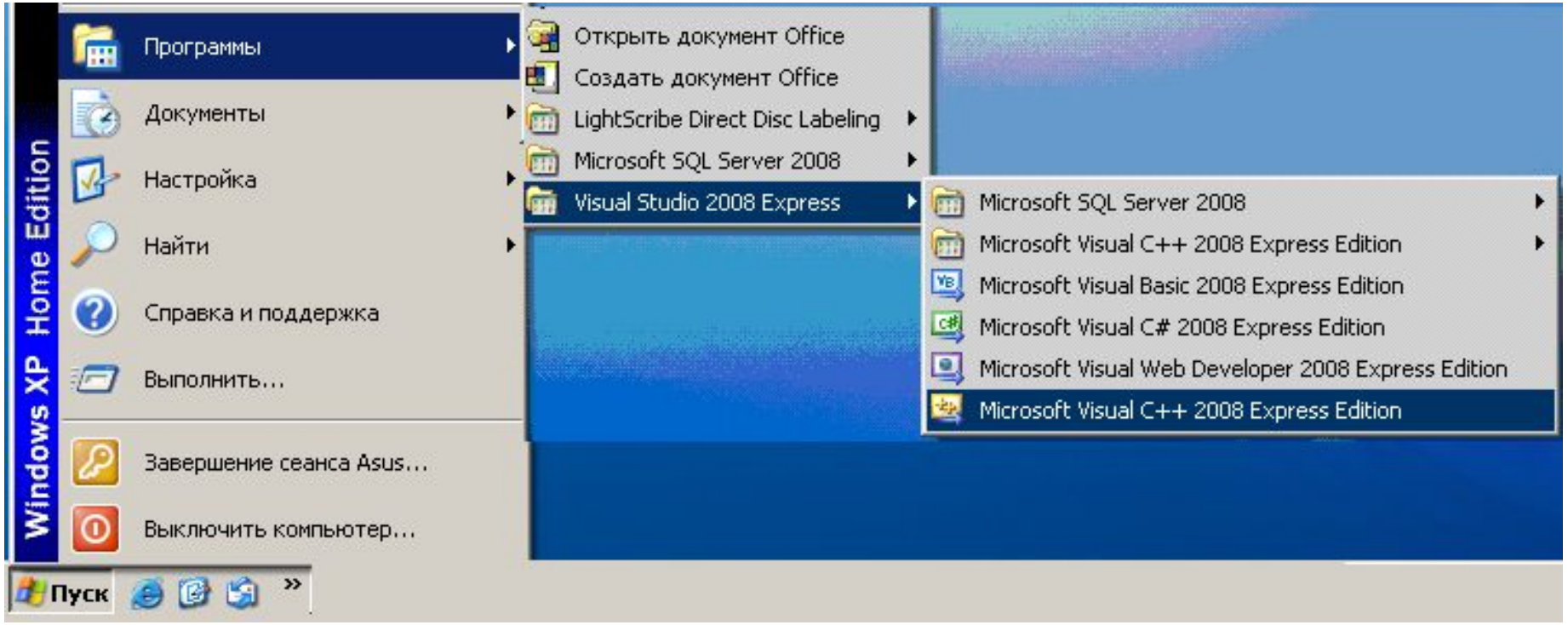
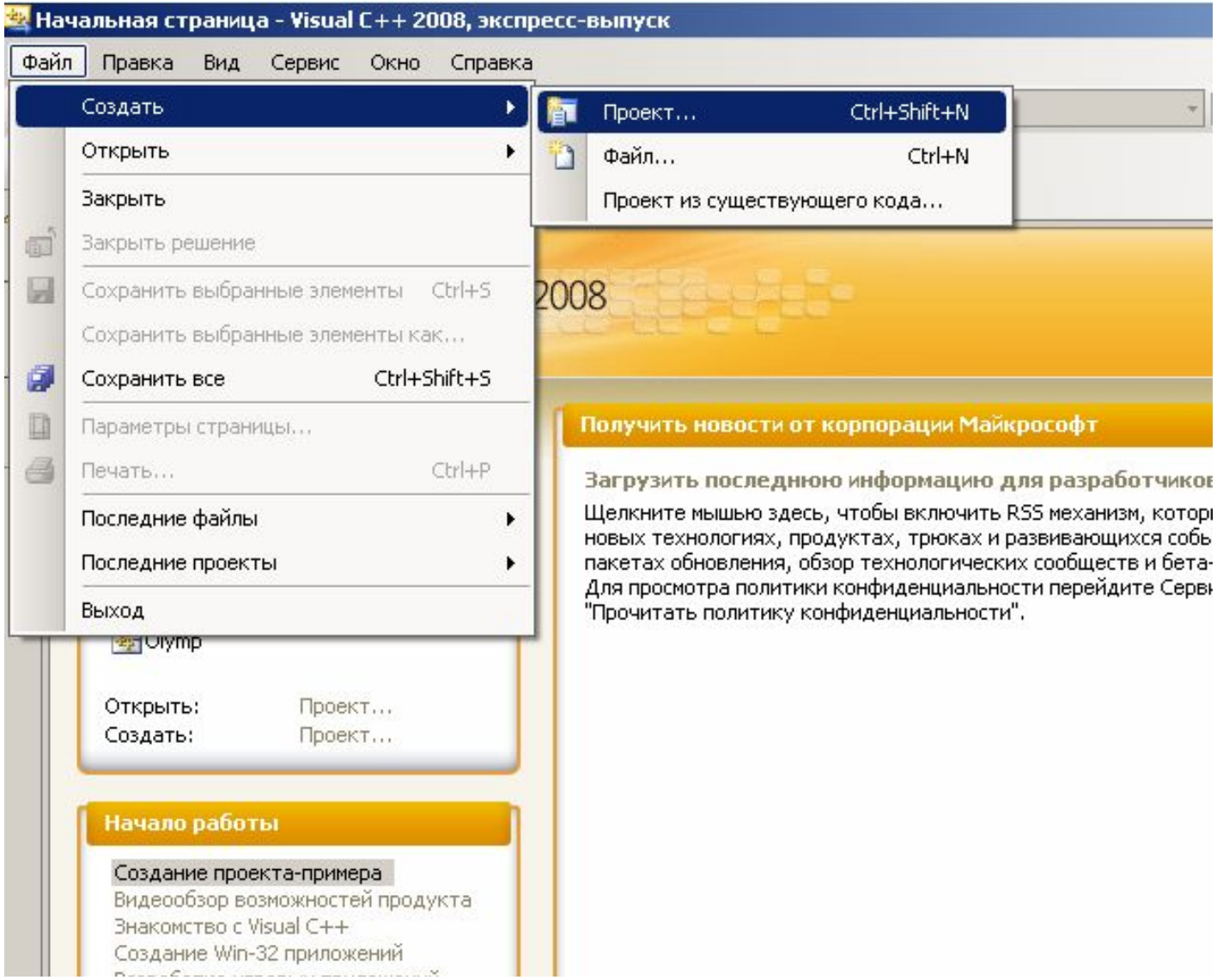
Курсовая работа
в среде
Microsoft Visual Studio 2008 Express

