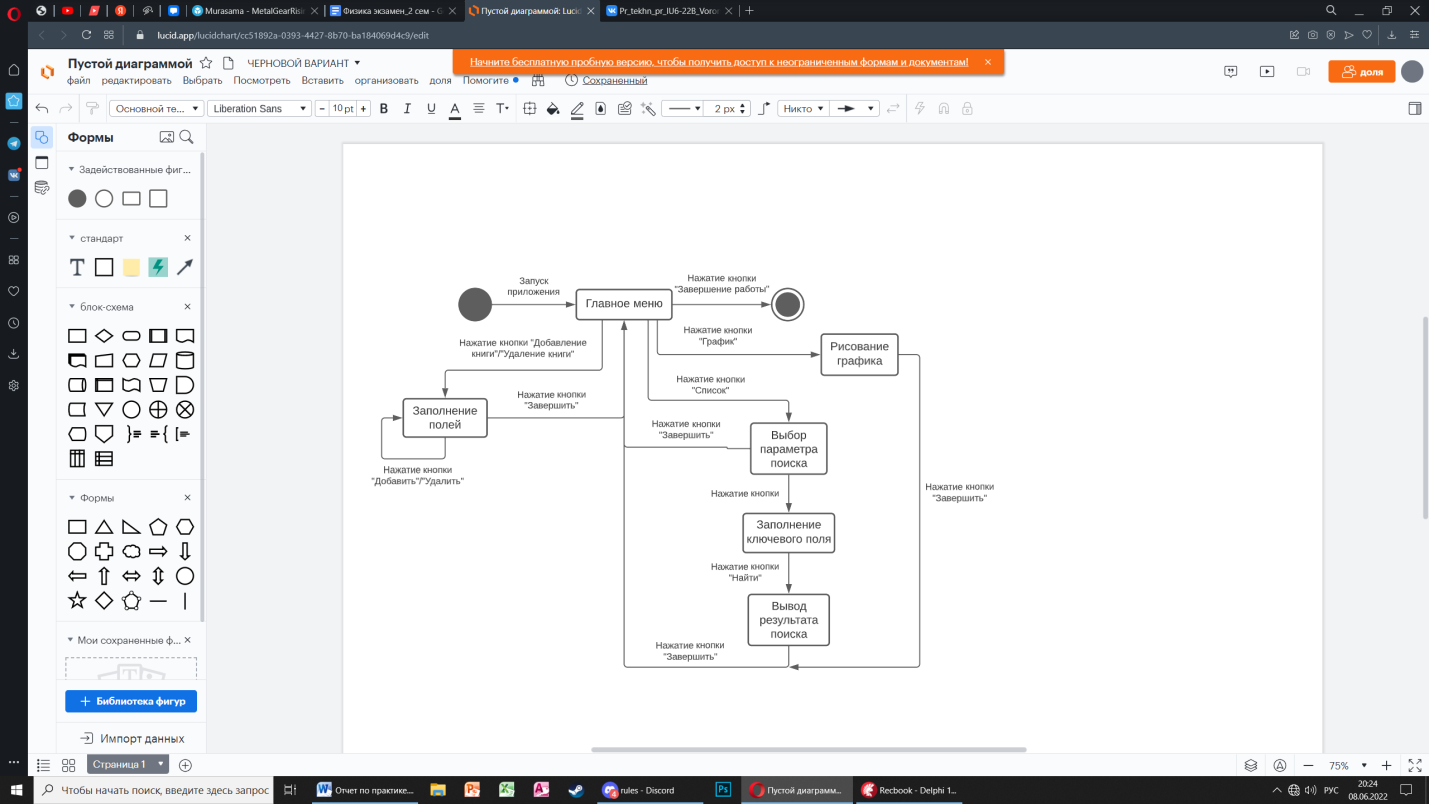


Диаграмма состояний интерфейса:

**Реализация задания 1. Текст программы**

unit Tot\_Unit;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.ExtCtrls,

Vcl.Imaging.jpeg;

type

TForm1 = class(TForm)

Button1: TButton;

Button2: TButton;

EndButton: TButton;

Button3: TButton;

Button4: TButton;

Button5: TButton;

Button6: TButton;

Image1: TImage;

Panel1: TPanel;

Label1: TLabel;

Button7: TButton;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure EndButtonClick(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button5Click(Sender: TObject);

procedure Button6Click(Sender: TObject);

procedure Button7Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

type zap=record

fam:string[22]; //Автор

name:string[22]; //Название

fon:string[22]; //Издательство

yr:integer; //Год

nc:integer; //Число копий

end;

type fz=file of zap;

var

Form1: TForm1;

f:fz;

z:zap;

implementation

uses rsearch, radd, rdelete, raut, rbook, rgraph, rbase;

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

var size:integer;

begin

AssignFile(f,'telefon.dat');

{$I-} Reset(F); {$I+}

if ioresult=0 then

begin

size := FileSize(f);

seek(f,size);

end

else rewrite(f);

Form2.Show;

Form2.edit1.setfocus;

end;

procedure TForm1.Button2Click(Sender: TObject);

begin

AssignFile(f,'telefon.dat');

reset(f);

form4.show;

form4.edit1.setfocus;

end;

procedure TForm1.Button3Click(Sender: TObject);

begin

AssignFile(f,'telefon.dat');

reset(f);

form3.show;

end;

procedure TForm1.Button4Click(Sender: TObject);

begin

AssignFile(f,'telefon.dat');

reset(f);

form5.show;

end;

procedure TForm1.Button5Click(Sender: TObject);

begin

AssignFile(f,'telefon.dat');

reset(f);

form6.show;

end;

procedure TForm1.Button6Click(Sender: TObject);

begin

AssignFile(f,'telefon.dat');

reset(f);

form7.show;

end;

procedure TForm1.Button7Click(Sender: TObject);

begin

AssignFile(f,'telefon.dat');

reset(f);

Base.show;

end;

procedure TForm1.EndButtonClick(Sender: TObject);

begin

Close;

end;

end.

unit radd;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls;

type

TForm2 = class(TForm)

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Edit1: TEdit;

Edit2: TEdit;

Edit3: TEdit;

Edit4: TEdit;

Button1: TButton;

Button2: TButton;

Edit5: TEdit;

Label5: TLabel;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

end;

var

Form2: TForm2;

implementation

uses Tot\_Unit;

{$R \*.dfm}

procedure TForm2.Button1Click(Sender: TObject);

var erp:integer;

begin

z.fam:=edit1.text;

z.name:=edit2.text;

z.fon:=edit3.text;

val(edit4.text,z.yr,erp);

val(edit5.text,z.nc,erp);

edit1.clear;

edit2.clear;

edit3.clear;

edit4.clear;

edit5.clear;

write(f,z);

edit1.setfocus;

end;

procedure TForm2.Button2Click(Sender: TObject);

begin

reset(f);

closefile(f);

self.hide;

end;

end.

unit rdelete;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls,Tot\_Unit, Vcl.ExtCtrls,

Vcl.Imaging.jpeg;

type

TForm4 = class(TForm)

Edit1: TEdit;

Label1: TLabel;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Image1: TImage;

procedure Button2Click(Sender:TObject);

procedure Button1Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

end;

var

Form4: TForm4;

implementation

{$R \*.dfm}

procedure TForm4.Button2Click(Sender:TObject);

var k,j,i:integer;

b:zap;

begin

k:=0;i:=0;

while i<=filesize(f)-2-k do{последнюю пока не проверяем, выйдем за файл}

begin

seek(f,i);

read(f,z);

if z.name=edit1.text then

begin

k:=k+1; {считаем}

for j:=i to filesize(f)-2 do{идем вперед}

begin

seek(f,j+1);{переставляем - сдвигаем - записи вверх на 1}

read(f,b);

seek(f,j);

write(f,b);

end

end

else i:=i+1;{если нет вперед}

end;

seek(f,filesize(f)-1-k);{проверяем последнюю, она сдвинулась на к вверх}

read(f,z);

if z.name=edit1.text then k:=k+1;

seek(f,filesize(f)-k);{встаем в последнюю не удаленную запись}

truncate(f);{обрезаем файл}

edit1.clear;

end;

procedure TForm4.Button3Click(Sender:TObject);

begin

closefile(f);

deletefile('telefon.dat');

AssignFile(f,'telefon.dat');

rewrite(f);

end;

procedure TForm4.Button1Click(Sender: TObject);

begin

reset(f);

closefile(f);

self.hide;

end;

end.

unit rsearch;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls;

type

TForm3 = class(TForm)

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Edit1: TEdit;

Edit2: TEdit;

Edit3: TEdit;

Edit4: TEdit;

PrevButton: TButton;

NextButton: TButton;

CloseButton: TButton;

procedure CloseButtonClick(Sender: TObject);

procedure FormActivate(Sender: TObject);

procedure PrevButtonClick(Sender: TObject);

procedure NextButtonClick(Sender: TObject);

end;

var

Form3: TForm3;

j:integer;

implementation

uses Tot\_Unit;

{$R \*.dfm}

var mas:array[1..20] of zap;

c:shortstring;

procedure TForm3.FormActivate(Sender: TObject);

var i:integer;

begin

i:=1;

while not eof(f) do begin

read(f,z);

if z.nc=1 then begin

mas[i]:=z;

inc(i);

end;

end;

reset(f);

edit1.text:=mas[1].fam;

edit2.text:=mas[1].name;

edit3.text:=mas[1].fon;

str(mas[1].yr,c);

edit4.text:=c;

j:=1;

end;

procedure TForm3.NextButtonClick(Sender: TObject);

begin

if mas[j+1].fam<>'' then begin

inc(j);

edit1.text:=mas[j].fam;

edit2.text:=mas[j].name;

edit3.text:=mas[j].fon;

str(mas[j].yr,c);

edit4.text:=c;

end;

end;

procedure TForm3.PrevButtonClick(Sender: TObject);

begin

if j>1 then begin

j:=j-1;

edit1.text:=mas[j].fam;

edit2.text:=mas[j].name;

edit3.text:=mas[j].fon;

str(mas[j].yr,c);

edit4.text:=c;

end;

end;

procedure TForm3.CloseButtonClick(Sender: TObject);

begin

reset(f);

closefile(f);

self.hide;

end;

end.

unit raut;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls;

type

TForm5 = class(TForm)

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

LeftBar: TEdit;

RightBar: TEdit;

FindButton: TButton;

Label4: TLabel;

CloseButton: TButton;

Output: TMemo;

procedure FindButtonClick(Sender:TObject);

procedure CloseButtonClick(Sender:TObject);

procedure FormActivate(Sender:TObject);

end;

var

Form5: TForm5;

implementation

uses Tot\_Unit;

{$R \*.dfm}

procedure TForm5.FormActivate(Sender:TObject);

var i:integer;

begin

Output.Lines.Clear;

end;

procedure TForm5.FindButtonClick(Sender: TObject);

var i,j,k,erc,vyr:integer;

begin

val(LeftBar.text,i,erc);

val(RightBar.text,j,erc);

while not eof(f) do begin

read(f,z);

if (z.yr>=i)and(z.yr<=j) then Output.Lines.Add(z.fam);

end;

end;

procedure TForm5.CloseButtonClick(Sender: TObject);

begin

LeftBar.clear;

RightBar.clear;

reset(f);

closefile(f);

self.hide;

end;

end.

unit rbook;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls;

type

TForm6 = class(TForm)

Label1: TLabel;

Edit1: TEdit;

Memo1: TMemo;

Label2: TLabel;

FindButton: TButton;

CloseButton: TButton;

procedure FormActivate(Sender:TObject);

procedure FindButtonClick(Sender:TObject);

procedure CloseButtonClick(Sender:TObject);

end;

var

Form6: TForm6;

implementation

uses Tot\_Unit;

{$R \*.dfm}

var mas:array[1..20] of zap;

i:integer;

procedure TForm6.FormActivate(Sender:TObject);

begin

i:=1;

Memo1.Lines.Clear;

end;

procedure TForm6.FindButtonClick(Sender:TObject);

var j:integer;

p:zap;

year:shortstring;

begin

Memo1.Lines.Clear;

i:=1;

while not eof(f) do begin

read(f,z);

if z.fon=Edit1.text then begin

mas[i]:=z;

inc(i);

end;

end;

for i:=1 to 20 do begin

for j:=1 to 20-i do begin

if mas[j].yr>mas[j+1].yr then begin

p:=mas[j];

mas[j]:=mas[j+1];

mas[j+1]:=P;

end;

end;

end;

for i := 1 to 20 do if mas[i].yr<>0 then begin

str(mas[i].yr,year);

year:=mas[i].name+', '+year;

Memo1.Lines.Add(year);

end;

end;

procedure TForm6.CloseButtonClick(Sender:TObject);

begin

for i := 1 to 20 do if mas[i].yr<>0 then begin

mas[i].yr:=0;

mas[i].name:='';

end;

reset(f);

closefile(f);

self.hide;

end;

end.

unit rgraph;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.ExtCtrls,

RSChartPanel, RSCharts;

type

TForm7 = class(TForm)

Panel1: TPanel;

CloseButton: TButton;

Image1: TImage;

Button1: TButton;

Panel2: TPanel;

procedure FormActivate(Sender:TObject);

procedure CloseButtonClick(Sender:TObject);

procedure Button1Click(Sender: TObject);

end;

var

Form7: TForm7;

implementation

uses Tot\_Unit;

{$R \*.dfm}

var i,x,w,minyear,maxyear,maxcopy:integer;

a:array of integer;

ait,cnt:shortstring;

h:real;

procedure TForm7.FormActivate(Sender:TObject);

begin

Image1.Canvas.Pen.Color:=clWhite;

Image1.Canvas.Rectangle(Rect(0,0,image1.width,image1.height));