1. Запускаем среду

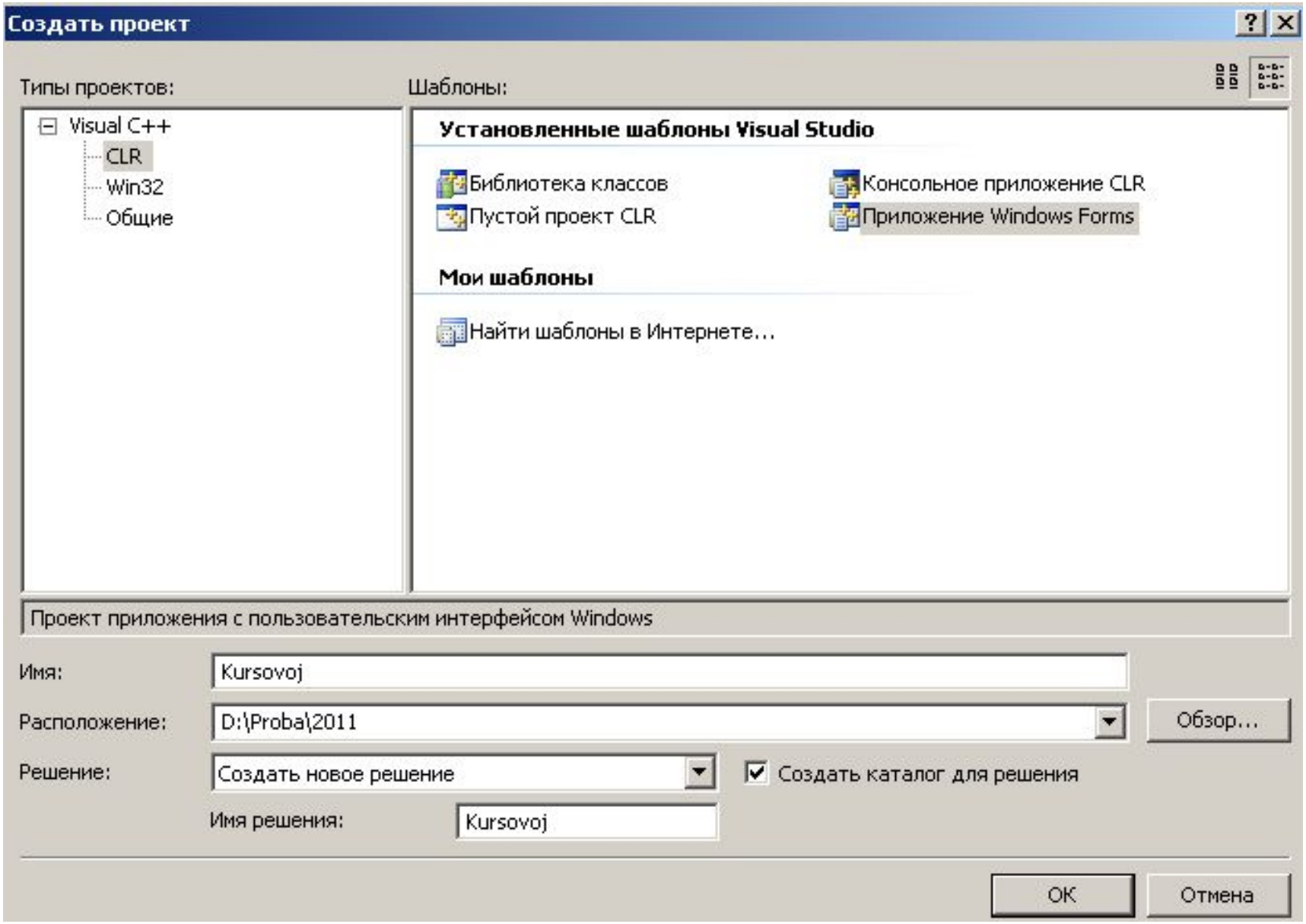
Microsoft Visual C++ Express Edition



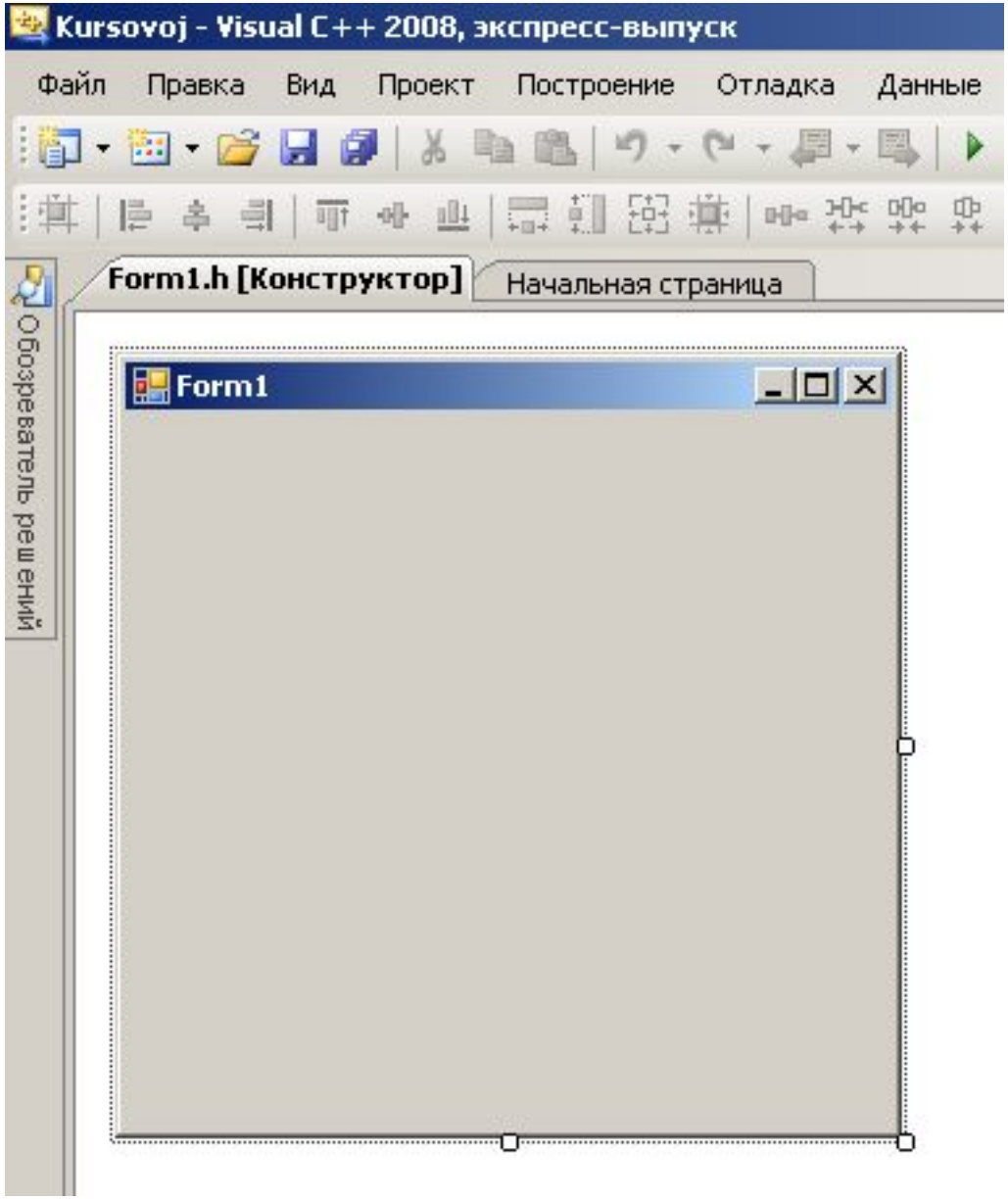
2. Создаем новый проект



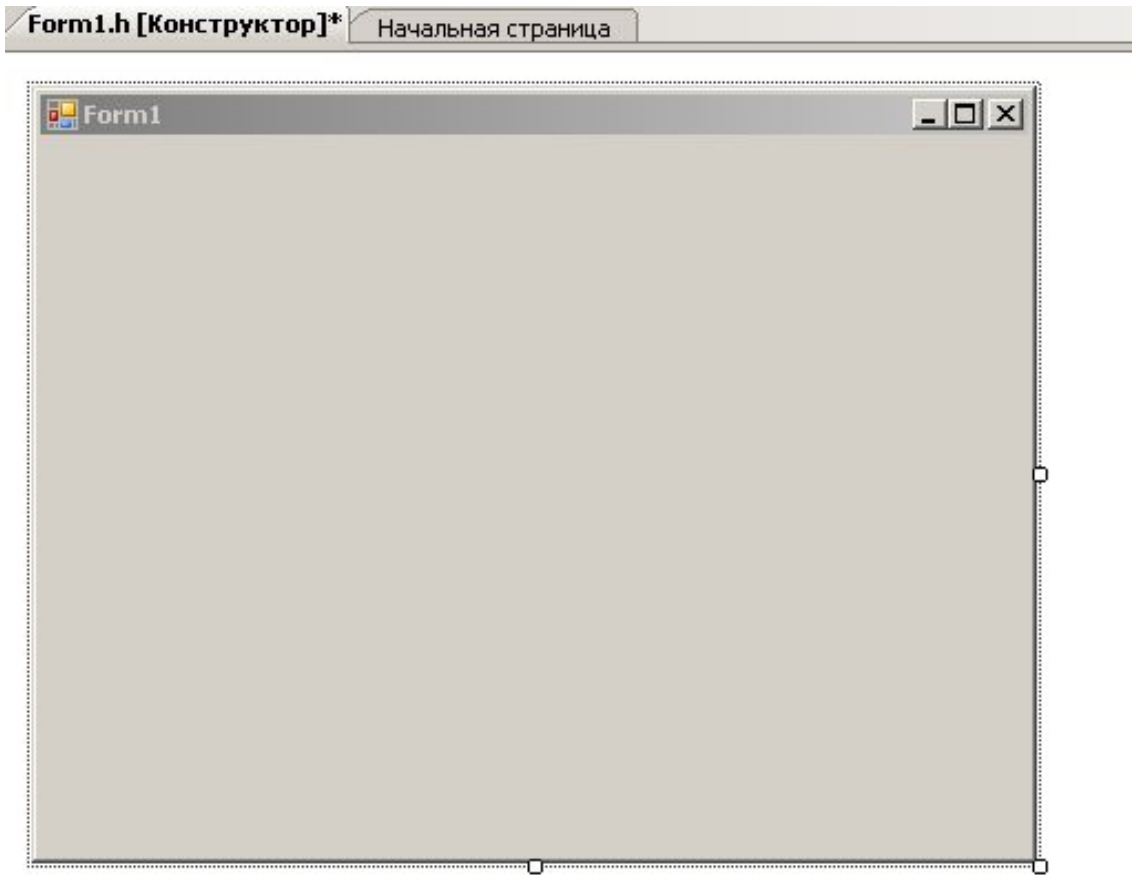
3. Задаем тип (Приложение Windows Forms), название (Graph1) и каталог для размещения проекта. Нажимаем <Ok>

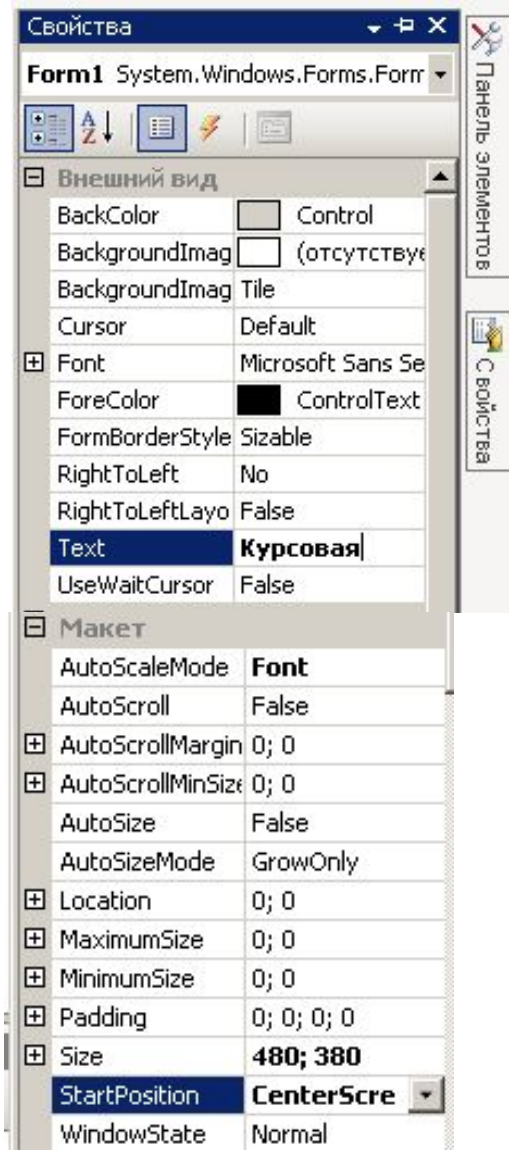


4. Мастер создает заготовку формы

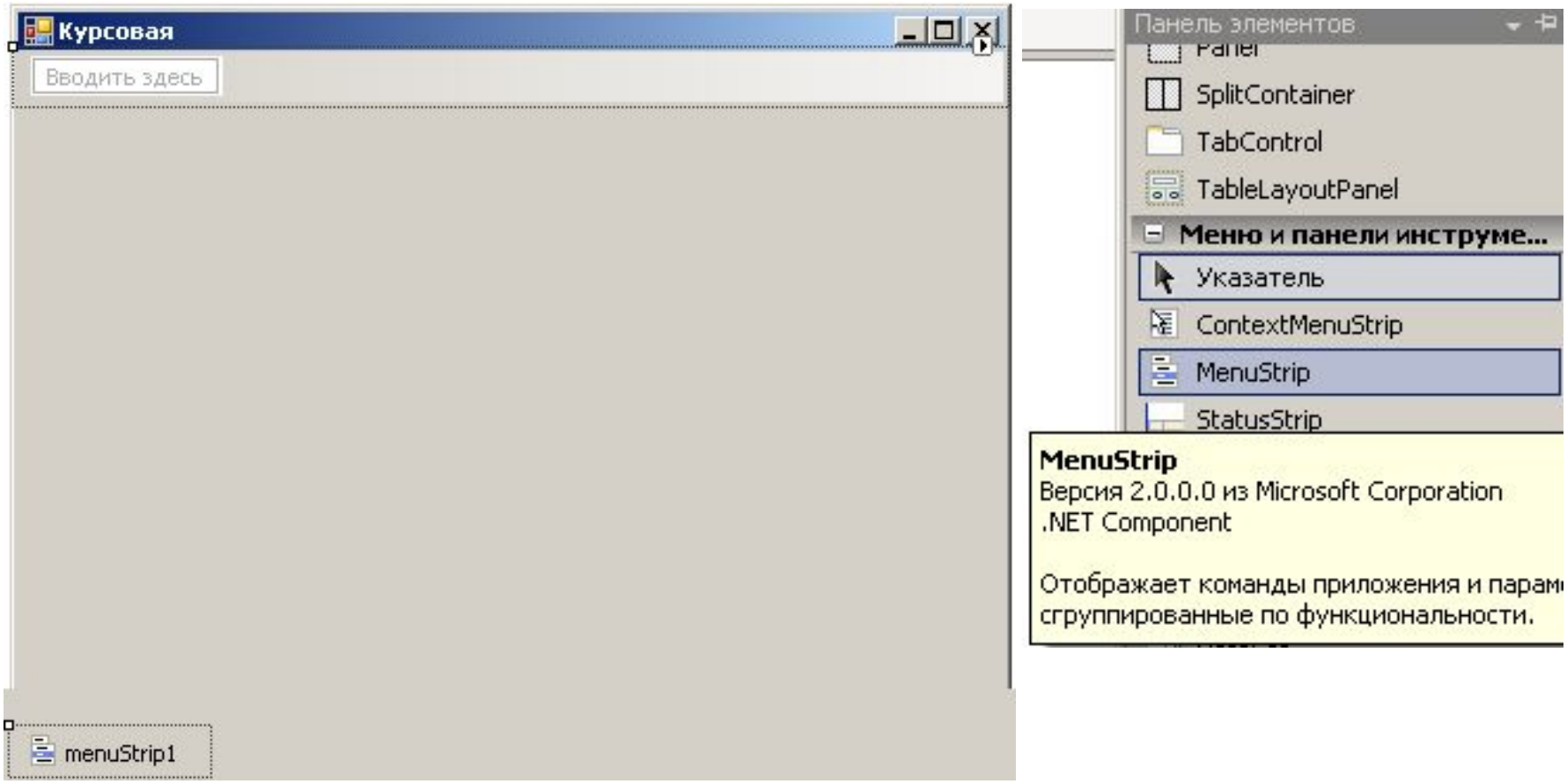


5. Устанавливаем размер и свойства формы

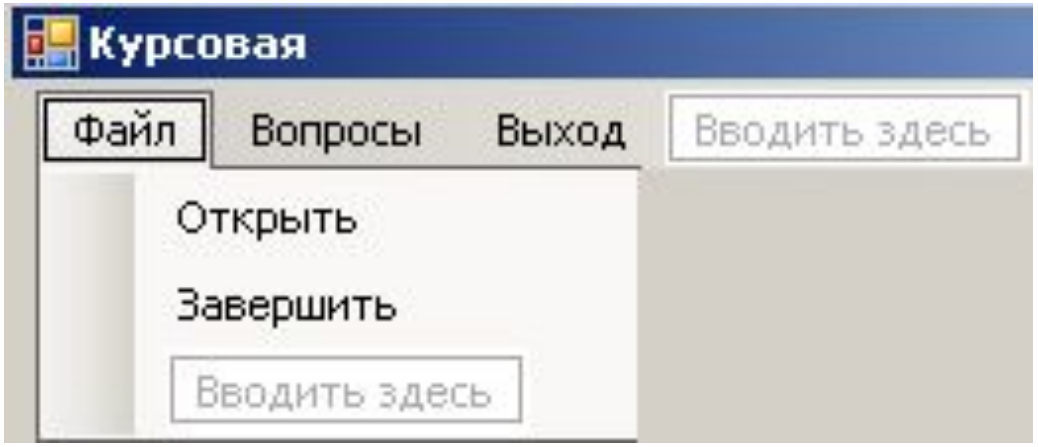


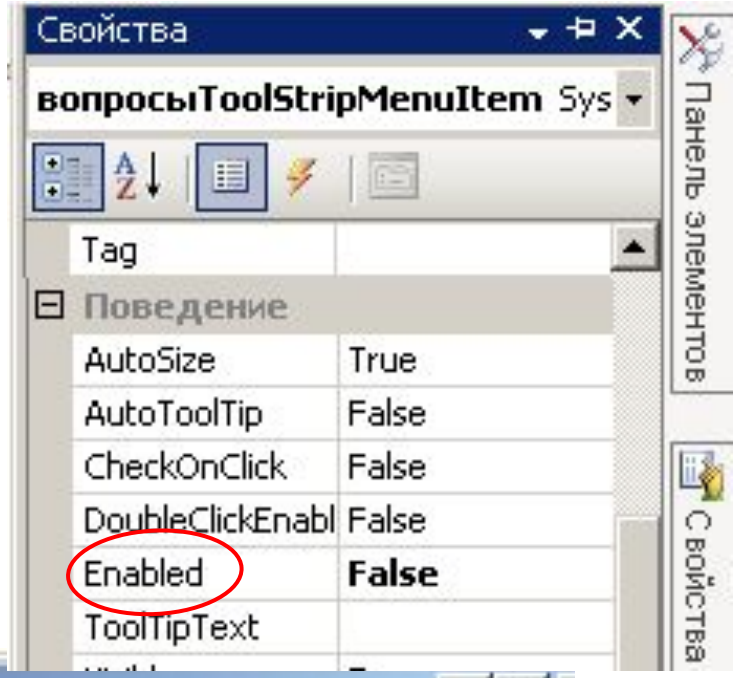


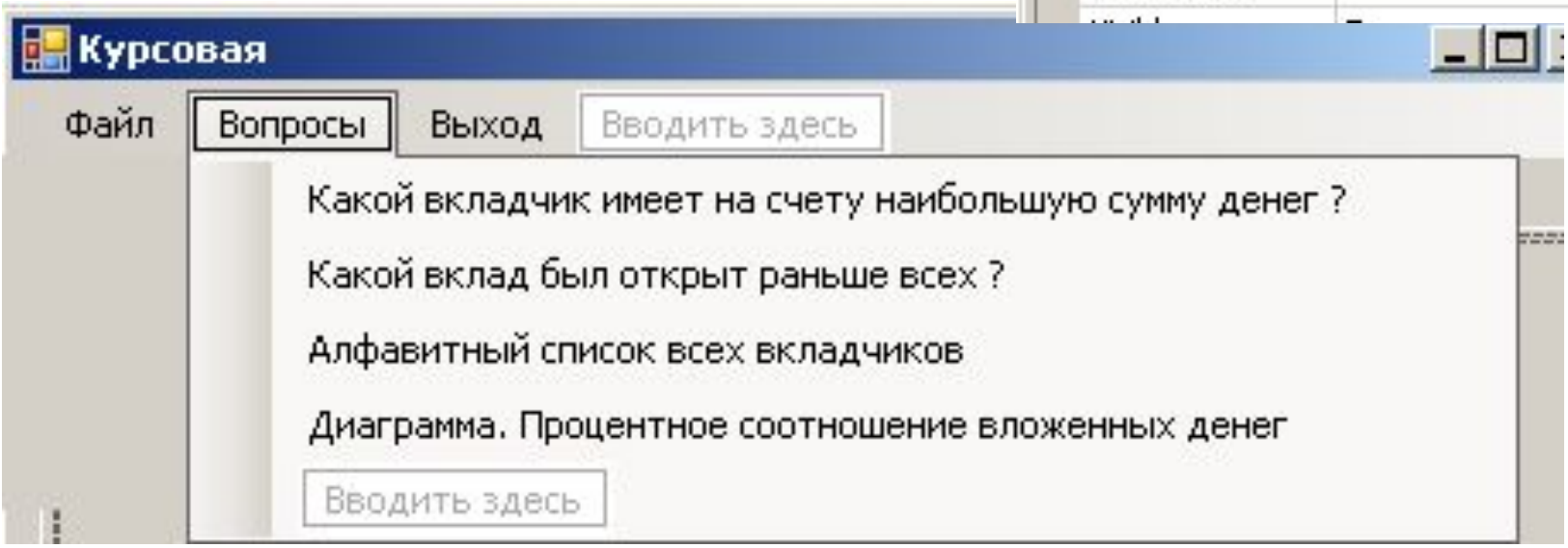
6. Перетаскиваем на форму элемент MenuStrip