Image1.Canvas.Pen.Color:=clBlack;

Image1.Canvas.moveto(10,image1.height-15);

Image1.Canvas.lineto(10,10);

Image1.Canvas.TextOut(10,10,'Кол-во экземпляров,шт');

Image1.Canvas.Moveto(10,10);

Image1.Canvas.moveto(10,image1.height-15);

Image1.Canvas.lineto(image1.width-10,image1.height-15);

Image1.Canvas.textout(image1.width-33,image1.height-15,'Год');

read(f,z);

minyear:=z.yr;

maxyear:=z.yr;

while not(eof(f)) do begin

read(f,z);

if z.yr>maxyear then maxyear:=z.yr;

if z.yr<minyear then minyear:=z.yr;

end;

setlength(a,maxyear-minyear+1);

for i := 0 to length(a)-1 do a[i]:=0;

reset(f);

while not(eof(f)) do begin

read(f,z);

a[z.yr-minyear]:=a[z.yr-minyear]+z.nc;

end;

for i := 0 to length(a)-1 do if a[i]>maxcopy then maxcopy:=a[i];

w:=(image1.width-40) div length(a);

h:=(image1.height-60) / maxcopy;

end;

procedure TForm7.Button1Click(Sender: TObject);

begin

Image1.Canvas.moveto(10,image1.height-15);

x:=10;

for i := 0 to length(a)-1 do begin

str(minyear+i,cnt);

Image1.Canvas.TextOut(x+(w div 2)-10,image1.height-13,cnt);

Image1.Canvas.MoveTo(x,image1.height-15);

Image1.Canvas.LineTo(x,image1.height-15-round(a[i]\*h));

if a[i]<>0 then begin

str(a[i],ait);

Image1.Canvas.TextOut(x+(w div 2)-10,image1.height-30-round(a[i]\*h),ait);

end

else Image1.Canvas.TextOut(x+(w div 2)-10,image1.height-30-round(a[i]\*h),'0');

Image1.Canvas.MoveTo(x,image1.height-15-round(a[i]\*h));

Image1.Canvas.LineTo(x+w,image1.height-15-round(a[i]\*h));

Image1.Canvas.LineTo(x+w,image1.height-15);

x:=x+w;

end;

end;

procedure TForm7.CloseButtonClick(Sender:TObject);

begin

reset(f);

closefile(f);

self.hide;

end;

end.

unit rbase;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Grids;

type

TBase = class(TForm)

StringGrid1: TStringGrid;

EndButton: TButton;

procedure FormActivate(Sender:TObject);

procedure EndButtonClick(Sender:TObject);

end;

var

Base: TBase;

implementation

{$R \*.dfm}

uses Tot\_Unit;

var i,j:integer;

numc,year,helpstr:shortstring;

procedure TBase.FormActivate(Sender:TObject);

begin;

StringGrid1.ColCount:=6;

StringGrid1.RowCount:=21;

i:=1;

while not eof(f) do begin

read(f,z);

StringGrid1.RowHeights[i]:=30;

for j := 1 to 3 do StringGrid1.ColWidths[j]:=180;

StringGrid1.ColWidths[4]:= 100;

StringGrid1.ColWidths[5]:= 150;

StringGrid1.ColWidths[0]:= 30;

StringGrid1.Cells[1,0]:='Автор';

StringGrid1.Cells[2,0]:='Название книги';

StringGrid1.Cells[3,0]:='Издательство';

StringGrid1.Cells[4,0]:='Год издания';

StringGrid1.Cells[5,0]:='Количество копий';

str(i,helpstr);

StringGrid1.Cells[0,i]:=helpstr+'.';

StringGrid1.Cells[1,i]:=z.fam;

StringGrid1.Cells[2,i]:=z.name;

StringGrid1.Cells[3,i]:=z.fon;

str(z.yr,year);

StringGrid1.Cells[4,i]:=year;

str(z.nc,numc);

StringGrid1.Cells[5,i]:=numc;

inc(i);

end;

end;

procedure TBase.EndButtonClick(Sender:TObject);

begin;

for i := 1 to 20 do for j := 1 to 5 do StringGrid1.Cells[j,i]:='';

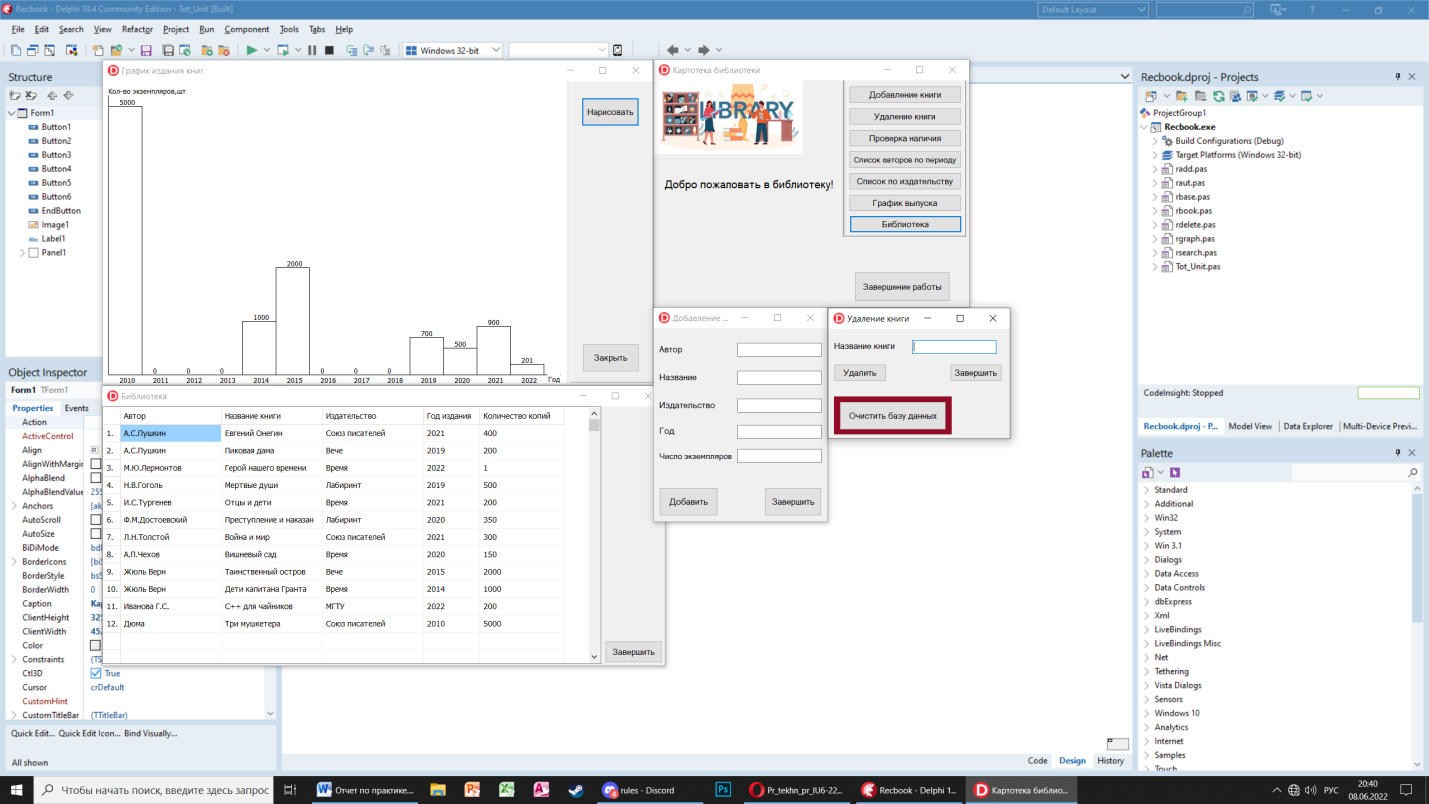
reset(f);

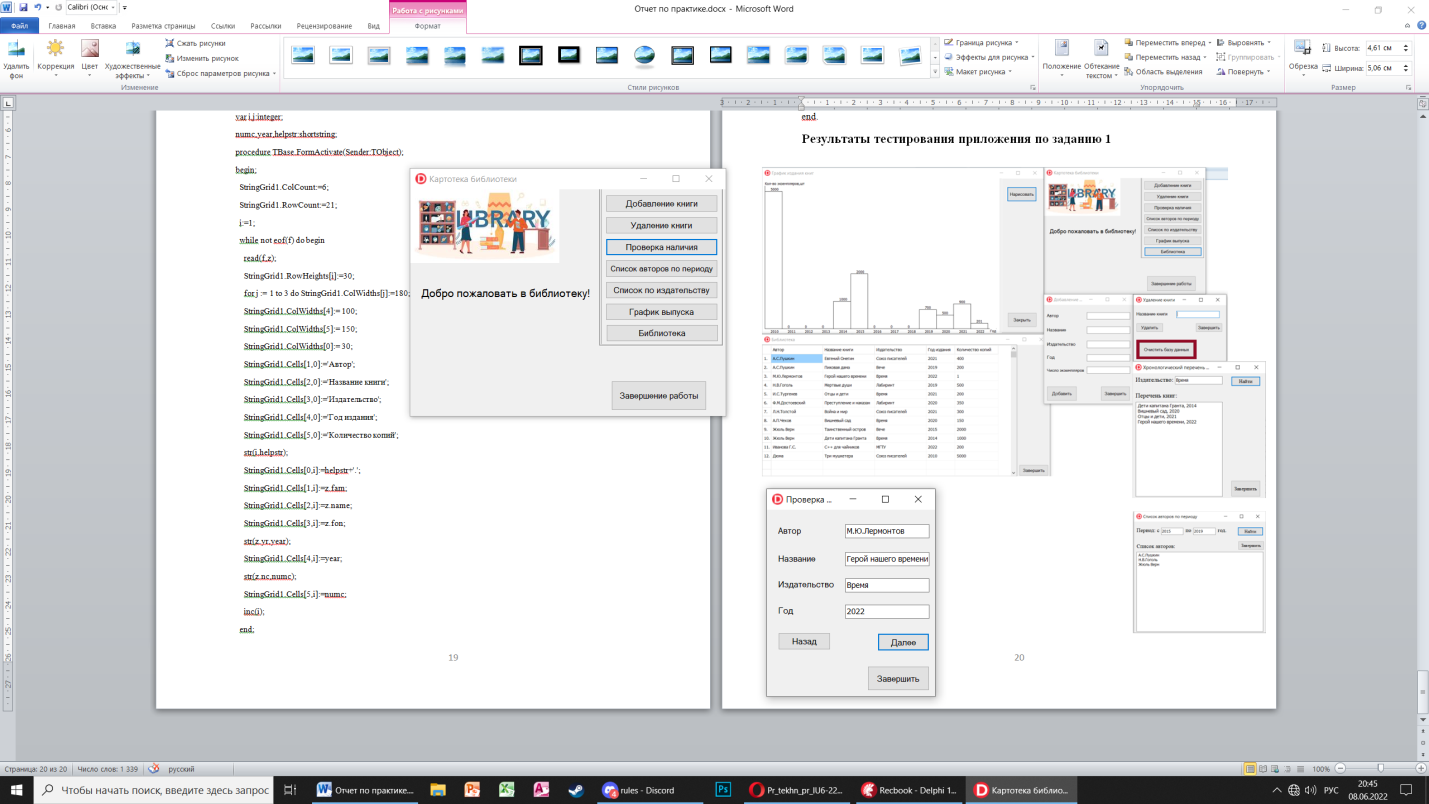
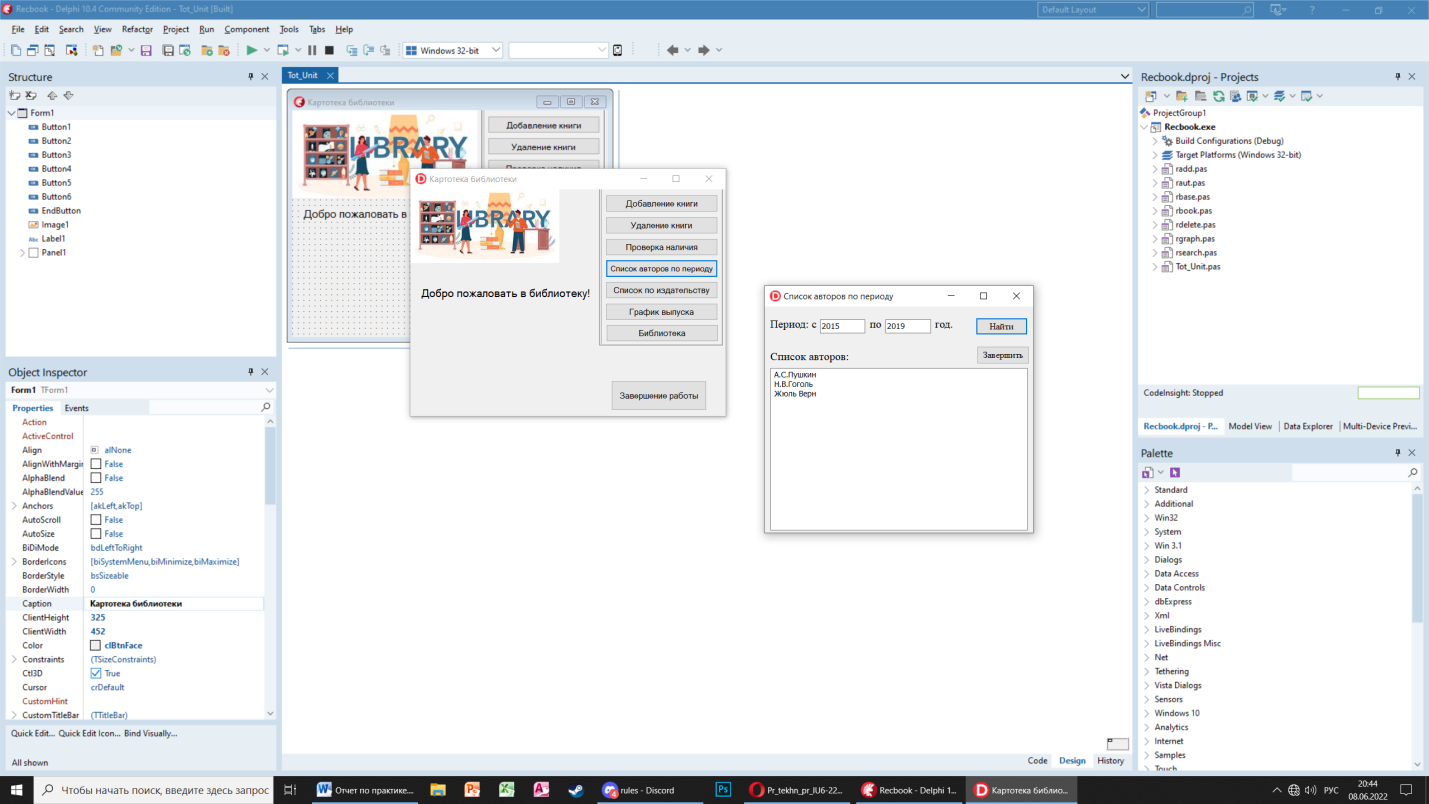
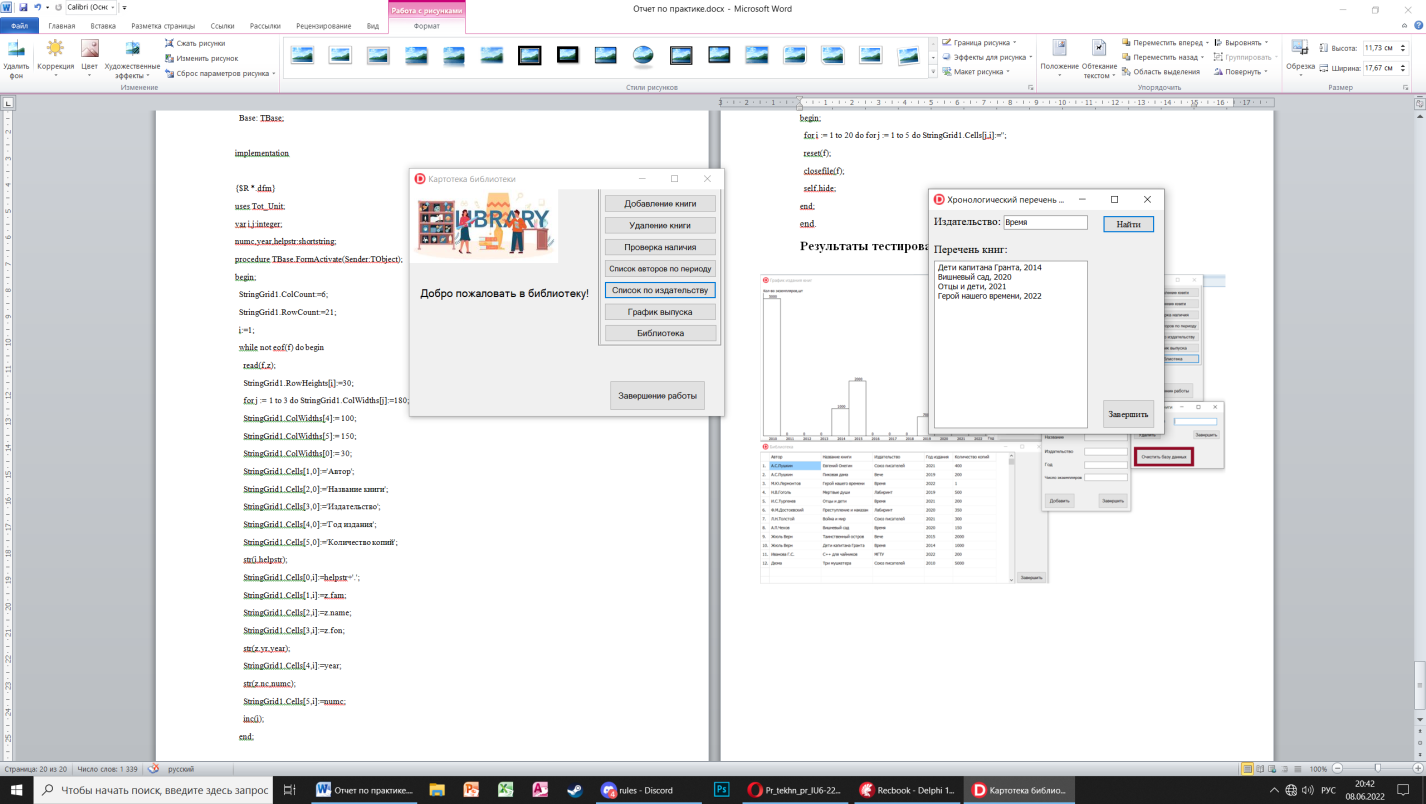
closefile(f);