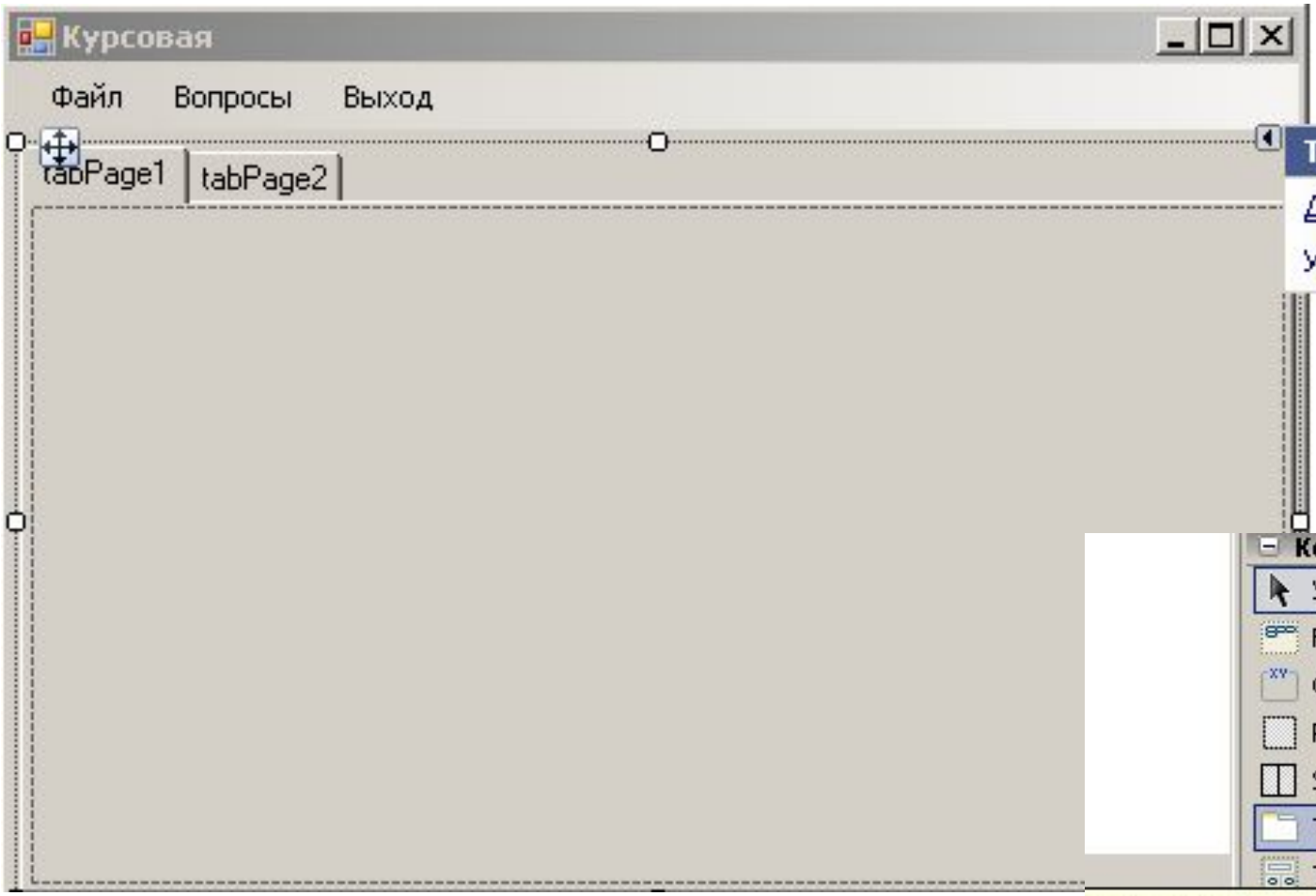
7. Наполняем меню нужными пунктами

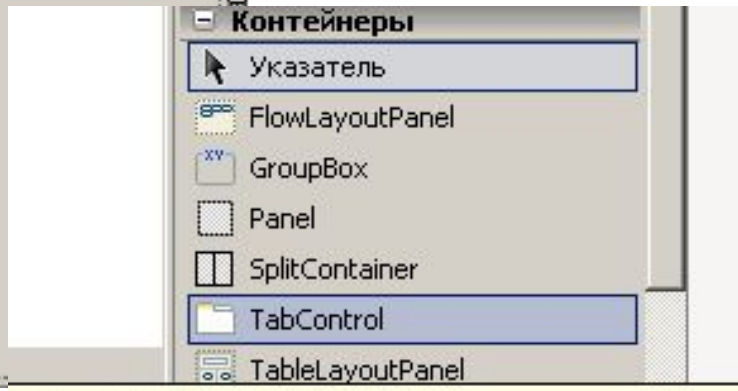






7. Добавляем элемент TabControl



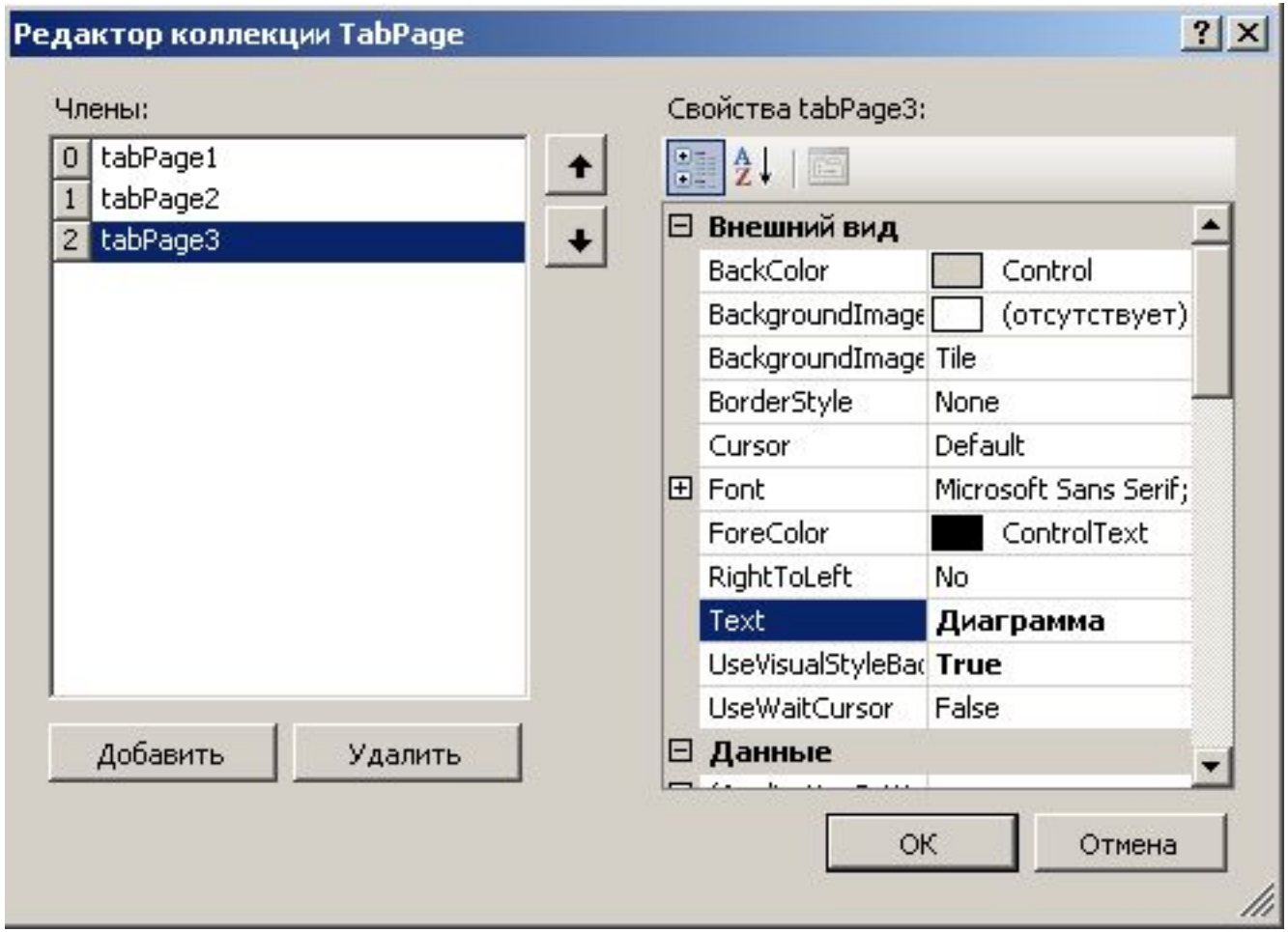


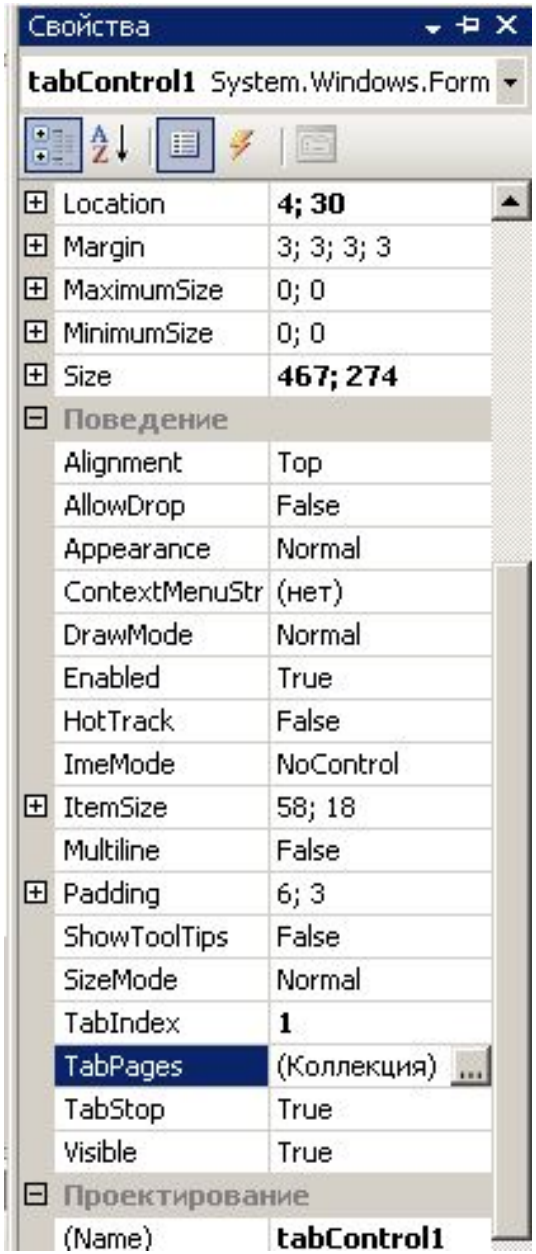
Это один из способов добавления / удаления вкладок

TabControl
Версия 2.0.0.0 из Microsoft Corporation
.NET Component

Обрабатывает и отображает для пользователя соответствующую коллекцию вкладок, содержащих элементы управления и компоненты.