self.hide;

end;

end.

**Результаты тестирования приложения по заданию 1**



**Вывод по заданию 1**

В результате работы создано приложение, позволяющее реализовывать следующие функции:

• Добавление издания

• Удаление издания

• Поиск по параметру

• Построение графика

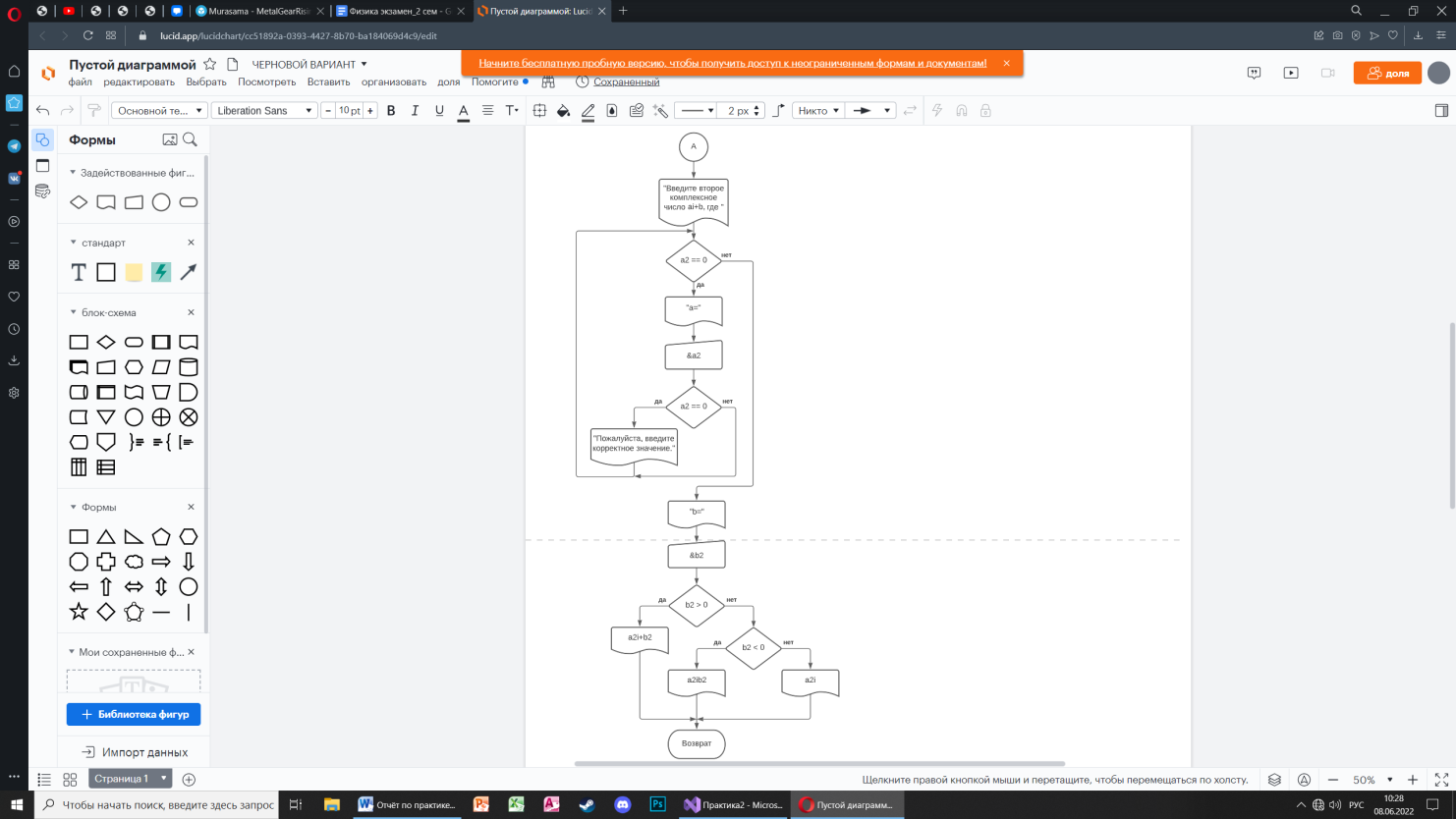
На заданных исходных данных система работает корректно.

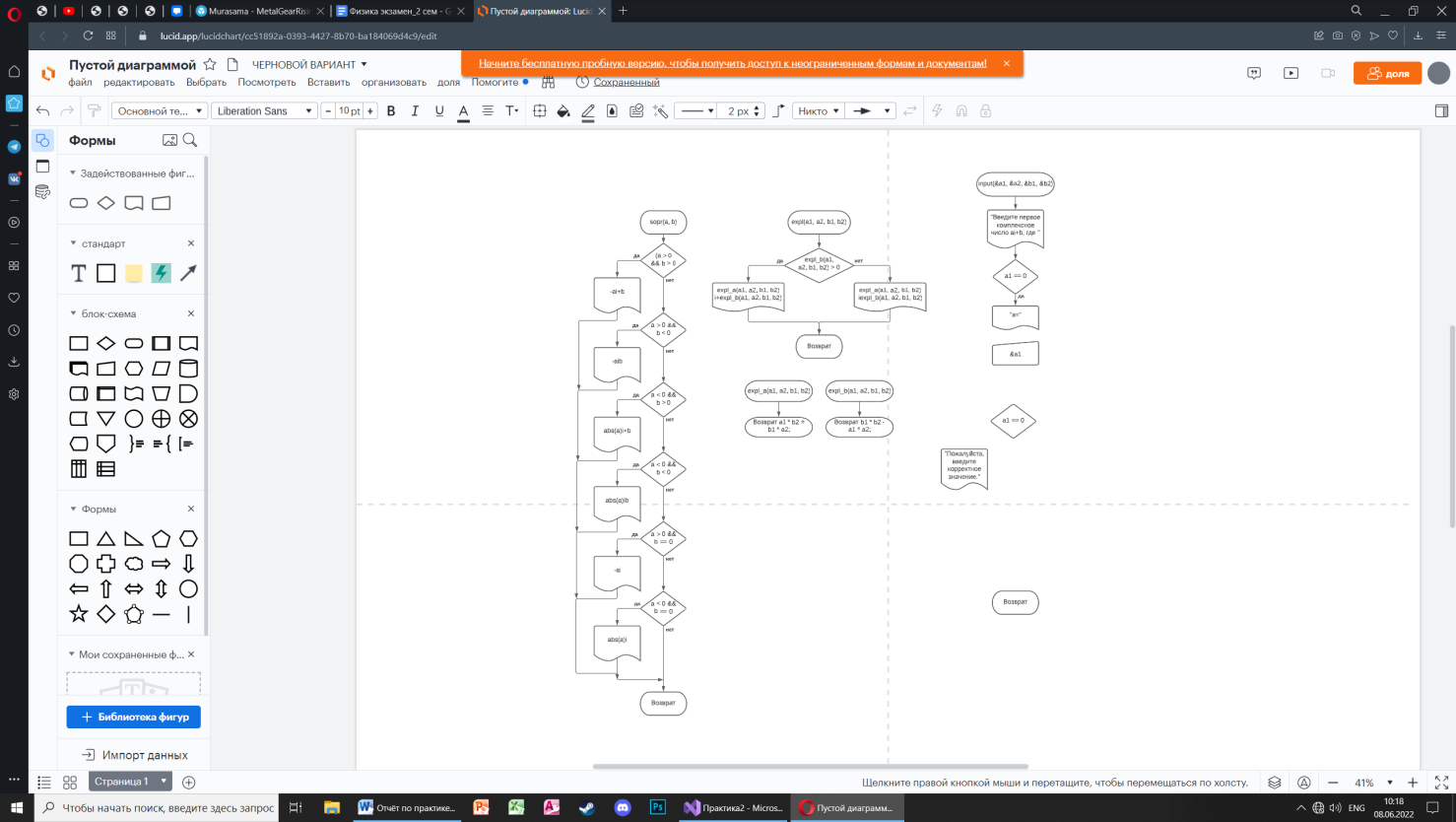
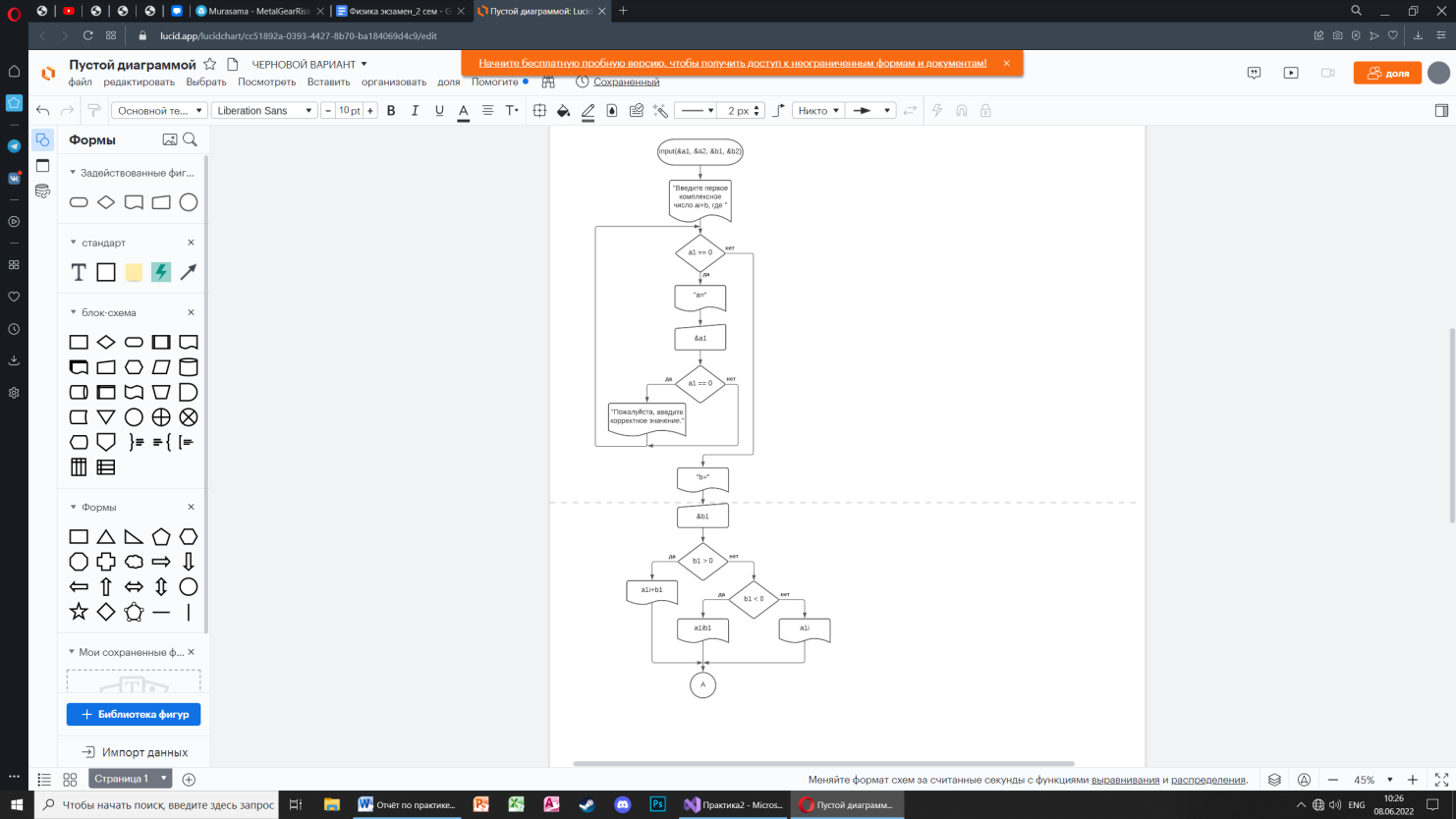
Глава 2