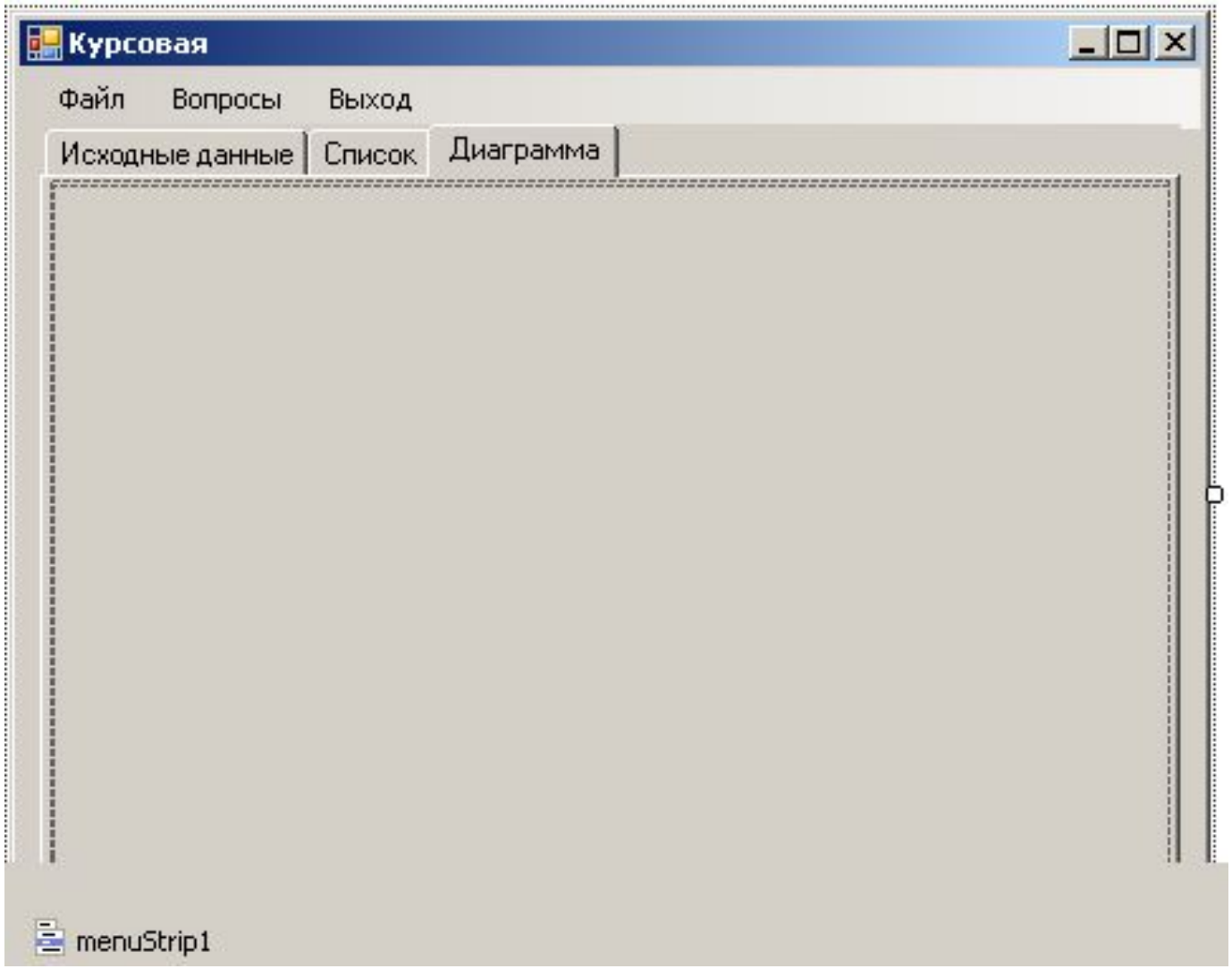
8. Редактируем вкладки TabControl



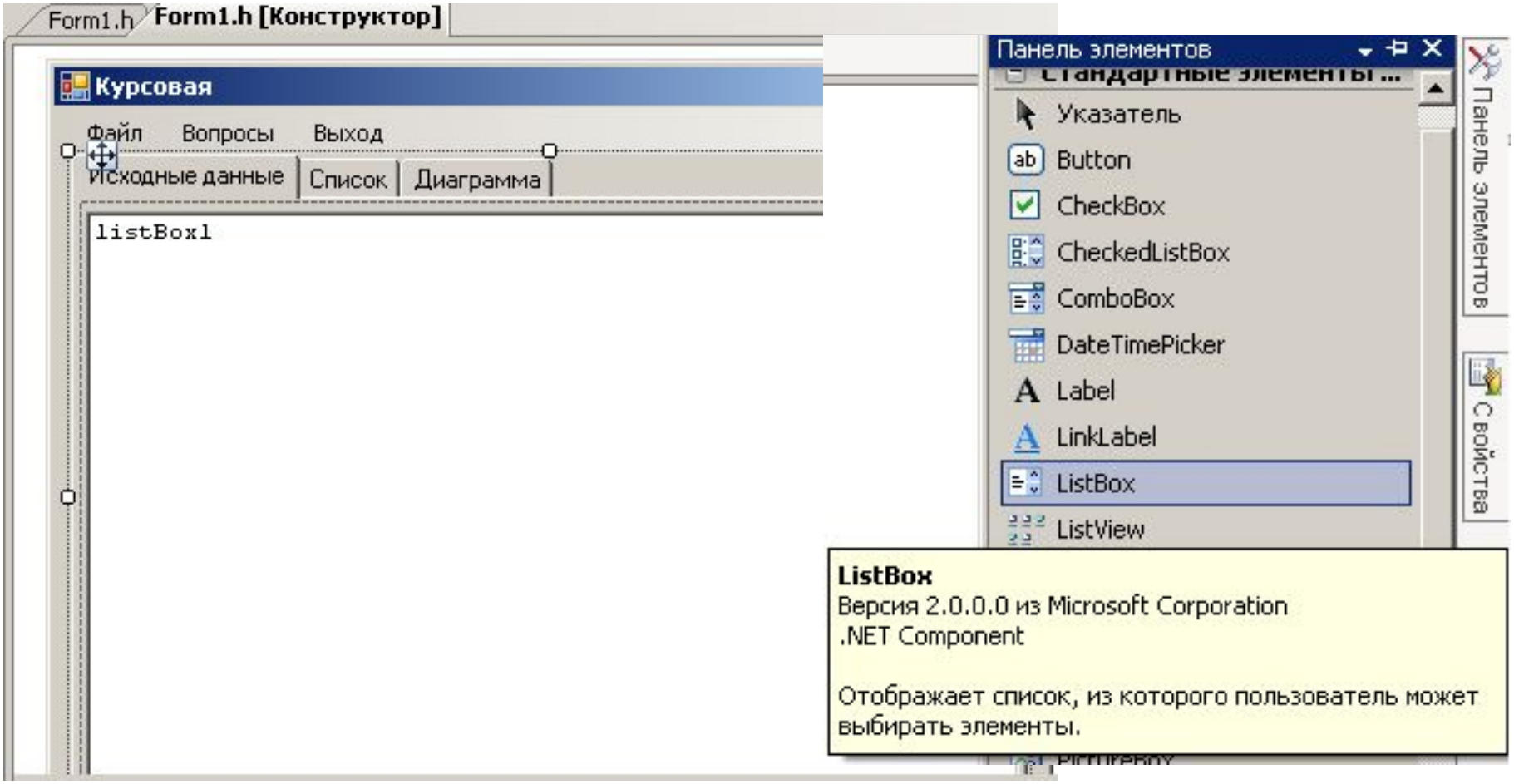


Гораздо больше возможностей обеспечивает свойство TabPages (Коллекция) …

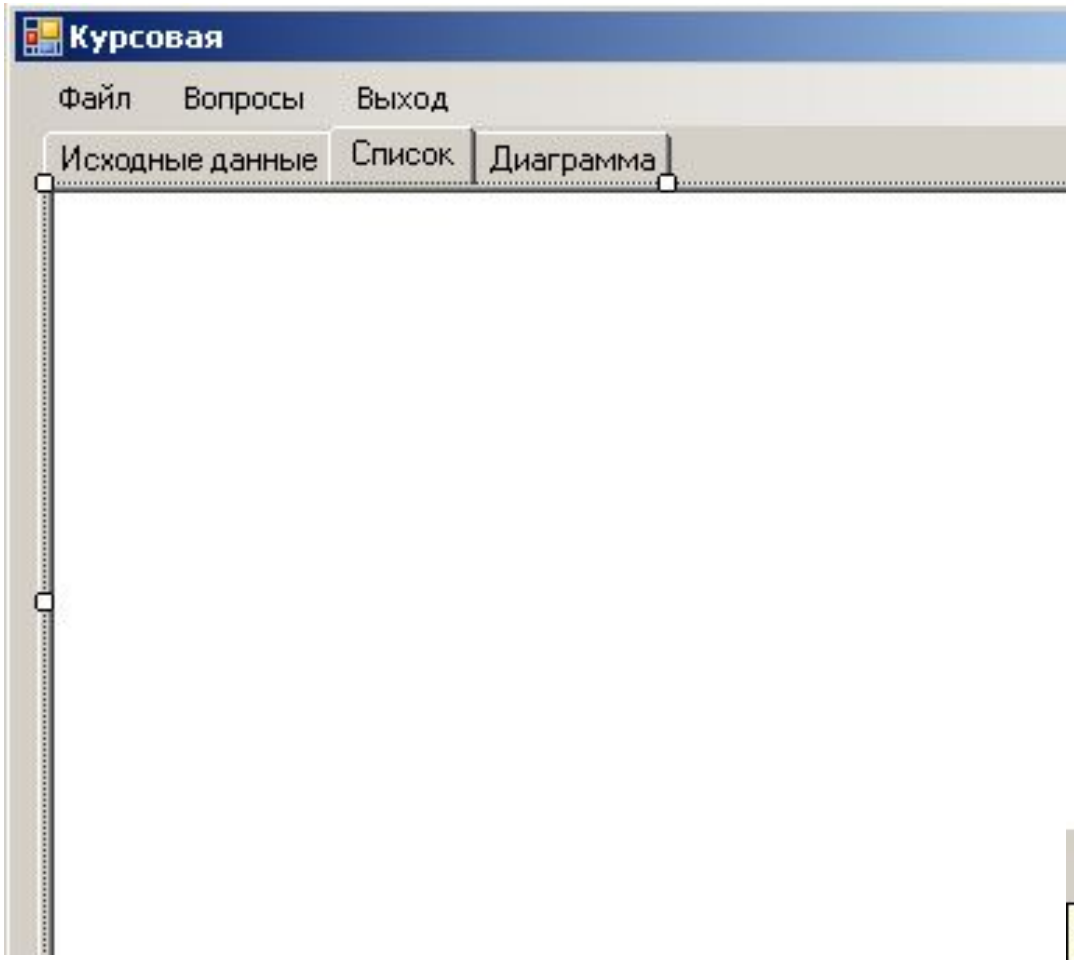
9. В результате форма приобретает вид

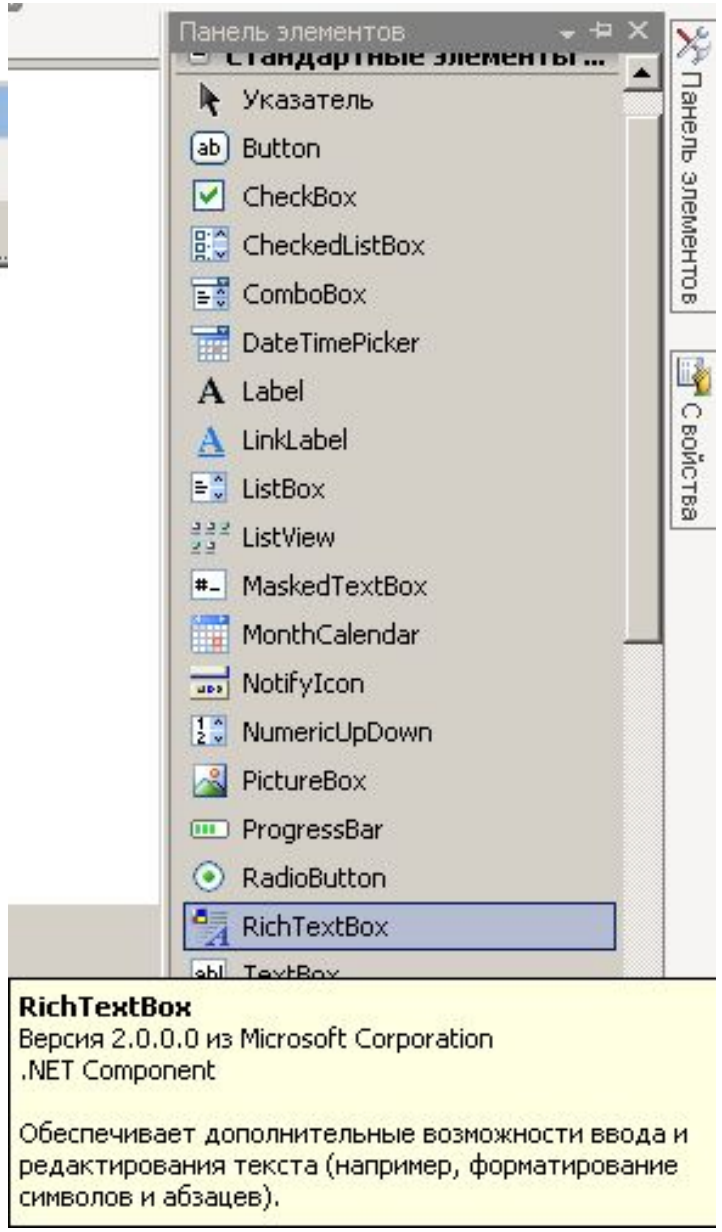


10. На вкладке «Исходные данные» размещаем элемент ListBox, который получит название **listBox1**

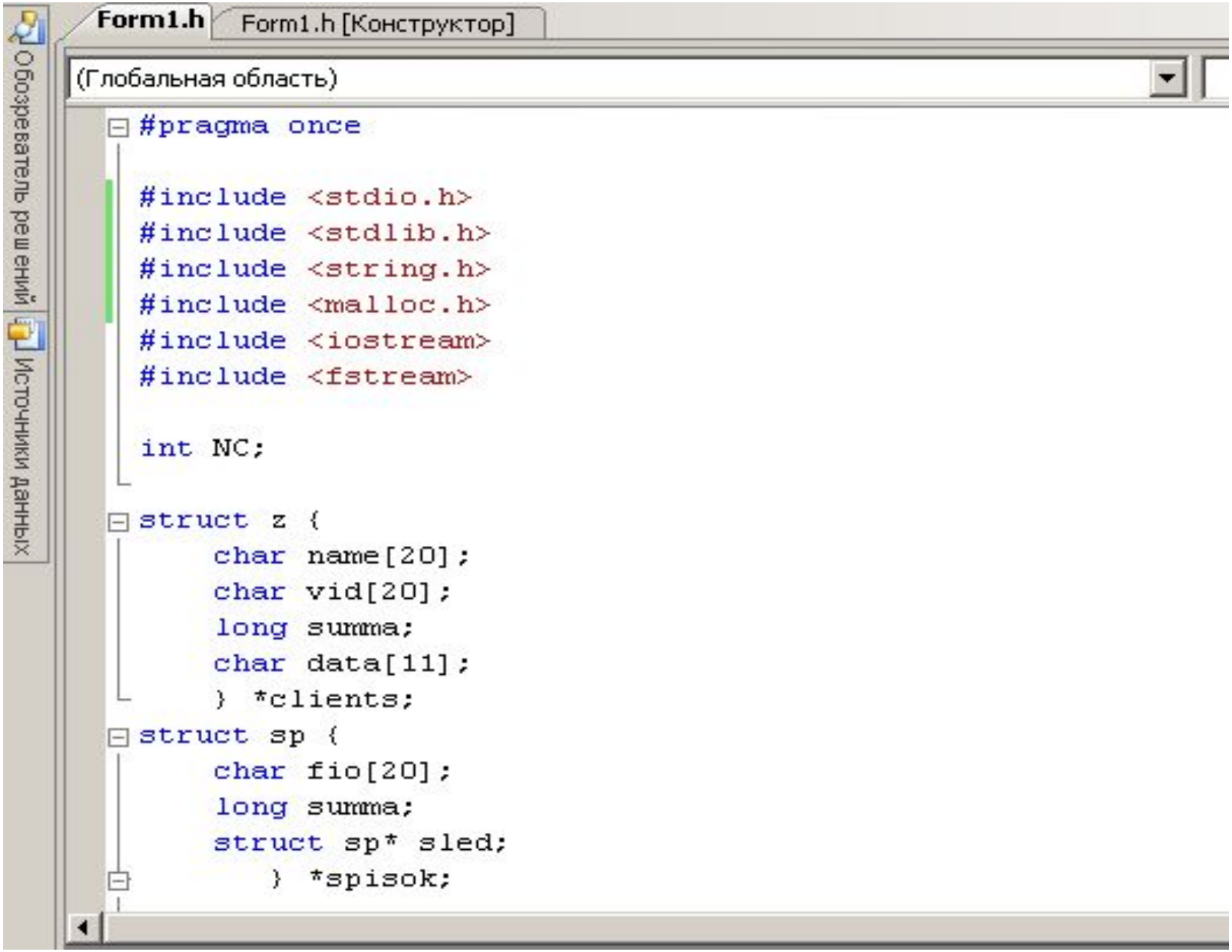


11. На вкладке «Список» размещаем элемент RichTextBox, который получит название **richTextBox1**





12. Приступаем к созданию программного кода



```
void vstavka(struct z* client,char* fio)
{
int i;
struct sp *nov,*nt,*z=0;
for(nt=spisok; nt!=0 && strcmp(nt->fio,fio)<0; z=nt, nt=nt->sled);
if(nt && strcmp(nt->fio,fio)==0) return;

nov=(struct sp *) malloc(sizeof(struct sp));
strcpy(nov->fio,fio);
nov->sled=nt;

nov->summa=0;
for(i=0;i<NC;i++)
      if(strcmp(client[i].name,fio)==0)
            nov->summa+=client[i].summa;

if(!z) spisok=nov;
else z->sled=nov;

return;
}
```

```
void text_data(char *s,char *sd)
{
int N_month;
char s0[3],month[12][9]={
"января","февраля","марта","апреля","мая","июня",
"июля","августа","сентября","октября","ноября","декабря"
                };
strcpy(s,sd+8);
strcat(s," ");
strncpy(s0,sd+5,2);
s0[2]=0;
sscanf(s0,"%d",&N_month);
strcat(s,month[N_month-1]);
strcat(s," ");
strncat(s,sd,4);
return;
}
```

13. Программируем пункты меню.
Выход.
Завершить.

```
private: System::Void выходToolStripMenuItem_Click
        (
        System::Object^  sender,
        System::EventArgs^  e
        )
                {
                this->Close();
                }


 private: System::Void завершитьToolStripMenuItem_Click
         (
         System::Object^  sender,
         System::EventArgs^  e
         )
                 {
                 this->Close();
                 }
```

14. Программируем пункты меню. Открыть.

```
private: System::Void открытьToolStripMenuItem_Click
        (
        System::Object^  sender,
         System::EventArgs^  e
        )
{
FILE *in;
int i;
char ctemp[80];
String ^s;
if ((in=fopen("Vklad.dat","r"))==NULL)
        {
        MessageBox::Show("Файл не открыт!",
                    "Ошибка!",
                    MessageBoxButtons::OK,
                    MessageBoxIcon::Error);
        return;
        }

вопросыToolStripMenuItem->Enabled=true;
```

```
                                    &clients[i].summa, clients[i].data);
    sprintf(ctemp,"%-20s %-20s %7ld %s",
                                clients[i].name,clients[i].vid,
                                clients[i].summa,clients[i].data);
    s=gcnew String(ctemp);
    listBox1->Items->Add(s);
    }
fclose(in);
}
    fscanf(in,"%s%s%ld%s",clients[i].name, clients[i].vid,
                                    &clients[i].summa, clients[i].data);
    sprintf(ctemp,"%-20s %-20s %7ld %s",
                                clients[i].name,clients[i].vid,
                                clients[i].summa,clients[i].data);
     s=gcnew String(ctemp);
     listBox1->Items->Add(s);
     }
fclose(in);
 }
```

```
listBox1->Items->Clear();
fscanf(in,"%d",&NC);
clients = new z[NC];
for(i=0;i<NC;i++)
{
```

15. Программируем пункты меню. Вопрос 1.

```
private: System::Void
какойВкладчикИмеетНаСчетуНаибольшуюСуммуДенегToolStripMenuItem_Click
                                        (System::Object^  sender, System::EventArgs^  e)
{
int i=0;
struct z best;
char ss[80];
String ^s;

strcpy(best.name,clients[0].name);
best.summa=clients[0].summa;
for(i=1;i<NC;i++)
      if (clients[i].summa>best.summa)
            {
            strcpy(best.name,clients[i].name);
            best.summa=clients[i].summa;
            }
sprintf(ss,"%s\n %ld рублей",best.name,best.summa);
s=gcnew String(ss);
MessageBox::Show(s,"Наибольший вклад");
 }
```

16. Программируем пункты меню. Вопрос 2.

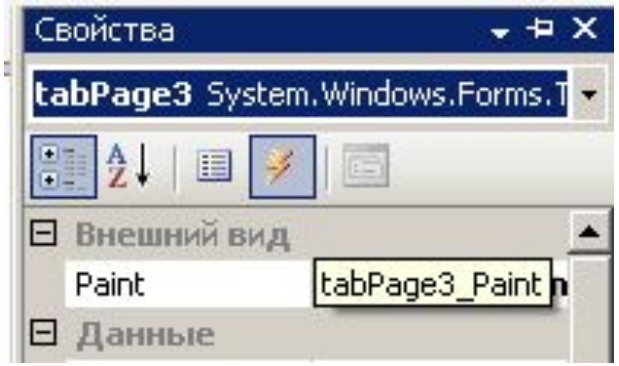
```
private: System::Void какойВкладБылОткрытРаньшеВсехToolStripMenuItem_Click
                                        (System::Object^  sender, System::EventArgs^  e)
{
int i;
char sd[17];
char ss[80];
String ^s;
struct z* best=clients;
for(i=1;i<NC;i++)
      if (strcmp(clients[i].data,best->data)<0) best=&clients[i];
text_data(sd,best->data);

sprintf(ss,"%s\n%s\n%ld рублей\nоткрыт %s",
                        best->vid,best->name, best->summa,sd);
s=gcnew String(ss);
MessageBox::Show(s,"Самый продолжительный вклад");
 }
```

17. Программируем пункты меню. Список.

```
private: System::Void алфавитныйСписокВсехВкладчиковToolStripMenuItem_Click
                                        (System::Object^  sender, System::EventArgs^  e)
{
int i;
struct sp* nt;
char ss[80];
String ^s;
tabControl1->SelectTab(1);  // переходим на вкладку "СПИСОК"
if(!spisok)
        for(i=0;i<NC;i++)
            vstavka(clients,clients[i].name);
richTextBox1->ReadOnly=1;
richTextBox1->Clear();
richTextBox1->Text ="\nАЛФАВИТНЫЙ СПИСОК\n==================\n";
for(nt=spisok; nt!=0; nt=nt->sled)
      {
      sprintf(ss,"%-20s %10ld",nt->fio,nt->summa);
      s=gcnew String(ss,0,31);
      richTextBox1->Text = richTextBox1->Text+"\n"+s;
      }
}
```

18. Программируем построение диаграммы

Метод **Paint** вызывается автоматически всегда, когда надо перерисовать элемент управления




```
private: System::Void tabPage3_Paint
(System::Object^  sender, System::Windows::Forms::PaintEventArgs^  e)
{
 Pen ^myPen= gcnew Pen(System::Drawing::Color::Black,3);
 SolidBrush ^myBrush= gcnew SolidBrush(Color::FromArgb(196,0,0,0));
 System::Drawing::Font^ myFont=gcnew System::Drawing::Font("Arial",8);

 int i,K;
 int iRed,iGreen,iBlue;
 int aStart,aEnd;
 long Sum;
 float xPos,yPos;
 struct sp *nt;

 Graphics ^g=tabPage3->CreateGraphics();
 g->Clear(System::Drawing::Color::White);
```

```
if(!spisok)
      for(i=0;i<NC;i++)
            vstavka(clients,clients[i].name);
K=0; Sum=0;
for(nt=spisok; nt!=0; nt=nt->sled)
      {
      K++;
      Sum+=nt->summa;
      }
g->DrawEllipse(myPen,25,25,185,185);
aEnd=0;

for(nt=spisok,i=0; nt!=0; nt=nt->sled,i++)
      {
      iRed=(((i+1)&4)>0)*255/(i/8+1);
      iGreen=(((i+1)&2)>0)*255/(i/8+1);
      iBlue=(((i+1)&1)>0)*255/(i/8+1);
      aStart=aEnd;
      aEnd+=nt->summa*360/Sum;
      if(i==K-1)aEnd=360;
```

```
    myBrush->Color::set(Color::FromArgb(196,iRed,iGreen,iBlue));
    g->FillPie(myBrush,25,25,185,185,aStart,aEnd-aStart);
    g->FillRectangle(myBrush,300,50+(i-1)*20,20,20);

    myBrush->Color::set(Color::FromArgb(196,0,0,0));
    xPos=30+(185-25)/2+(185-25)/1.5*Math::Cos(Math::PI*(aStart+aEnd)/360);
    yPos=30+(185-25)/2+(185-25)/1.5*Math::Sin(Math::PI*(aStart+aEnd)/360);

    g->DrawString(gcnew String(nt->fio),myFont,myBrush,320,55+(i-1)*20);
    g->DrawString(Convert::ToString(nt->summa*100/Sum)+"%",
                                myFont,myBrush,xPos,yPos);
    }  //конец цикла for…
}  // конец функции
```

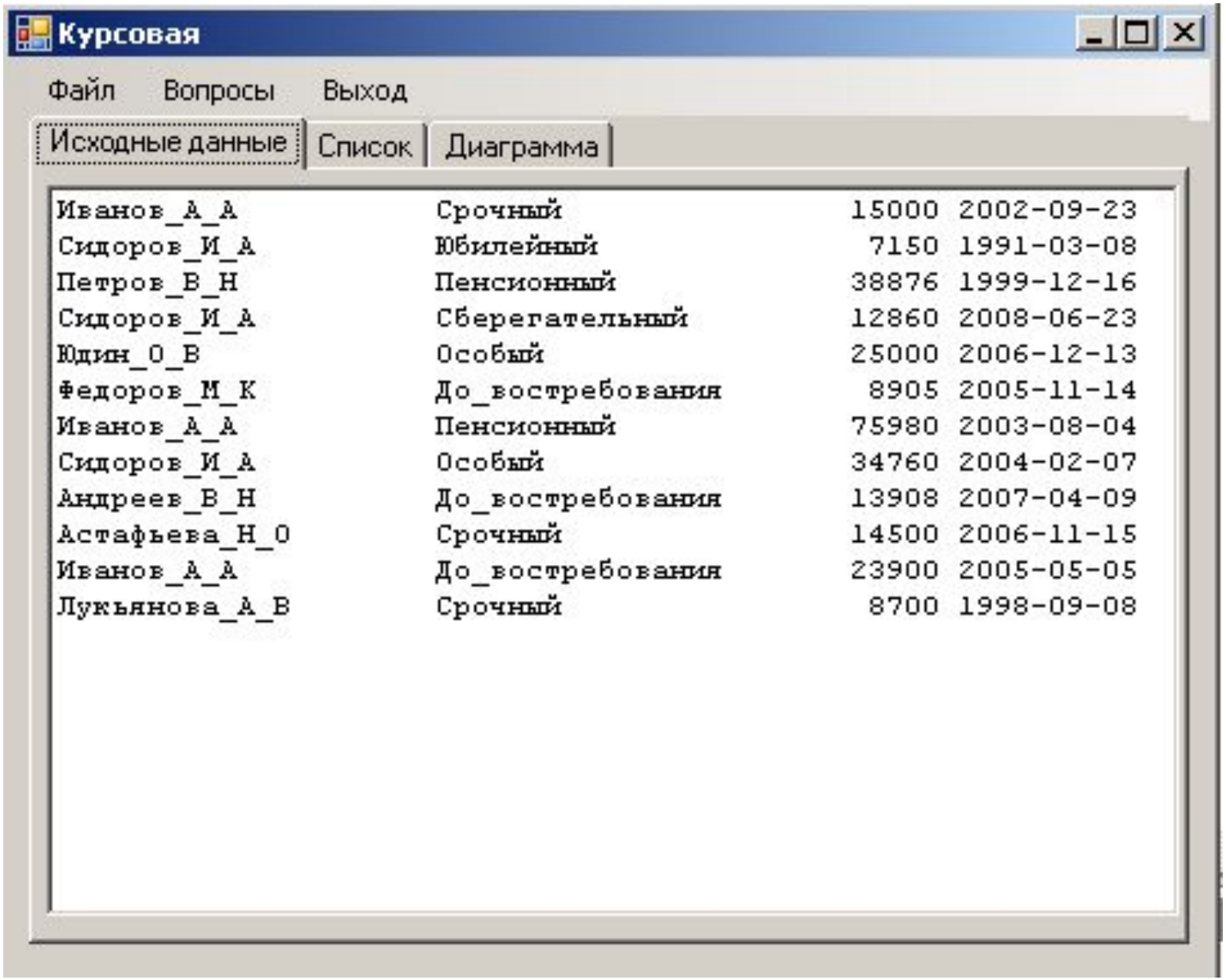
19. Программируем пункт меню Диаграмма…

Здесь нужно просто передать фокус вкладке «**Диаграмма**»,
чтобы автоматически вызвать метод **Paint** для **tabPage3**

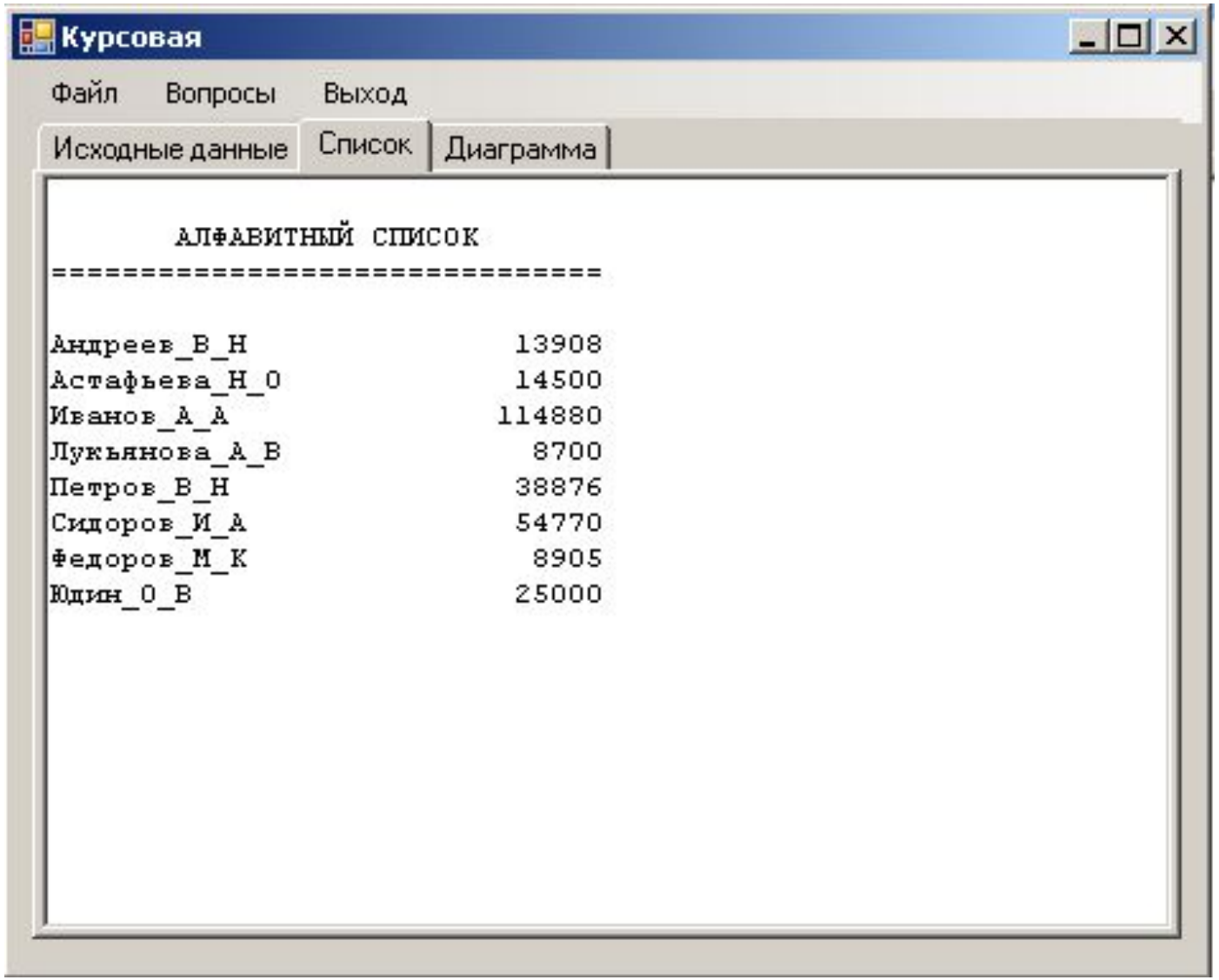
И описанный в 18-м пункте метод нарисует диаграмму.

```
private: System::Void
диаграммаПроцентноеСоотношениеВложенныхДенегToolStripMenuItem_Click
                (System::Object^  sender, System::EventArgs^  e)
 {
 tabControl1->SelectTab(2);  // переходим на вкладку «ДИАГРАММА"
 }
```