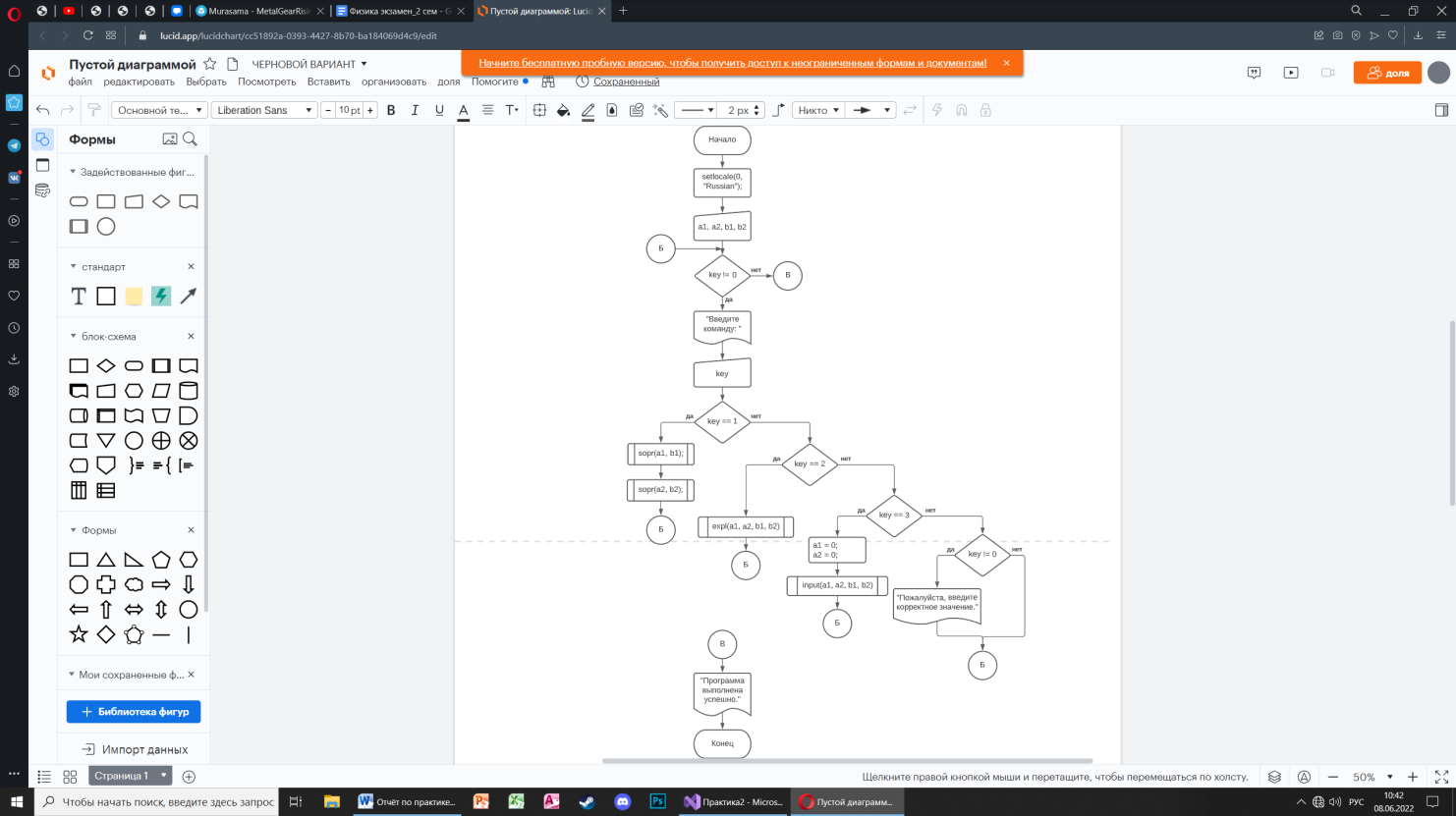
**Анализ задания 2**

В задании 2 необходимо было разработать консольное приложение на C++, позволяющее реализовывать операции над комплексными числами: ввод, поиск сопряженного числа, умножение двух чисел, а также вывод результатов операций на экран.

Для реализации данной системы реализована программа, содержащая три подпрограммы: ввод новых значений, поиск сопряженного числа, умножение двух чисел.

**Реализация задания 2. Чертежи**





**Реализация задания 2. Текст программы**

using namespace std;

#include <iostream>

#include <string.h>

#include <conio.h>

#include <stdio.h>

#include <locale.h>

int expl\_a(int a1, int a2, int b1, int b2)

{

return a1 \* b2 + b1 \* a2;

}

int expl\_b(int a1, int a2, int b1, int b2)

{

return b1 \* b2 - a1 \* a2;

}

void sopr(int a, int b)

{

if (a > 0 && b > 0) cout << "Число, сопряженное первому: " << "-" << a << "i" << "+" << b << "\n";

else if (a > 0 && b < 0) cout << "Число, сопряженное первому: " << "-" << a << "i" << b << "\n";

else if (a < 0 && b > 0) cout << "Число, сопряженное первому: " << abs(a) << "i" << "+" << b << "\n";

else if (a < 0 && b < 0) cout << "Число, сопряженное первому: " << abs(a) << "i" << b << "\n";

else if (a > 0 && b == 0) cout << "Число, сопряженное первому: " << "-" << a << "i" << "\n";

else if (a < 0 && b == 0) cout << "Число, сопряженное первому: " << abs(a) << "i" << "\n";

}

void expl(int a1, int a2, int b1, int b2)

{

if (expl\_b(a1, a2, b1, b2) > 0) cout << "Результат: " << expl\_a(a1, a2, b1, b2) << "i" << "+" << expl\_b(a1, a2, b1, b2) << "\n";

else cout << "Результат: " << expl\_a(a1, a2, b1, b2) << "i" << expl\_b(a1, a2, b1, b2) << "\n";

}

void input(int &a1, int &a2, int &b1, int &b2)

{

cout << "Введите первое комплексное число ai+b, где " << "\n";

while (a1 == 0) {

printf("%s", "a=");

scanf\_s("%d", &a1);

if (a1 == 0) cout << "Пожалуйста, введите корректное значение." << "\n";

}

printf("%s", "b=");

scanf\_s("%d", &b1);

if (b1 > 0) cout << "Первое комплексное число: " << a1 << "i" << "+" << b1;

else if (b1 < 0)cout << "Первое комплексное число: " << a1 << "i" << b1;

else cout << "Первое комплексное число: " << a1 << "i";

cout << "\n";

cout << "Введите второе комплексное число ai+b, где " << "\n";

while (a2 == 0) {

printf("%s", "a=");

scanf\_s("%d", &a2);

if (a2 == 0) cout << "Пожалуйста, введите корректное значение." << "\n";

}

printf("%s", "b=");

scanf\_s("%d", &b2);

if (b2 > 0) cout << "Второе комплексное число: " << a2 << "i" << "+" << b2;

else if (b2 < 0)cout << "Второе комплексное число: " << a2 << "i" << b2;

else cout << "Второе комплексное число: " << a2 << "i";

}

int main()

{

int a1=0, a2=0, b1, b2, key=-1;

setlocale(0, "Russian");

input(a1, a2, b1, b2);

cout << "\n-----------------------------------------------------";

cout << "\n" << "Для поиска сопряженного числа введите 1" << "\n" << "Для умножения первого числа на второе число введите 2" << "\n" << "Для ввода новых значений введите 3" << "\n" << "Для выхода введите 0" << "\n";

cout << "-----------------------------------------------------\n";

while (key != 0) {

cout << "Введите команду: ";

cin >> key;

cout << "\n";

if (key == 1) {

sopr(a1, b1);

sopr(a2, b2);

}

else if (key == 2) expl(a1, a2, b1, b2);

else if (key == 3) {

a1 = 0; a2 = 0;

input(a1, a2, b1, b2);

cout << "\n";

}

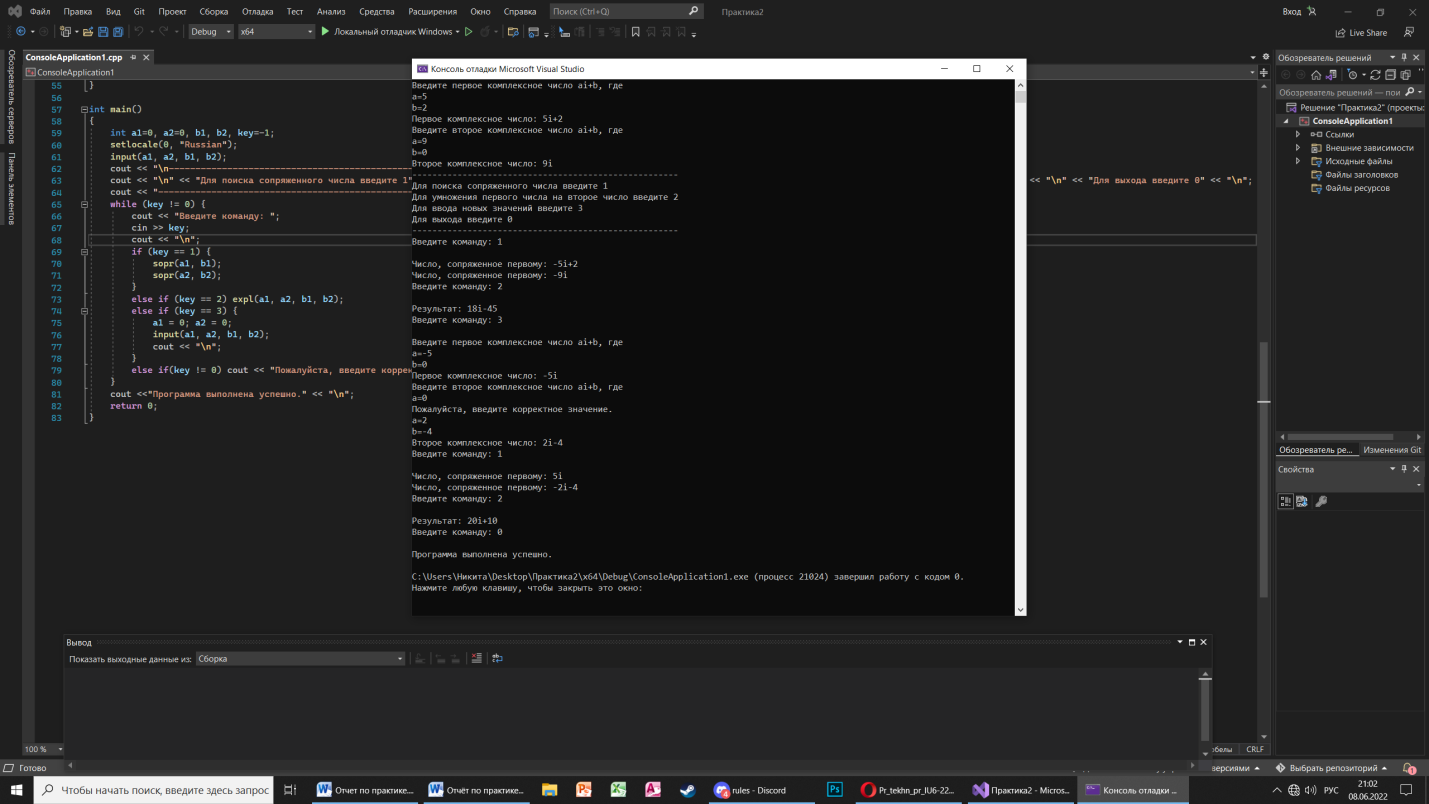
else if(key != 0) cout << "Пожалуйста, введите корректное значение."<<"\n";

}

cout <<"Программа выполнена успешно." << "\n";

return 0;

}

**Результаты тестирования программы по заданию 2**

**Вывод по заданию 2**

В результате работы создана программа, позволяющая реализовывать следующие функции:

• Ввод комплексного числа

• Поиск сопряженного числа

• Умножение двух комплексных чисел

• Вывод результатов операций на экран

На заданных исходных данных система работает корректно.

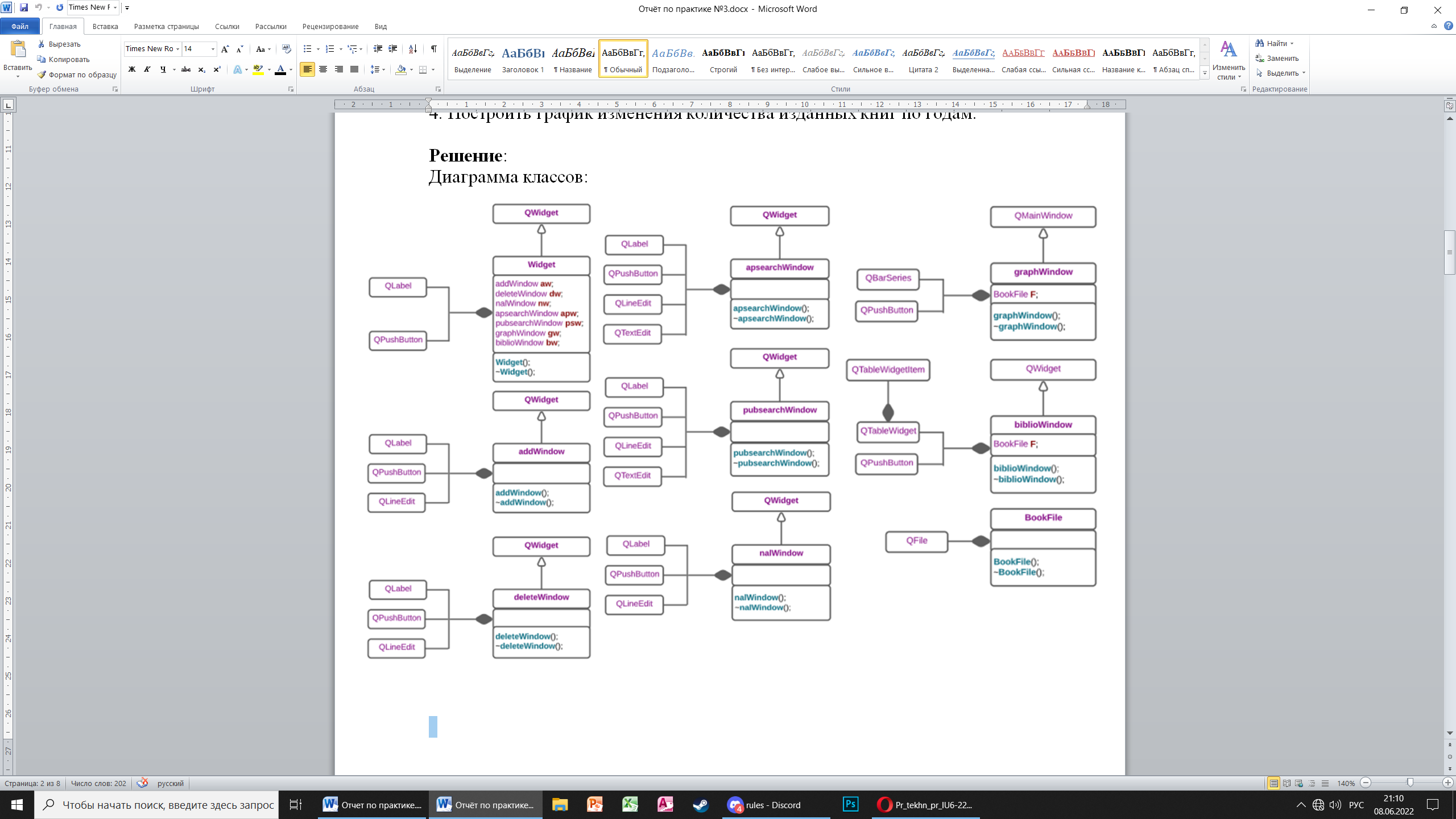
Глава 3

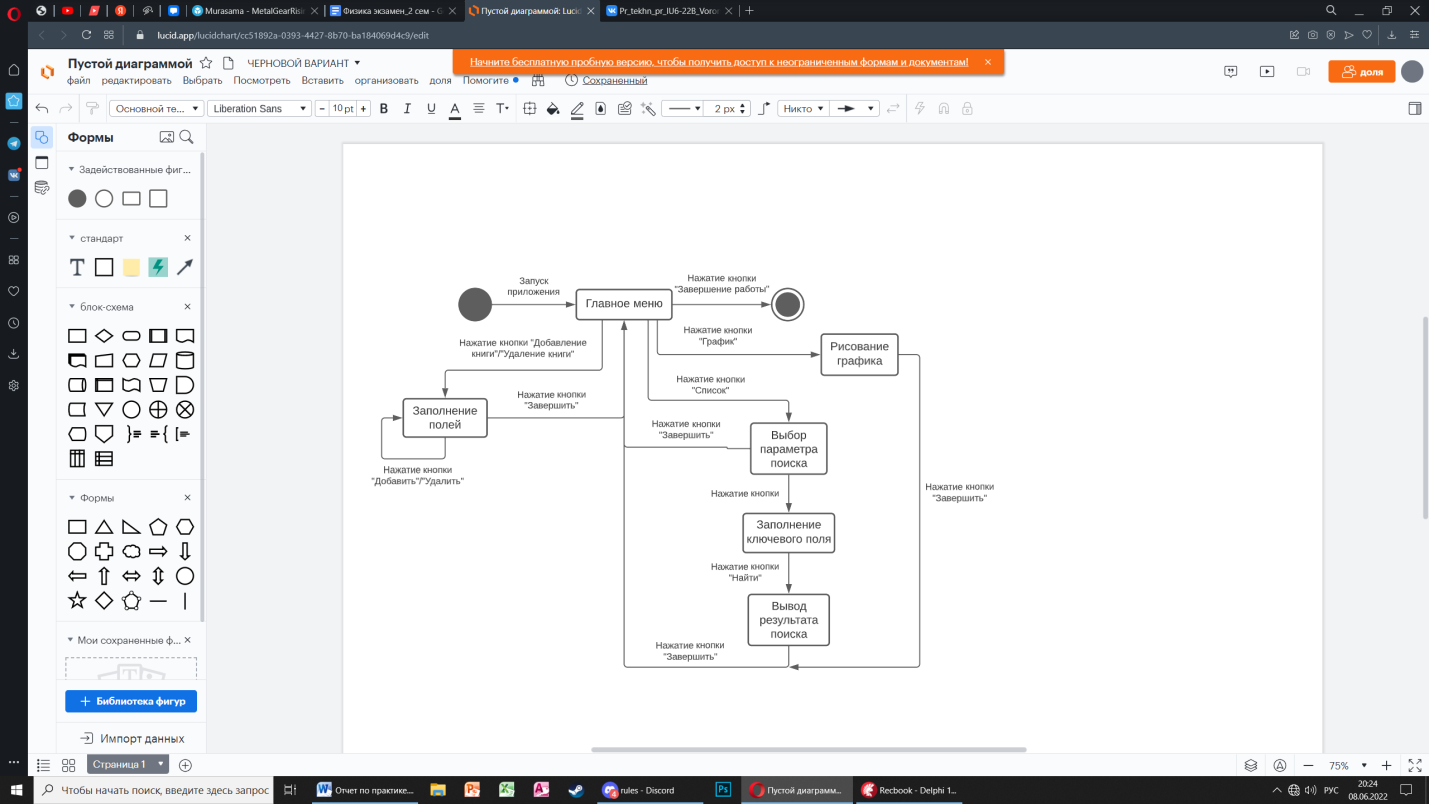
**Анализ задания 3**

В задании 3 необходимо было разработать программную систему на C++ c использованием библиотеки Qt, позволяющую реализовывать добавление и удаление записей, а также поиск записей по определенным параметрам или построения графика.

Для реализации данной системы создано приложение, включающее формы главного меню, добавления и удаления записей, поиска, рисования графика и вывода созданной библиотеки, осуществляющее все предполагаемые действия.

**Реализация задания 3. Чертежи**

Диаграмма классов:

Диаграмма состояний интерфейса:

**Реализация задания 3. Текст программы**

**addwindow.h**

#ifndef ADDWINDOW\_H

#define **ADDWINDOW\_H**

#include "file.h"

#include <QWidget>

#include <QLabel>

#include <QPushButton>

#include <QVBoxLayout>

#include <QHBoxLayout>

#include <QLineEdit>

#include <QTextEdit>

#include <QPainter>

#include <QTableWidget>

class **addWindow** : public QWidget

{

Q\_OBJECT

public:

BookFile **F**;

QWidget \***addWin**;

QLabel \***author**;

QLabel \***name**;

QLabel \***publishment**;

QLabel \***year**;

QLabel \***copynum**;

QLineEdit \***Tauthor**;

QLineEdit \***Tname**;

QLineEdit \***Tpublishment**;

QLineEdit \***Tyear**;

QLineEdit \***Tcopynum**;

QPushButton \***add**;

QPushButton \***end**;

QVBoxLayout \***mainlay**;

QVBoxLayout \***Labellay**;

QVBoxLayout \***Textlay**;

QHBoxLayout \***Editlay**;

QHBoxLayout \***btnlay**;

public:

**addWindow**(QWidget \***parent** = 0);

~**addWindow**();

public slots:

void **addbook**();

};

#endif // ADDWINDOW\_H

**apsearchwindow.h**

#ifndef APSEARCHWINDOW\_H

#define **APSEARCHWINDOW\_H**

#include <QWidget>

#include <QLabel>

#include <QPushButton>

#include <QVBoxLayout>

#include <QHBoxLayout>

#include <QLineEdit>

#include <QTextEdit>

#include <QPainter>

#include <QTableWidget>

#include "file.h"

class **apsearchWindow** : public QWidget

{

Q\_OBJECT

public:

BookFile **F**;

QLabel \***L1**;

QLabel \***L2**;

QLabel \***L3**;

QLabel \***Llist**;

QLineEdit \***LBord**;

QLineEdit \***RBord**;

QPushButton \***search**;

QPushButton \***end**;

QTextEdit \***list**;

QVBoxLayout \***mainlay**;

QVBoxLayout \***Labellay**;

QVBoxLayout \***btnlay**;

QHBoxLayout \***bordlay**;

QHBoxLayout \***actlay**;

public:

**apsearchWindow**(QWidget \***parent** = 0);

~**apsearchWindow**();

public slots:

void **searchfor**();

};

#endif // APSEARCHWINDOW\_H

**bibliowindow.h**

#ifndef BIBLIOWINDOW\_H

#define **BIBLIOWINDOW\_H**

#include "file.h"

#include <QWidget>

#include <QLabel>

#include <QPushButton>

#include <QVBoxLayout>

#include <QHBoxLayout>

#include <QLineEdit>

#include <QTextEdit>

#include <QPainter>

#include <QTableWidget>

class **biblioWindow** : public QWidget

{

Q\_OBJECT

public:

BookFile **F**;

QTableWidget \***Table**;

QPushButton \***end**;

QHBoxLayout \***mainlay**;

public:

**biblioWindow**(QWidget \***parent** = 0);

~**biblioWindow**();

public slots:

void **Tshow**();

};

#endif // BIBLIOWINDOW\_H

**deletewindow.h**

#ifndef DELETEWINDOW\_H

#define **DELETEWINDOW\_H**

#include "file.h"

#include <QWidget>

#include <QLabel>

#include <QPushButton>

#include <QVBoxLayout>

#include <QHBoxLayout>

#include <QLineEdit>

#include <QTextEdit>

#include <QPainter>

#include <QTableWidget>

class **deleteWindow** : public QWidget

{

Q\_OBJECT

public:

BookFile **F**;

QLabel \***name**;

QLineEdit \***Tname**;

QPushButton \***del**;

QPushButton \***end**;

QPushButton \***delAll**;

QVBoxLayout \***mainlay**;

QHBoxLayout \***namelay**;

QHBoxLayout \***btnlay**;

public:

**deleteWindow**(QWidget \***parent** = 0);

~**deleteWindow**();

public slots:

void **delbook**();

void **delall**();

};

#endif // DELETEWINDOW\_H

**graphwindow.h**

#ifndef GRAPHWINDOW\_H

#define GRAPHWINDOW\_H

#include "file.h"