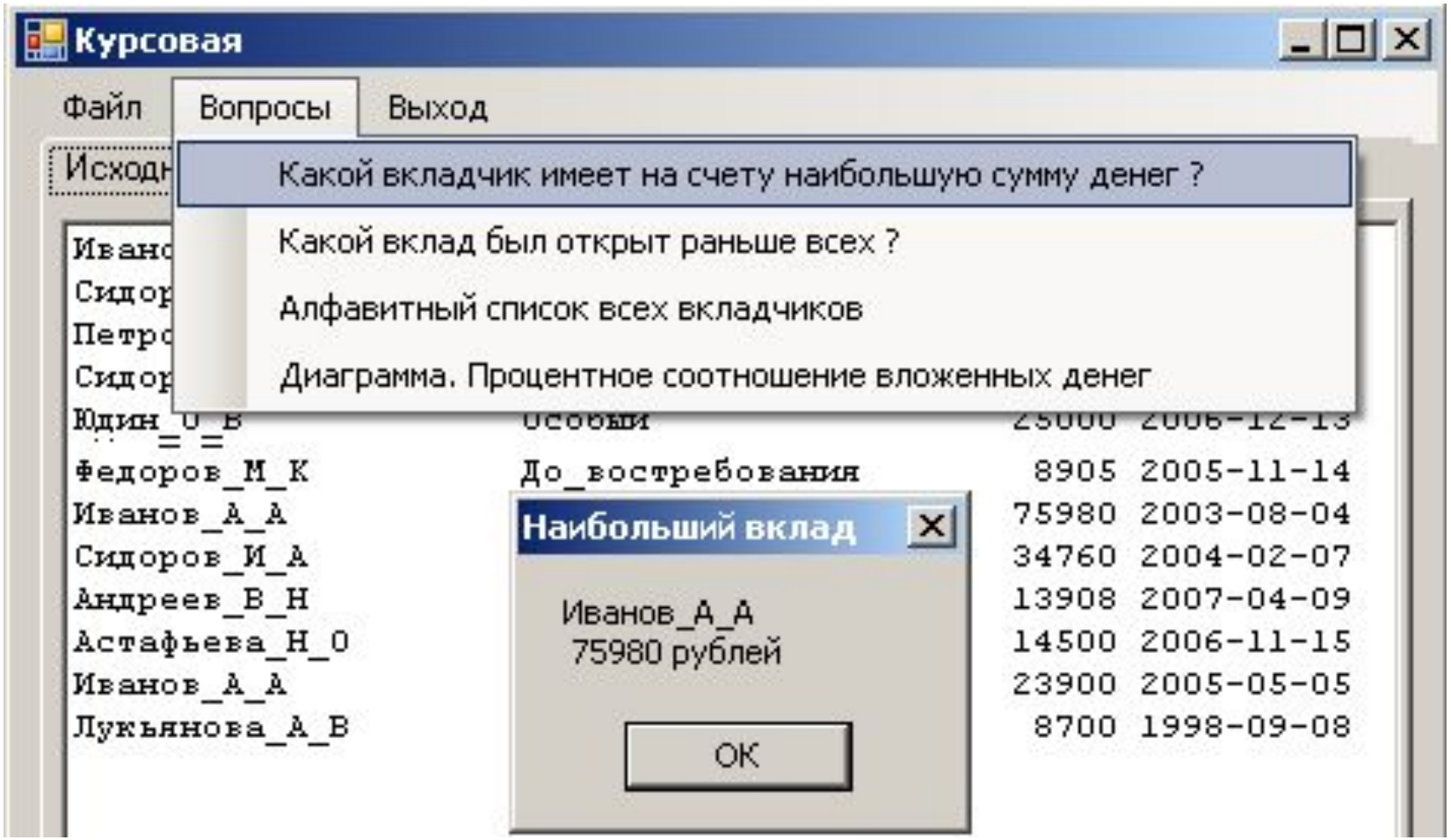
Запускаем на выполнение. Файл – Открыть …



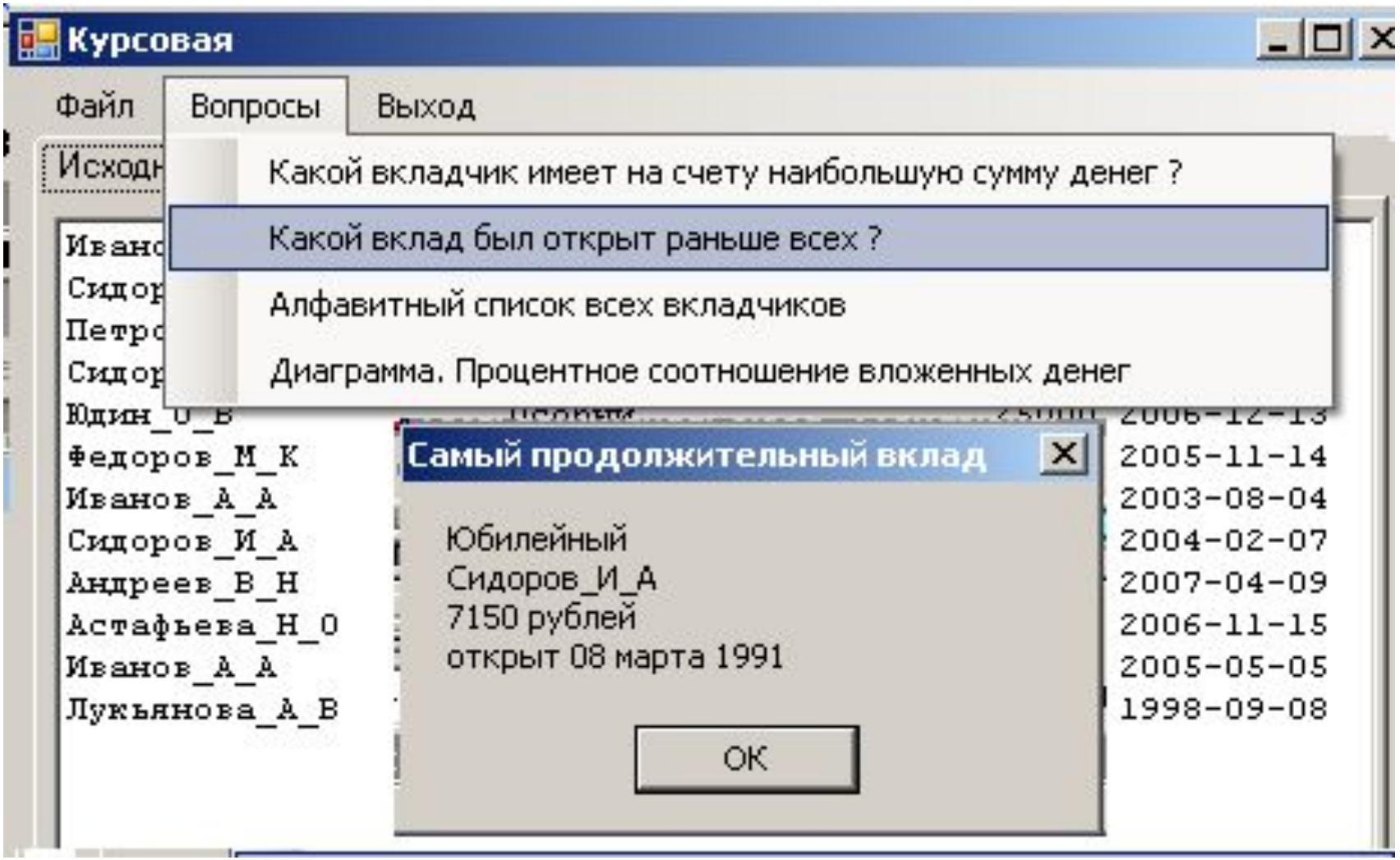
Вопросы – Алфавитный список всех вкладчиков



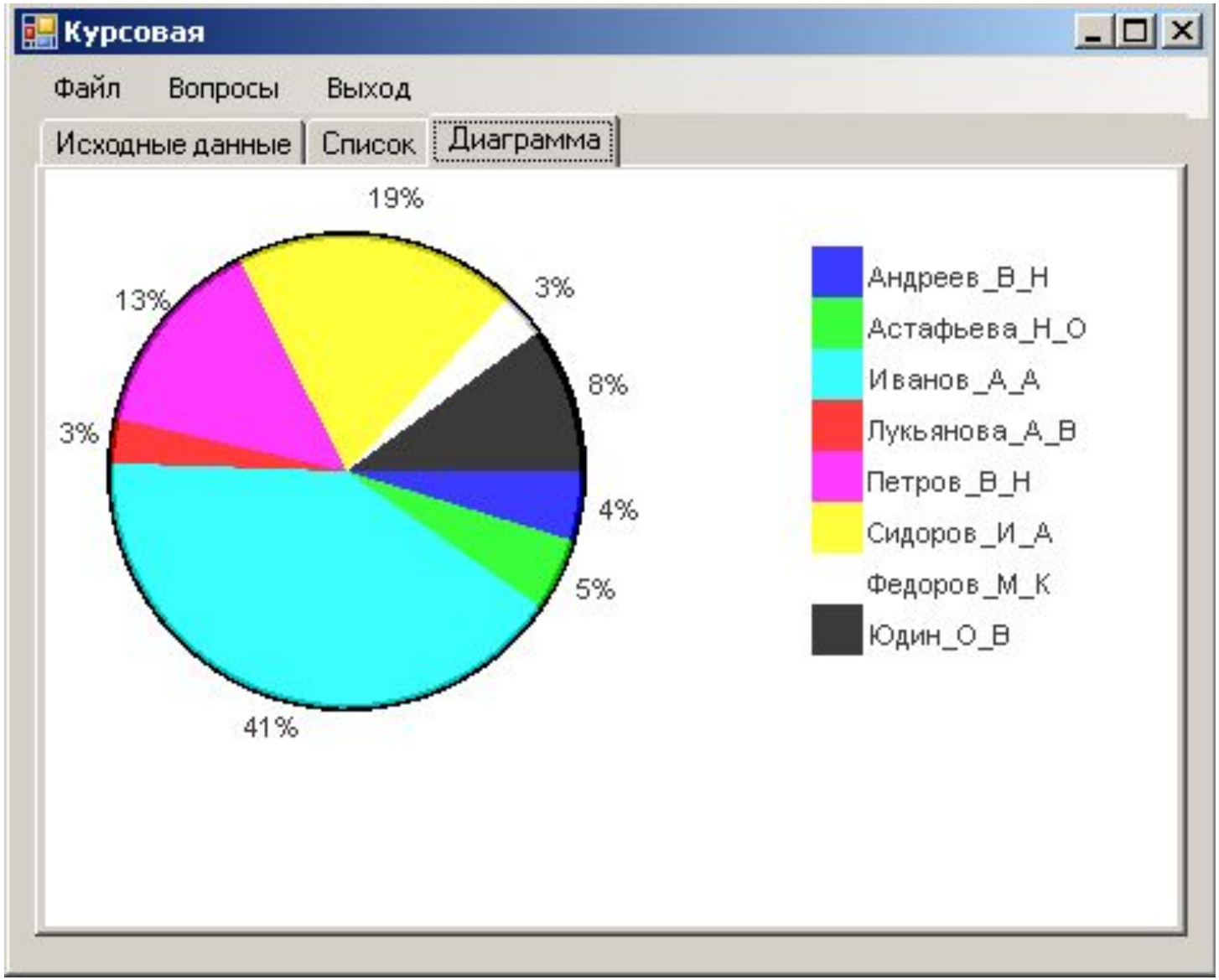
Вопросы – 1


Курсовая
Файл
Вопросы
Выход
Какой вкладчик имеет на счету наибольшую сумму денег ?
Какой вклад был открыт раньше всех ?
Алфавитный список всех вкладчиков
Диаграмма. Процентное соотношение вложенных денег
Федоров_М_К До_востребования 8905 2005-11-14
Иванов_А_А 75980 2003-08-04
Сидоров_И_А 34760 2004-02-07
Андреев_В_Н 13908 2007-04-09
Астафьева_Н_О 14500 2006-11-15
Иванов_А_А 23900 2005-05-05
Лукьянова_А_В 8700 1998-09-08
Наибольший вклад
Иванов_А_А
75980 рублей
OK