#include <QtCharts>

#include <QMainWindow>

#include <QLegend>

#include <QPushButton>

#include <QVBoxLayout>

#include <QHBoxLayout>

class graphWindow : public QMainWindow

{

Q\_OBJECT

public:

BookFile F;

QPushButton \*paint;

QPushButton \*end;

QHBoxLayout \*mainlay;

QVBoxLayout \*btnlay;

QBarSeries \*graph;

public:

graphWindow(QWidget \*parent = 0);

~graphWindow();

public slots:

void activate();

};

#endif // GRAPHWINDOW\_H

**nalwindow.h**

#ifndef NALWINDOW\_H

#define **NALWINDOW\_H**

#include "file.h"

#include <QWidget>

#include <QLabel>

#include <QPushButton>

#include <QVBoxLayout>

#include <QHBoxLayout>

#include <QLineEdit>

#include <QTextEdit>

#include <QPainter>

#include <QTableWidget>

class **nalWindow** : public QWidget

{

Q\_OBJECT

public:

BookFile **F**;

QLabel \***author**;

QLabel \***name**;

QLabel \***publishment**;

QLabel \***year**;

QLineEdit \***Tauthor**;

QLineEdit \***Tname**;

QLineEdit \***Tpublishment**;

QLineEdit \***Tyear**;

QPushButton \***back**;

QPushButton \***next**;

QPushButton \***end**;

QVBoxLayout \***mainlay**;

QVBoxLayout \***Labellay**;

QVBoxLayout \***Textlay**;

QHBoxLayout \***Editlay**;

QHBoxLayout \***btnlay**;

int **j**;

book **mas**[20];

book **b**;

public:

**nalWindow**(QWidget \***parent** = 0);

~**nalWindow**();

public slots:

void **nextbutton**();

void **prevbutton**();

};

#endif // NALWINDOW\_H

**pubsearchwindow.h**

#ifndef PUBSEARCHWINDOW\_H

#define **PUBSEARCHWINDOW\_H**

#include "file.h"

#include <QWidget>

#include <QLabel>

#include <QPushButton>

#include <QVBoxLayout>

#include <QHBoxLayout>

#include <QLineEdit>

#include <QTextEdit>

#include <QPainter>

#include <QTableWidget>

class **pubsearchWindow** : public QWidget

{

Q\_OBJECT

public:

BookFile **F**;

QLabel \***pub**;

QLabel \***list**;

QLineEdit \***Tpub**;

QPushButton \***search**;

QPushButton \***end**;

QTextEdit \***Tlist**;

QHBoxLayout \***mainlay**;

QVBoxLayout \***Labellay**;

QVBoxLayout \***btnlay**;

QHBoxLayout \***bordlay**;

QVBoxLayout \***actlay**;

public:

**pubsearchWindow**(QWidget \***parent** = 0);

~**pubsearchWindow**();

public slots:

void **searchfor**();

};

#endif // PUBSEARCHWINDOW\_H

**widget.h**

#ifndef WIDGET\_H

#define WIDGET\_H

#include <QWidget>

#include <QLabel>

#include <QPushButton>

#include <QVBoxLayout>

#include <QHBoxLayout>

#include <QLineEdit>

#include <QTextEdit>

#include <QPainter>

#include <QTableWidget>

#include "addWindow.h"

#include "apsearchWindow.h"

#include "biblioWindow.h"

#include "deleteWindow.h"

#include "graphWindow.h"

#include "nalWindow.h"

#include "pubsearchWindow.h"

class Widget : public QWidget

{

Q\_OBJECT

public:

QLabel \*welcome;

QPushButton \*add;

QPushButton \*del;

QPushButton \*nal;

QPushButton \*authper;

QPushButton \*izd;

QPushButton \*graph;

QPushButton \*biblio;

QPushButton \*end;

QVBoxLayout \*btnlay;

QVBoxLayout \*endlay;

QHBoxLayout \*mainlay;

addWindow aw;

deleteWindow dw;

nalWindow nw;

apsearchWindow apw;

pubsearchWindow psw;

graphWindow gw;

biblioWindow bw;

public:

Widget(QWidget \*parent = 0);

~Widget();

public slots:

void addWindowActivate();

void deleteWindowActivate();

void nalWindowActivate();

void apsearchWindowActivate();

void pubsearchWindowActivate();

void graphWindowActivate();

void biblioWindowActivate();

};

#endif // WIDGET\_H

**file.h**

#ifndef FILE\_H

#define **FILE\_H**

#include <QFile>

struct **book**

{

QString **auth**;

QString **name**;

QString **pub**;

int **year**;

int **cnum**;

};

class **BookFile**

{

public:

QFile \***library**;

**BookFile**();

~**BookFile**();

};

#endif // FILE\_H

**addwindow.cpp**

#include "addWindow.h"

#include "file.h"

addWindow::**addWindow**(QWidget \***parent**): QWidget(*parent*)

{

author=new QLabel("Author:", this);

name=new QLabel("Name:", this);

publishment=new QLabel("Publishment:", this);

year=new QLabel("Year:", this);

copynum=new QLabel("Number of copies:", this);

Tauthor=new QLineEdit(this);

Tname=new QLineEdit(this);

Tpublishment=new QLineEdit(this);

Tyear=new QLineEdit(this);

Tcopynum=new QLineEdit(this);

add=new QPushButton("Add book", this);

end=new QPushButton("Exit", this);

mainlay=new QVBoxLayout(this);

Labellay=new QVBoxLayout;

Textlay=new QVBoxLayout;

Editlay=new QHBoxLayout;

btnlay=new QHBoxLayout;

Labellay->addWidget(*author*);

Textlay->addWidget(*Tauthor*);

Labellay->addWidget(*name*);

Textlay->addWidget(*Tname*);

Labellay->addWidget(*publishment*);

Textlay->addWidget(*Tpublishment*);

Labellay->addWidget(*year*);

Textlay->addWidget(*Tyear*);

Labellay->addWidget(*copynum*);

Textlay->addWidget(*Tcopynum*);

btnlay->addWidget(*add*);

btnlay->addWidget(*end*);

Editlay->addLayout(*Labellay*);

Editlay->addLayout(*Textlay*);

mainlay->addLayout(*Editlay*);

mainlay->addLayout(*btnlay*);

connect(add,SIGNAL(clicked()),this,SLOT(addbook()));

connect(end,SIGNAL(clicked()),this,SLOT(close()));

}

addWindow::~**addWindow**()

{

}

void addWindow::**addbook**()

{

book **b**;

QString **yr**, **num**;

b.auth=Tauthor->text();

b.name=Tname->text();

b.pub=Tpublishment->text();

yr=Tyear->text();

b.year=yr.toInt();

num=Tcopynum->text();

b.cnum=num.toInt();

F.library->*seek*(F.library->*size*());

QDataStream **out**(*F*.*library*);

out<<b.auth<<b.name<<b.pub<<b.year<<b.cnum;

Tauthor->clear();

Tname->clear();

Tpublishment->clear();

Tyear->clear();

Tcopynum->clear();

}

**apsearchwindow.cpp**

#include "apsearchWindow.h"

#include "file.h"

apsearchWindow::**apsearchWindow**(QWidget \***parent**): QWidget(*parent*)

{

L1=new QLabel("Period: from ", this);

L2=new QLabel(" till ", this);

L3=new QLabel(" year.", this);

Llist=new QLabel("Authors list:", this);

LBord=new QLineEdit(this);

RBord=new QLineEdit(this);

search=new QPushButton("Search", this);

end=new QPushButton("Exit", this);

list=new QTextEdit(this);

mainlay=new QVBoxLayout(this);

Labellay=new QVBoxLayout;

btnlay=new QVBoxLayout;

bordlay=new QHBoxLayout;

actlay=new QHBoxLayout;

bordlay->addWidget(*L1*);

bordlay->addWidget(*LBord*);

bordlay->addWidget(*L2*);

bordlay->addWidget(*RBord*);

bordlay->addWidget(*L3*);

Labellay->addLayout(*bordlay*);

Labellay->addWidget(*Llist*);

btnlay->addWidget(*search*);

btnlay->addWidget(*end*);

actlay->addLayout(*Labellay*);

actlay->addLayout(*btnlay*);

mainlay->addLayout(*actlay*);

mainlay->addWidget(*list*);

list->setReadOnly(true);

connect(end,SIGNAL(clicked()),this,SLOT(close()));

connect(search,SIGNAL(clicked()),this,SLOT(searchfor()));

}

apsearchWindow::~**apsearchWindow**()

{

}

void apsearchWindow::**searchfor**()

{

list->clear();

F.library->*reset*();

QDataStream **in**(*F*.*library*);

book **b**;

int **LB**, **RB**;

LB=(LBord->text()).toInt();

RB=(RBord->text()).toInt();

while(!in.atEnd()){

in>>b.auth>>b.name>>b.pub>>b.year>>b.cnum;

if(b.year>=LB && b.year<=RB) list->append(b.auth);

}

}

**bibliowindow.cpp**

#include "widget.h"

biblioWindow::biblioWindow(QWidget \*parent): QWidget(parent)

{

Table=new QTableWidget(5,5,this);

end=new QPushButton("Exit", this);

mainlay=new QHBoxLayout(this);

mainlay->addWidget(Table);

mainlay->addWidget(end);

QStringList line;

line<<"Author"<<"Name"<<"Publishment"<<"Year"<<"Copy number";

Table->setHorizontalHeaderLabels(line);

connect(end,SIGNAL(clicked()),this,SLOT(close()));

}

biblioWindow::~biblioWindow()

{

}

void biblioWindow::Tshow()

{

F.library->reset();

QDataStream in(F.library);

book b;

int i=0;

while(!in.atEnd()){

in>>b.auth>>b.name>>b.pub>>b.year>>b.cnum;

QTableWidgetItem \*author=new QTableWidgetItem;

author->setFlags(Qt::NoItemFlags);

author->setText(b.auth);

Table->setItem(i, 0, author);

QTableWidgetItem \*name=new QTableWidgetItem(b.name);

Table->setItem(i, 1, name);

QTableWidgetItem \*pub=new QTableWidgetItem(b.pub);

Table->setItem(i, 2, pub);

QString yr;

yr.setNum(b.year);

QTableWidgetItem \*year=new QTableWidgetItem(yr);

Table->setItem(i, 3, year);

QString nc;

nc.setNum(b.cnum);

QTableWidgetItem \*copynum=new QTableWidgetItem(nc);

Table->setItem(i, 4, copynum);

i++;

Table->setRowCount(Table->rowCount()+1);

}

show();

}

**deletewindow.cpp**

#include "deleteWindow.h"

#include "file.h"

deleteWindow::**deleteWindow**(QWidget \***parent**): QWidget(*parent*)

{

name=new QLabel("Name:", this);

Tname=new QLineEdit(this);

del=new QPushButton("Delete book", this);

end=new QPushButton("Exit", this);

delAll=new QPushButton("Delete all", this);

mainlay=new QVBoxLayout(this);

namelay=new QHBoxLayout;

btnlay=new QHBoxLayout;

namelay->addWidget(*name*);

namelay->addWidget(*Tname*);

btnlay->addWidget(*del*);

btnlay->addWidget(*end*);

mainlay->addLayout(*namelay*);

mainlay->addLayout(*btnlay*);

mainlay->addWidget(*delAll*);

connect(delAll,SIGNAL(clicked()),this,SLOT(delall()));

connect(end,SIGNAL(clicked()),this,SLOT(close()));

}

**graphwindow.cpp**

#include "graphWindow.h"

#include "widget.h"

#include "file.h"

graphWindow::graphWindow(QWidget \*parent): QMainWindow(parent)

{

paint=new QPushButton("Paint", this);

setCentralWidget(paint);

connect(paint, SIGNAL(clicked()),this,SLOT(activate()));

resize(800,400);

}

graphWindow::~graphWindow()

{

}

void graphWindow::activate()

{

QBarSet \*soldcopies = new QBarSet("Copy number");

QDataStream in(F.library);

QStringList categories;

book b;

QString yr;

int minyear, maxyear, maxcopy=0;

F.library->reset();

in>>b.auth>>b.name>>b.pub>>b.year>>b.cnum;

minyear=b.year;

maxyear=b.year;

while(!in.atEnd()){

in>>b.auth>>b.name>>b.pub>>b.year>>b.cnum;

if(b.year>maxyear) maxyear=b.year;

if(b.year<minyear) minyear=b.year;

}

int a[maxyear-minyear+1];

for(int i=0;i<maxyear-minyear+1;i++) a[i]=0;

F.library->reset();

while(!in.atEnd()){

in>>b.auth>>b.name>>b.pub>>b.year>>b.cnum;

a[b.year-minyear]=a[b.year-minyear]+b.cnum;

}

for(int i=0;i<maxyear-minyear+1;i++) if(a[i]>maxcopy) maxcopy=a[i];

for(int i=0;i<maxyear-minyear+1;i++) {

yr.setNum(minyear+i);

\*soldcopies<<a[i];

categories << yr;

}

QBarSeries \*series = new QBarSeries();

series->append(soldcopies);

QChart \*chart = new QChart();

chart->addSeries(series);

chart->setTitle("Number of sold copies per year");

chart->setAnimationOptions(QChart::SeriesAnimations);

QBarCategoryAxis \*axisX = new QBarCategoryAxis();

axisX->append(categories);

chart->addAxis(axisX, Qt::AlignBottom);

series->attachAxis(axisX);

QValueAxis \*axisY = new QValueAxis();

axisY->setRange(0,maxcopy);

chart->addAxis(axisY, Qt::AlignLeft);

series->attachAxis(axisY);

QChartView \*chartView = new QChartView(chart);

chartView->setRenderHint(QPainter::Antialiasing);

setCentralWidget(chartView);

resize(800, 400);

}

**nalwindow.cpp**

#include "nalWindow.h"

nalWindow::**nalWindow**(QWidget \***parent**): QWidget(*parent*)

{

author=new QLabel("Author:", this);

name=new QLabel("Name:", this);

publishment=new QLabel("Publishment:", this);

year=new QLabel("Year:", this);

Tauthor=new QLineEdit(this);

Tname=new QLineEdit(this);

Tpublishment=new QLineEdit(this);

Tyear=new QLineEdit(this);

back=new QPushButton("Back", this);

next=new QPushButton("Next", this);

end=new QPushButton("Exit", this);

mainlay=new QVBoxLayout(this);

Labellay=new QVBoxLayout;

Textlay=new QVBoxLayout;

Editlay=new QHBoxLayout;

btnlay=new QHBoxLayout;

Tauthor->setReadOnly(true);

Tname->setReadOnly(true);

Tpublishment->setReadOnly(true);

Tyear->setReadOnly(true);

Labellay->addWidget(*author*);

Textlay->addWidget(*Tauthor*);

Labellay->addWidget(*name*);

Textlay->addWidget(*Tname*);

Labellay->addWidget(*publishment*);

Textlay->addWidget(*Tpublishment*);

Labellay->addWidget(*year*);

Textlay->addWidget(*Tyear*);

btnlay->addWidget(*back*);

btnlay->addWidget(*next*);

Editlay->addLayout(*Labellay*);

Editlay->addLayout(*Textlay*);

mainlay->addLayout(*Editlay*);

mainlay->addLayout(*btnlay*);

mainlay->addWidget(*end*);

connect(end,SIGNAL(clicked()),this,SLOT(close()));

connect(next,SIGNAL(clicked()),this,SLOT(nextbutton()));

connect(back,SIGNAL(clicked()),this,SLOT(prevbutton()));

QDataStream **in**(*F*.*library*);

int **i**=0;

while(!in.atEnd()){

in>>b.auth>>b.name>>b.pub>>b.year>>b.cnum;

if(b.cnum==1){mas[i]=b;i++;}

}

F.library->*reset*();

Tauthor->setText(mas[0].auth);

Tname->setText(mas[0].name);

Tpublishment->setText(mas[0].pub);

QString **yr**;

yr.setNum(mas[0].year);

Tyear->setText(yr);

j=0;

}

nalWindow::~**nalWindow**()

{

}

void nalWindow::**nextbutton**()

{

if(mas[j+1].auth!="") {

j++;

Tauthor->setText(mas[j].auth);

Tname->setText(mas[j].name);

Tpublishment->setText(mas[j].pub);

QString **yr**;

yr.setNum(mas[j].year);

Tyear->setText(yr);

}

}

void nalWindow::**prevbutton**()

{

if(j>0) {

j--;

Tauthor->setText(mas[j].auth);

Tname->setText(mas[j].name);

Tpublishment->setText(mas[j].pub);

QString **yr**;

yr.setNum(mas[j].year);

Tyear->setText(yr);

}

}

**pubsearchwindow.cpp**

#include "pubsearchWindow.h"

pubsearchWindow::**pubsearchWindow**(QWidget \***parent**): QWidget(*parent*)

{

pub=new QLabel("Publisher: ", this);

list=new QLabel("Book list:", this);

Tpub=new QLineEdit(this);

search=new QPushButton("Search", this);

end=new QPushButton("Exit", this);

Tlist=new QTextEdit(this);

mainlay=new QHBoxLayout(this);

Labellay=new QVBoxLayout;

btnlay=new QVBoxLayout;

bordlay=new QHBoxLayout;

actlay=new QVBoxLayout;

bordlay->addWidget(*pub*);

bordlay->addWidget(*Tpub*);

Labellay->addWidget(*list*);

Labellay->addWidget(*Tlist*);

actlay->addLayout(*bordlay*);

actlay->addLayout(*Labellay*);

btnlay->addWidget(*search*);

btnlay->addWidget(*end*);

mainlay->addLayout(*actlay*);

mainlay->addLayout(*btnlay*);

Tlist->setReadOnly(true);

connect(end,SIGNAL(clicked()),this,SLOT(close()));

connect(search,SIGNAL(clicked()),this,SLOT(searchfor()));

}

pubsearchWindow::~**pubsearchWindow**()

{

}

void pubsearchWindow::**searchfor**()

{

Tlist->clear();

F.library->*reset*();

QDataStream **in**(*F*.*library*);

book **b**;

while(!in.atEnd()){

in>>b.auth>>b.name>>b.pub>>b.year>>b.cnum;

if(Tpub->text()==b.pub) {

QString **info**;

QString **yr**;

yr.setNum(b.year);

info=b.name+", "+yr;

Tlist->append(info);

}

}

}

**widget.cpp**

#include "widget.h"

#include "area.h"

#include "stdlib.h"

#include "file.h"

#include <iostream>

#include <fstream>

#include <string>

BookFile F;

Widget::Widget(QWidget \*parent)

: QWidget(parent)

{

welcome=new QLabel("Welcome to the library!", this);

add=new QPushButton("Add book", this);

del=new QPushButton("Delete book", this) ;

nal=new QPushButton("Availiability", this) ;

authper=new QPushButton("Sort authors by period", this) ;

izd=new QPushButton("Sort by publisher", this) ;

graph=new QPushButton("Publication graph", this) ;

biblio=new QPushButton("Library", this) ;

end=new QPushButton("Exit", this) ;

btnlay=new QVBoxLayout;

endlay=new QVBoxLayout;

mainlay=new QHBoxLayout(this);

btnlay->addWidget(add);

btnlay->addWidget(del);

btnlay->addWidget(nal);

btnlay->addWidget(authper);

btnlay->addWidget(izd);

btnlay->addWidget(graph);

btnlay->addWidget(biblio);

endlay->addWidget(welcome);

endlay->addWidget(end);

mainlay->addLayout(btnlay);

mainlay->addLayout(endlay);

connect(add,SIGNAL(clicked()),this,SLOT(addWindowActivate()));

connect(del,SIGNAL(clicked()),this,SLOT(deleteWindowActivate()));

connect(nal,SIGNAL(clicked()),this,SLOT(nalWindowActivate()));

connect(authper,SIGNAL(clicked()),this,SLOT(apsearchWindowActivate()));

connect(izd,SIGNAL(clicked()),this,SLOT(pubsearchWindowActivate()));

connect(graph,SIGNAL(clicked()),this,SLOT(graphWindowActivate()));

connect(biblio,SIGNAL(clicked()),this,SLOT(biblioWindowActivate()));

connect(end,SIGNAL(clicked()),this,SLOT(close()));

}

Widget::~Widget()

{

}

void Widget::addWindowActivate()

{

aw.show();

};

void Widget::deleteWindowActivate()

{

dw.show();

};

void Widget::nalWindowActivate()

{

nw.show();

};

void Widget::apsearchWindowActivate()

{

apw.show();

};

void Widget::pubsearchWindowActivate()

{

psw.show();

};

void Widget::graphWindowActivate()

{

gw.show();

};

void Widget::biblioWindowActivate()

{

bw.Tshow();

};

**file.cpp**

#include "file.h"

BookFile::**BookFile**()

{

library=new QFile("library.txt");

library->*open*(QFile::ReadWrite);

}

BookFile::~**BookFile**()

{

library->*close*();

}

**main.cpp**

#include "widget.h"

#include <QApplication>

int main(int **argc**, char \***argv**[])

{

QApplication **a**(*argc*, *argv*);

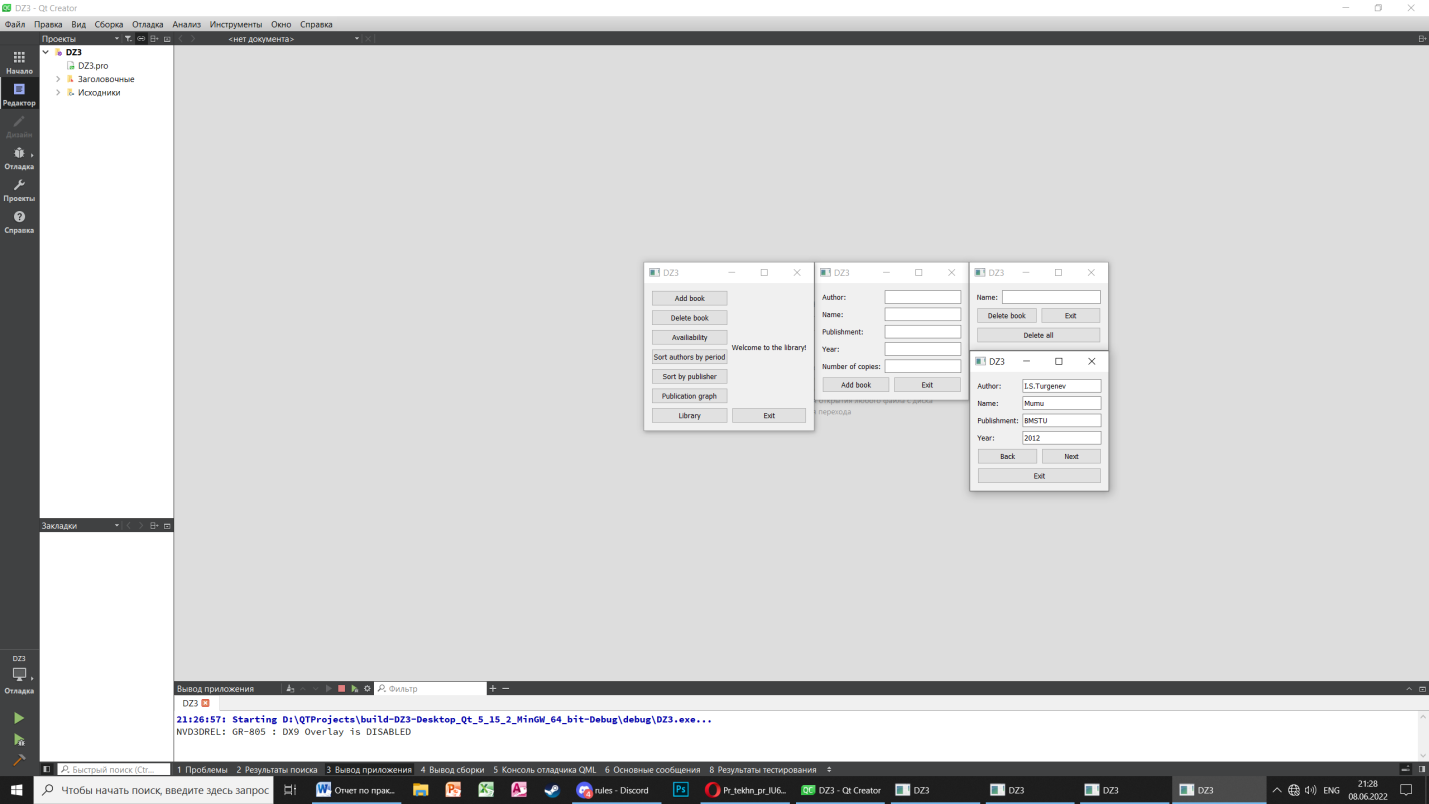
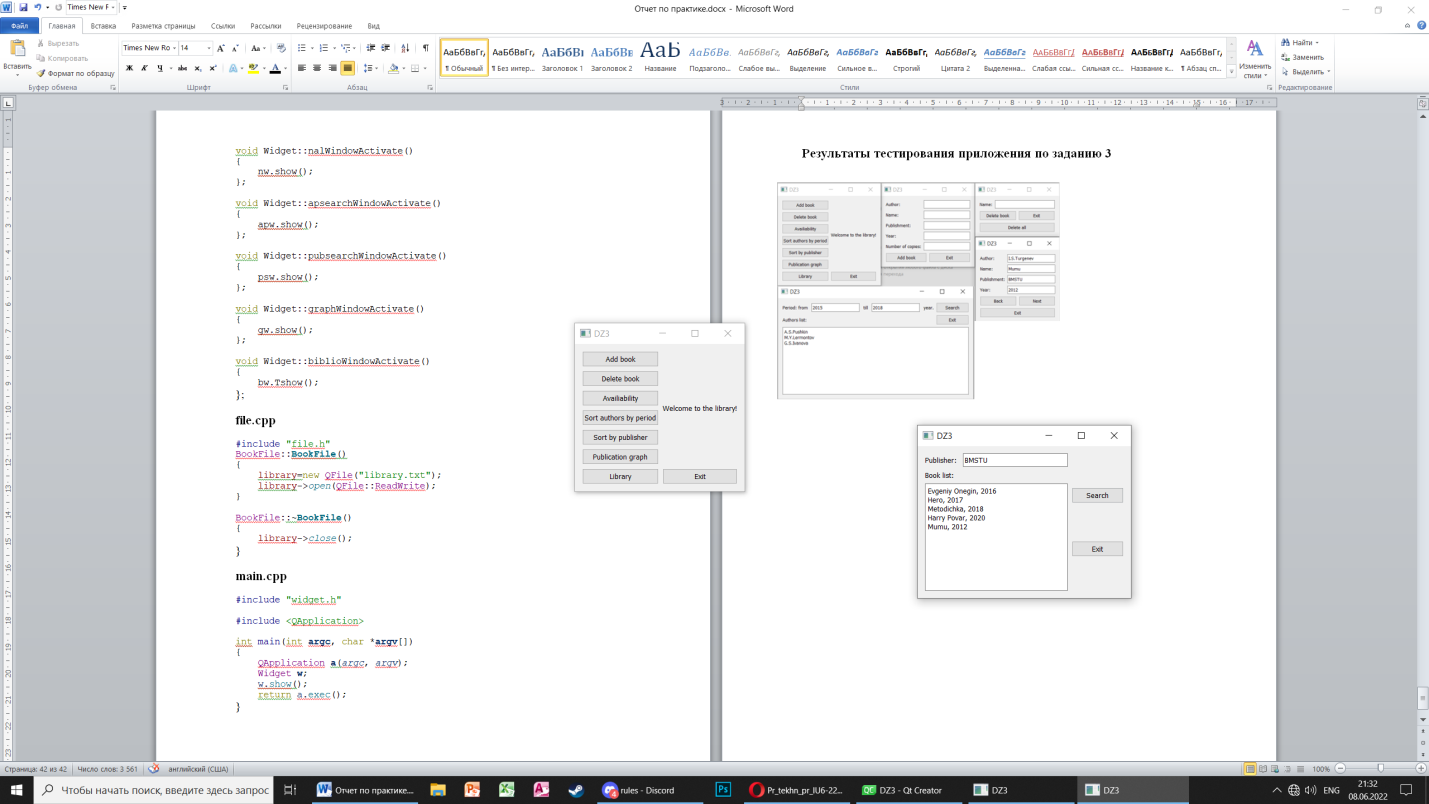
Widget **w**;