Вопросы – 2

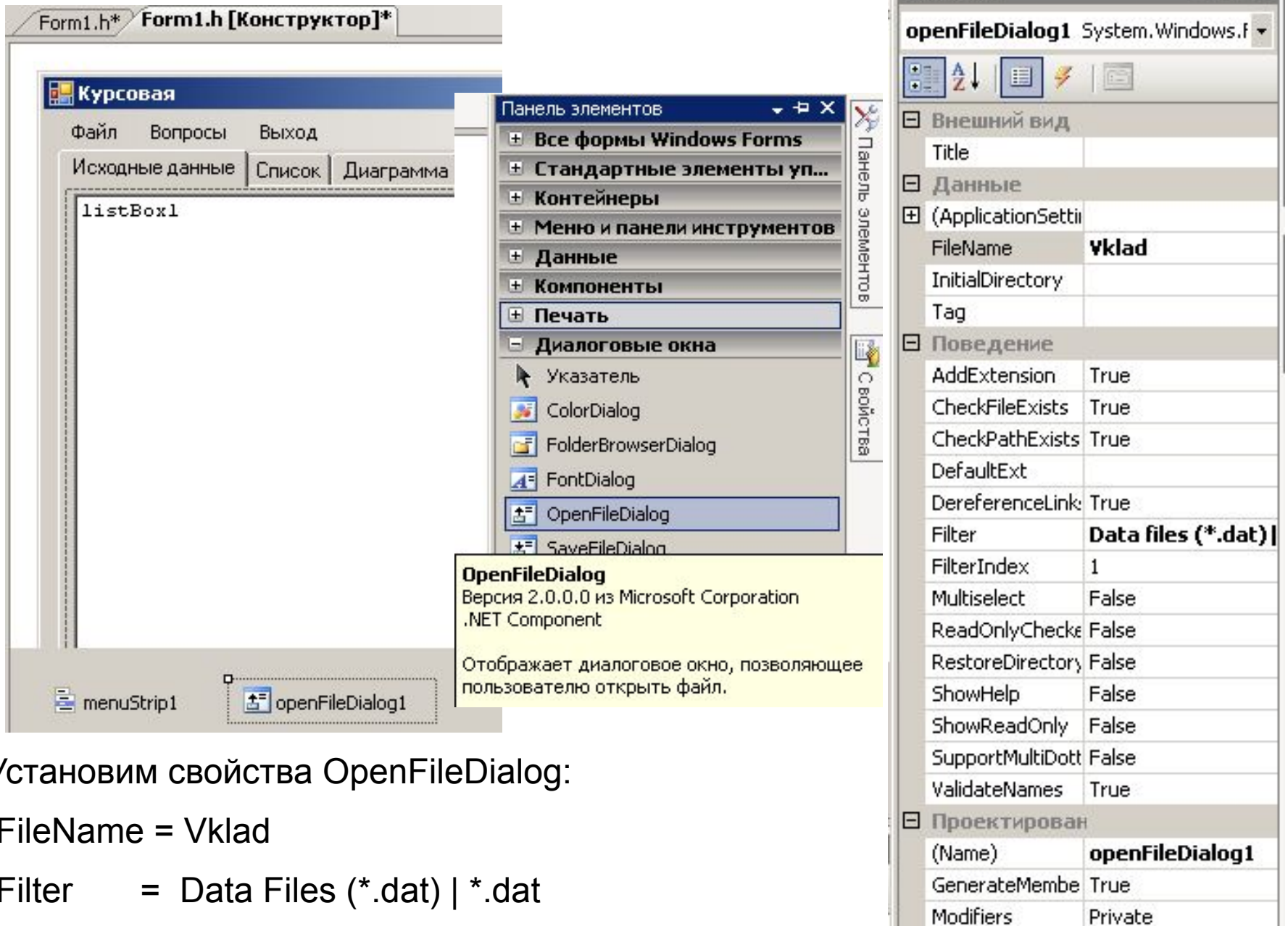

Курсовая
Файл
Вопросы
Выход
Какой вкладчик имеет на счету наибольшую сумму денег ?
Какой вклад был открыт раньше всех ?
Алфавитный список всех вкладчиков
Диаграмма. Процентное соотношение вложенных денег
Федоров_М_К
Иванов_А_А
Сидоров_И_А
Андреев_В_Н
Астафьева_Н_О
Иванов_А_А
Лукьянова_А_В
2005-11-14
2003-08-04
2004-02-07
2007-04-09
2006-11-15
2005-05-05
1998-09-08
Самый продолжительный вклад
Юбилейный
Сидоров_И_А
7150 рублей
открыт 08 марта 1991
OK

Диаграмма


Курсовая
Файл
Вопросы
Выход
Исходные данные
Список
Диаграмма
19%
3%
13%
8%
3%
4%
5%
41%
Андреев_В_Н
Астафьева_Н_О
Иванов_А_А
Лукьянова_А_В
Петров_В_Н
Сидоров_И_А
Федоров_М_К
Юдин_О_В

Доработка проекта

20. Добавим на форму компонент OpenFileDialog



Установим свойства OpenFileDialog:

FileName = Vklad

Filter = Data Files (*.dat) | *.dat

21. Перепрограммируем пункт меню «Открыть».

Было

```
if ((in=fopen("Vklad.dat","r"))==NULL)
        {
        MessageBox::Show("Файл не открыт!","Ошибка!",
            MessageBoxButtons::OK,MessageBoxIcon::Error);
        return;
        }
```

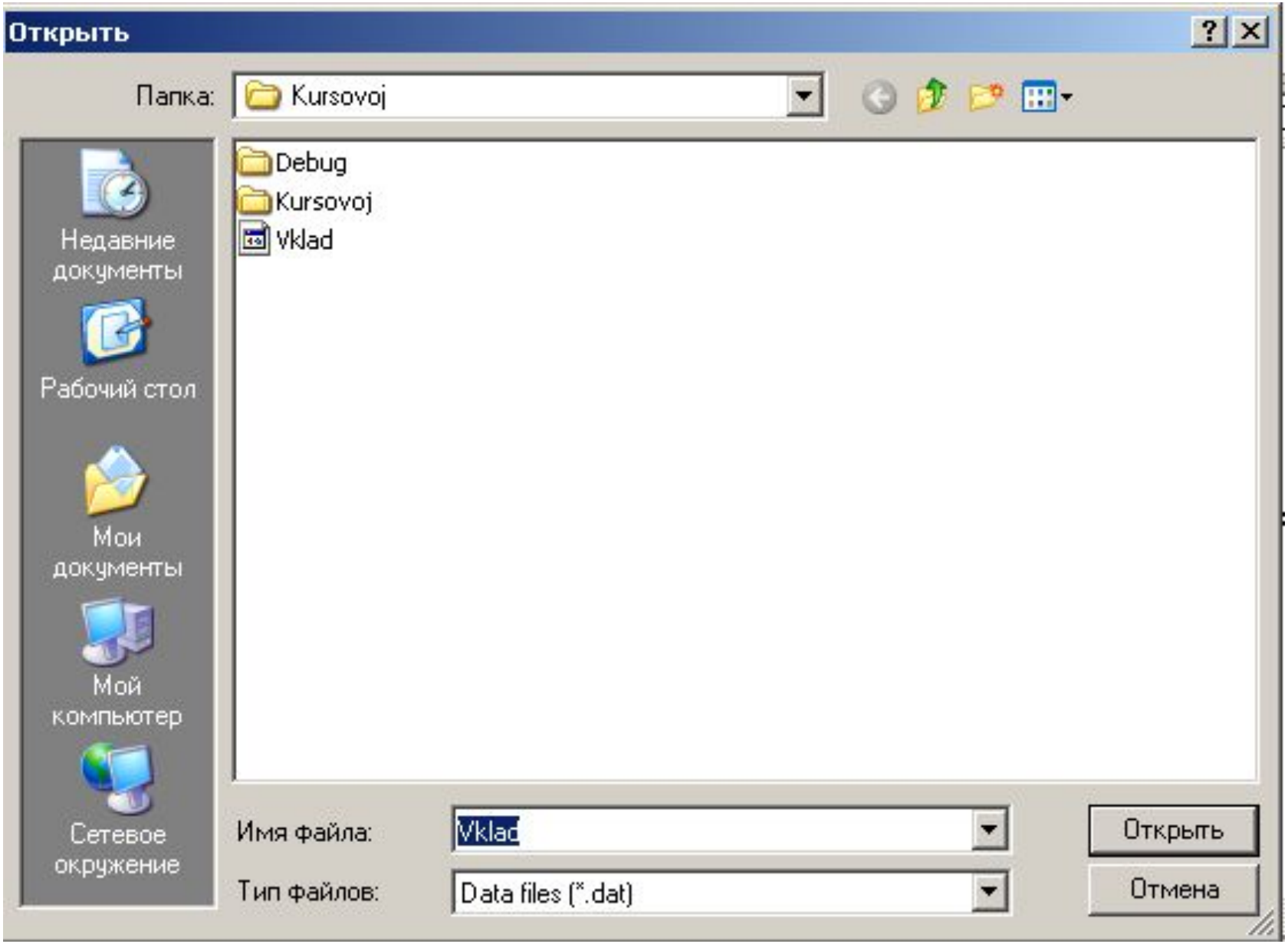
Делаем

```
if (openFileDialog1->ShowDialog()==System::Windows::Forms::DialogResult::OK)
    {
    s=openFileDialog1->FileName;
    char *str_tmp=(char*)(void*)Marshal::StringToHGlobalAnsi(s);
    if ((in=fopen(str_tmp,"r"))==NULL)
        {   MessageBox::Show("Файл не открыт!","Ошибка!",
            MessageBoxButtons::OK,MessageBoxIcon::Error);
        return;
        }
    }
else return;
```

В начало добавим (где идет серия команд **usung namespace)**

```
using namespace System::Runtime::InteropServices;
```

```
namespace Kursovoj {

    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;
    using namespace System::Runtime::InteropServices;

    /// <summary>
    /// Сводка для Form1
```


Открыть
Папка:
Kursovoj
Debug
Kursovoj
Vklad
Недавние документы
Рабочий стол
Мои документы
Мой компьютер
Сетевое окружение
Имя файла:
Vklad
Тип файлов:
Data files (*.dat)
Открыть
Отмена

Как один проект разбить на несколько файлов

```
// Файл Struct.h – описание шаблонов структур
//                        + прототипы функций
#ifndef Struct_H   /*При подключении к другим файлам фрагмент от #IFNDEF*/
#define Struct_H  /*до #ENDIFбудет обрабатывается только один раз*/
/* Трактуется так: если Struct_H не задана, то . . . , ИНАЧЕ ничего не делать*/
struct z {
        char name[20];
        char vid[20];
        long summa;
        char data[11];
        };
struct sp {
        char fio[20];
        long summa;
        struct sp* sled;
          };
#endif //  Struct_H
/*При многократном использовании исключено повторное описание шаблона*/
/*
Здесь размещаются прототипы всех функций. Отовсюду их теперь можно
убирать, т.к. за счет #inclide “Struct.h” они везде будут  подключаться.
Вообще, прототипы могут обрабатываться компилятором несколько раз!!!
А вот повторное описание шаблона структуры НЕДОПУСТИМО!!!
Именно по этой причине мы применяем механизм #ifndef-#define-#endif
*/
```

Как один проект разбить на несколько файлов

// Файл Main.c

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <alloc.h>
/* <malloc.h> для c++ */

#include "Struct.h "

int NC;
struct z *clients;
struct sp *spisok;


main()
{
. . .
}
```

// Файл Func.c

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <alloc.h>
/* <malloc.h> для c++ */

#include "Struct.h "

extern int NC;
extern struct z *clients;
extern struct sp *spisok;

/*
Здесь можно разместить реализацию  функций,
прототипы которых в Struct.h.
А вызываются они, например, из main()
*/
```

Файлы **Struct.h** и **Func.c** просто добавляются в проект, к уже имеющемуся там (как правило) Main.c

Файлов типа **Func.c** может несколько. Их строения подобны!

Преобразования для типа **String**.

```
//    Преобразование String в char

//    Требуется подключить
      using namespace System::Runtime::InteropServices;

      String ^s;
      char *str_tmp=(char*)(void*)Marshal::StringToHGlobalAnsi(s);

//    Обратно
      char *str;
      String ^s=gcnew String(str);

//    Число в String
      String ^s=Convert::ToString(число);

//    Обратно
      int N=Int32::Parse(s);
```