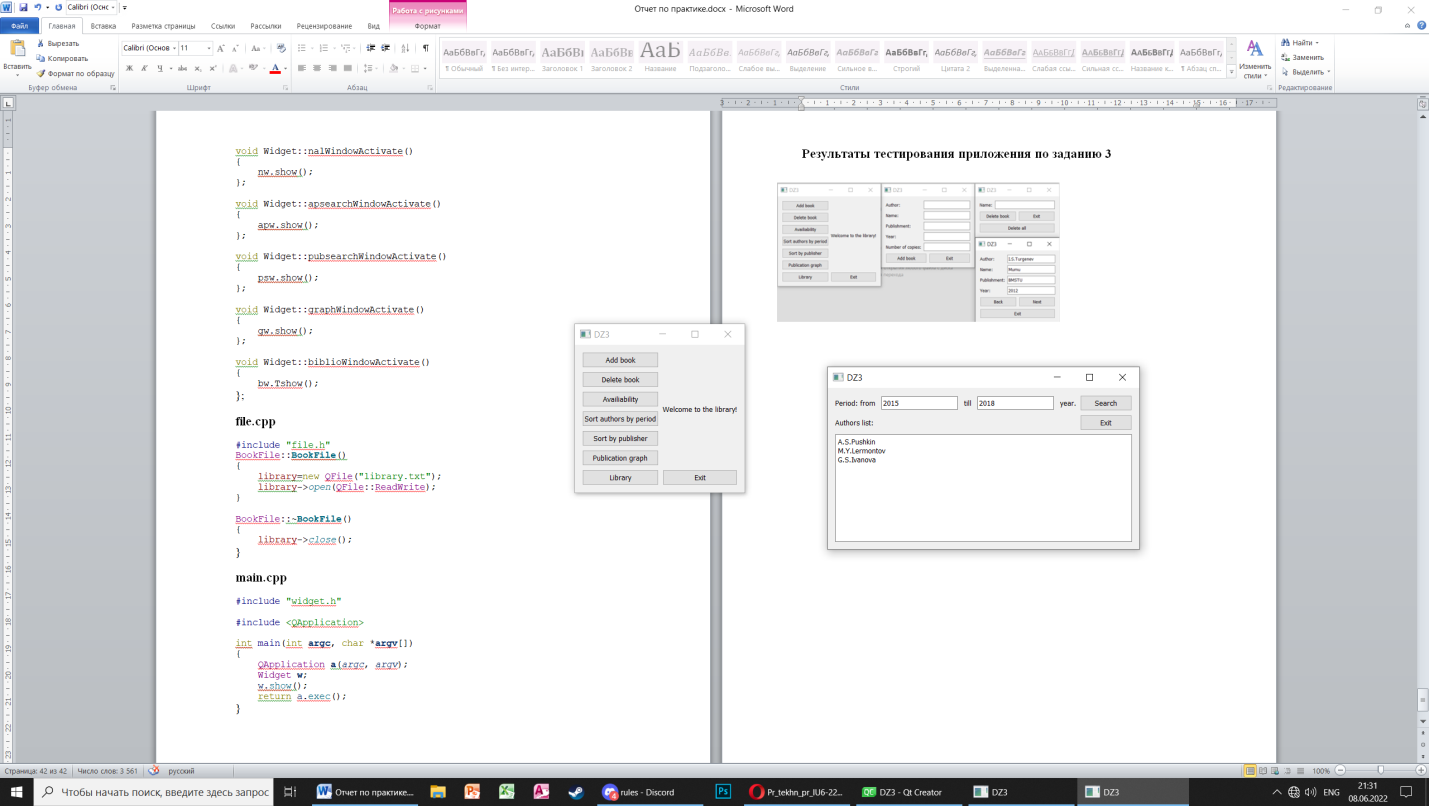
w.show();

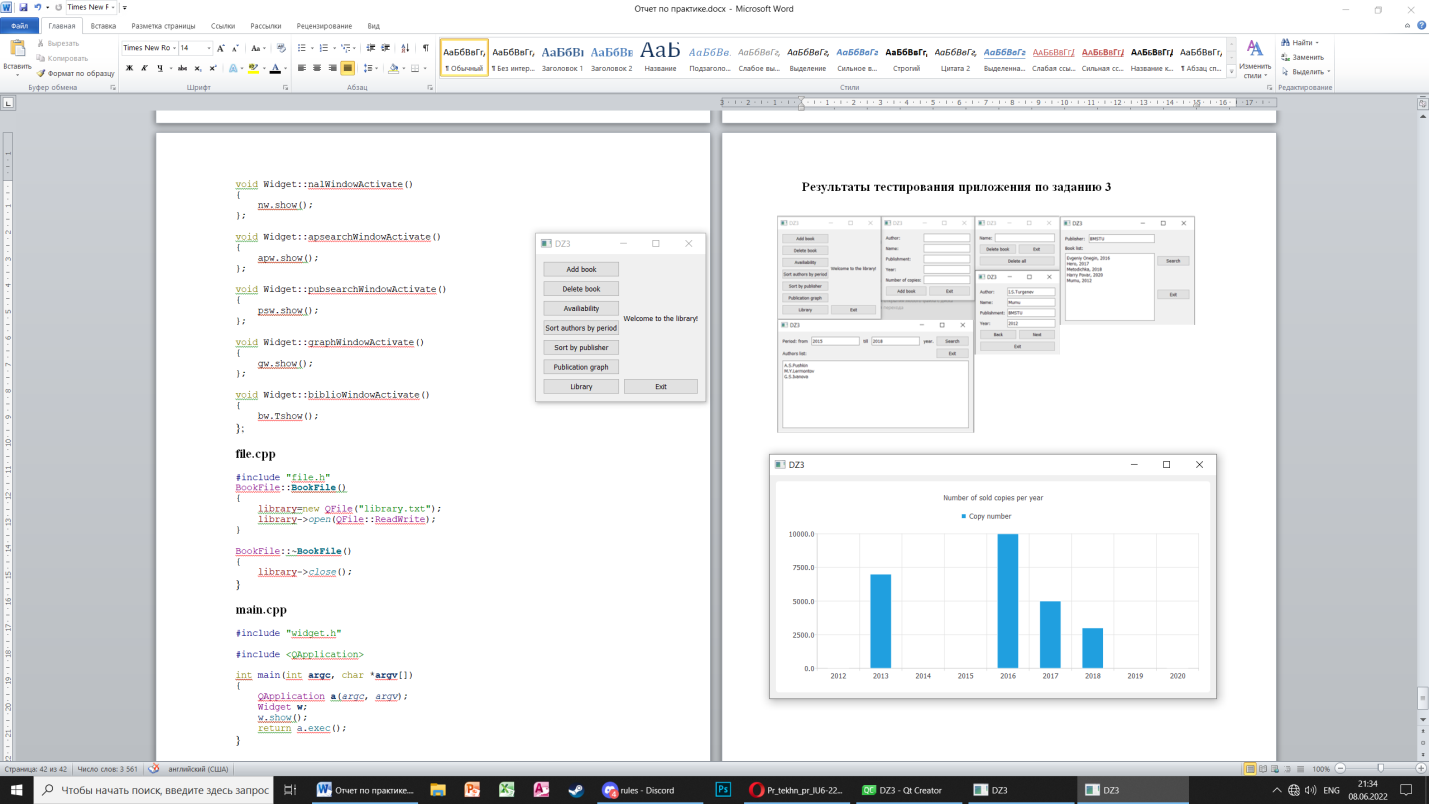
return a.exec();

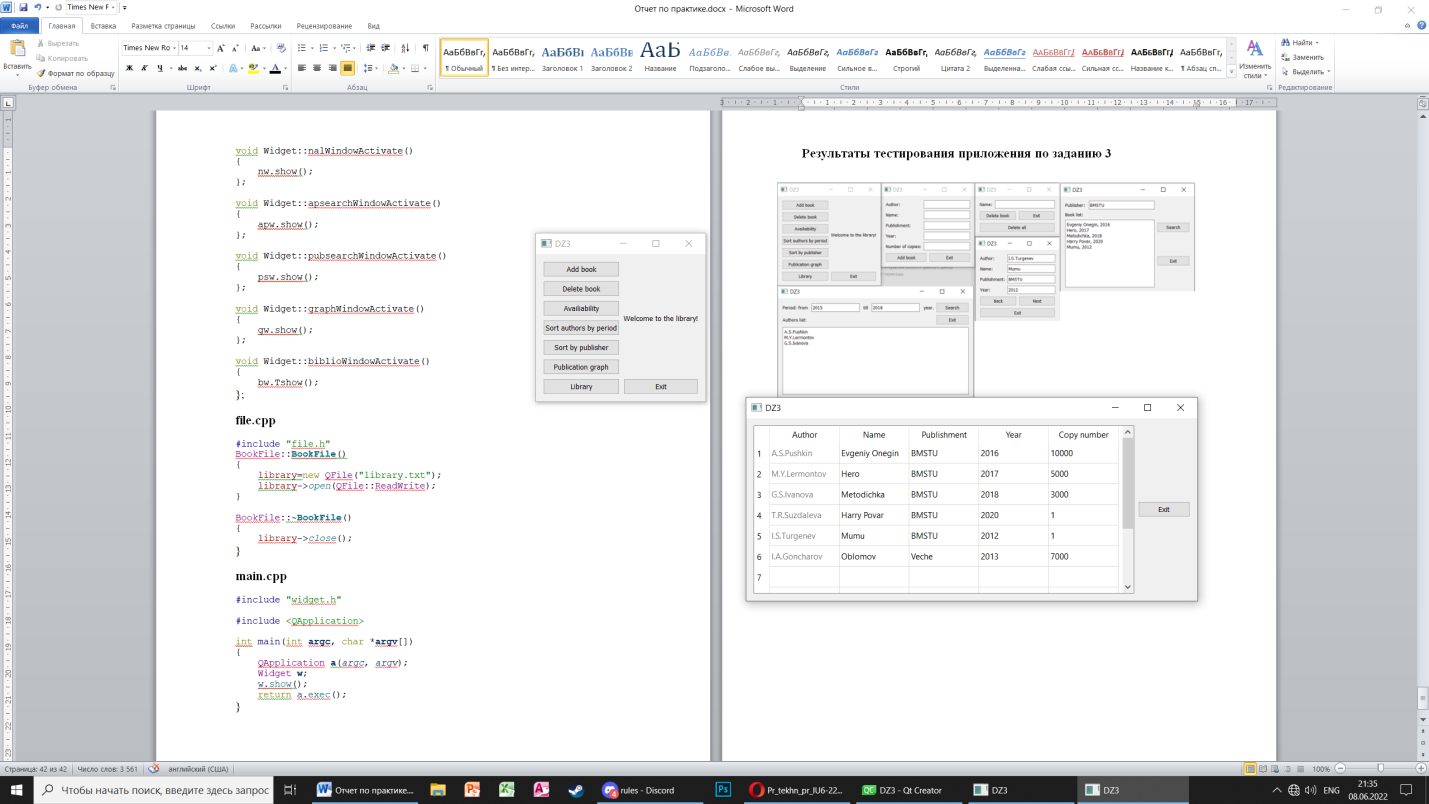
}

**Результаты тестирования приложения по заданию 3**









**Вывод по заданию 3**

В результате работы создана система, позволяющая реализовывать следующие функции:

• Добавление книги

• Удаление книги

• Поиск по параметру

• Построение графика

На заданных исходных данных система работает корректно.

Заключение

При работе над практикумом мною получены навыки создания программных систем с оконными и консольными интерфейсами. Я изучил средства реализации программ на языках Delphi Pascal и C++, а также овладел методикой проектирования небольших программных систем при структурном и объектном подходах.

Список литературы

1. Иванова Г.С., Ничушкина Т.Н., Пугачев Е.К. Объектно-ориентированное программирование. Учеб. для вузов. М.: Изд-во МГТУ им. Н.Э. Баумана, 2007. Гл. 5. С 218-245.

2. Иванова Г.С., Ничушкина Т.Н., Самарев Р.С. . C++. Часть 1.Средства процедурного программирования Microsoft Visual С++ 2008: Учебное пособие. – М.: МГТУ им. Н.Э. Баумана, 2010. 126 с.

3. Иванова Г.С., Ничушкина Т.Н. C++. Часть 2. Объектно-ориентированное программирование на языке С++ в среде Visual Studio 2008: Учебное пособие. М.: МГТУ им. Н.Э. Баумана, 2012. 161 с.

4. Иванова Г.С. . C++. Часть 3. Создание пользовательских интерфейсов в программах на С++ с использованием библиотеки Qt: Учебное пособие. – М.: МГТУ им. Н.Э. Баумана, 2010. 52 с.