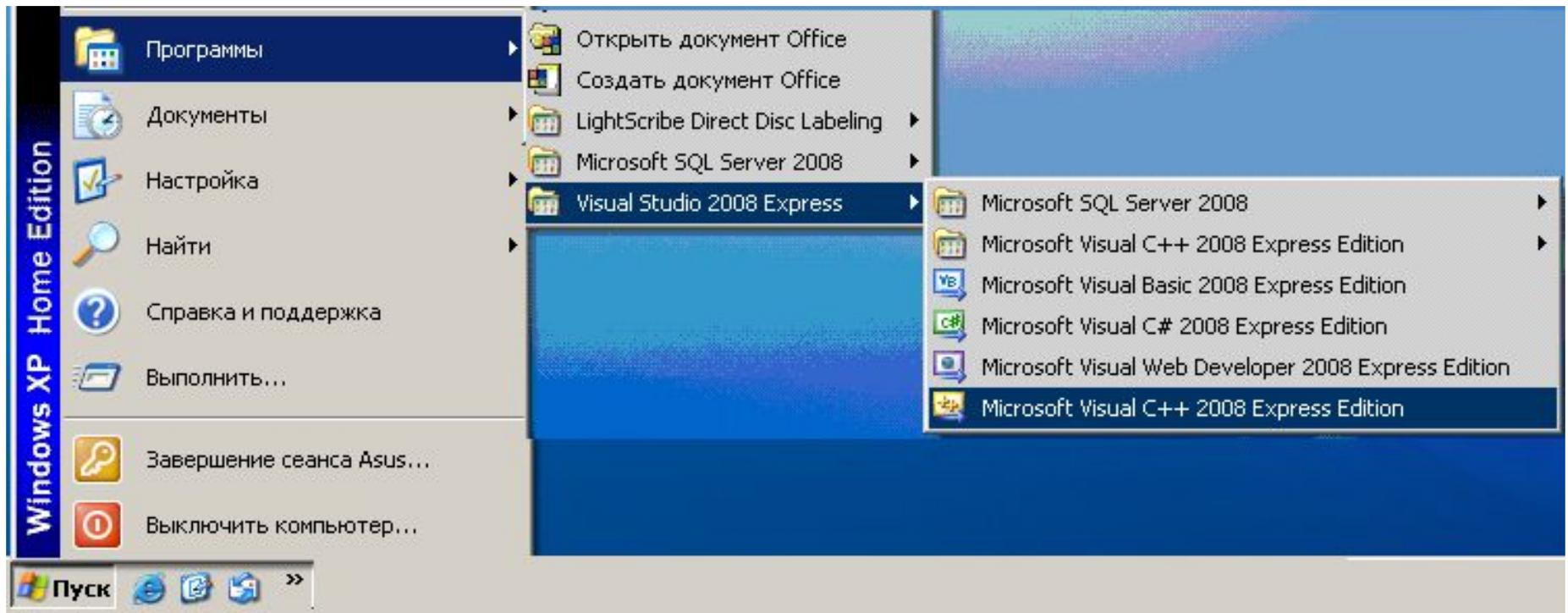


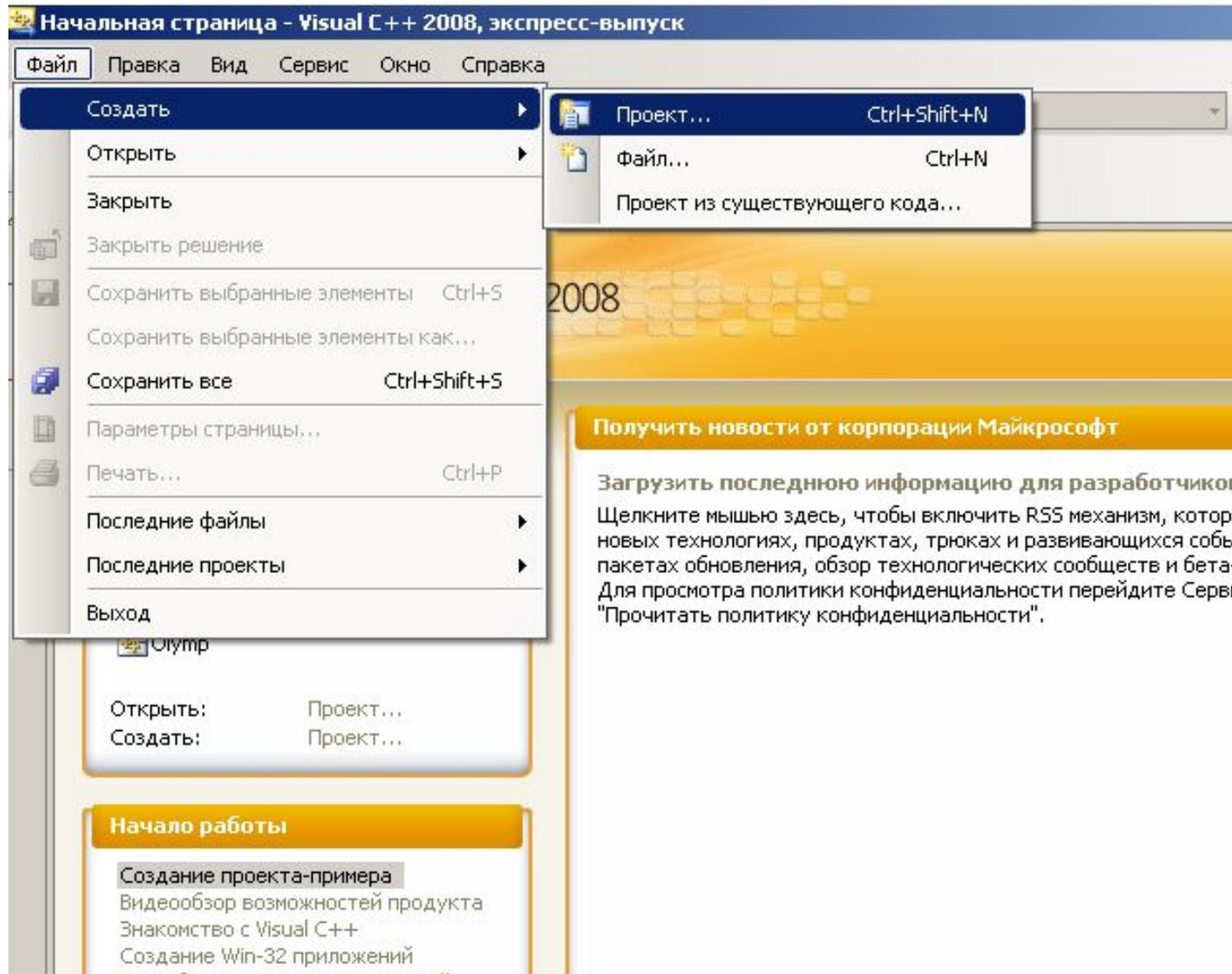
Отладка простейшей программы в Microsoft Visual Studio 2008 Express

1. Запускаем среду

Microsoft Visual C++ Express Edition

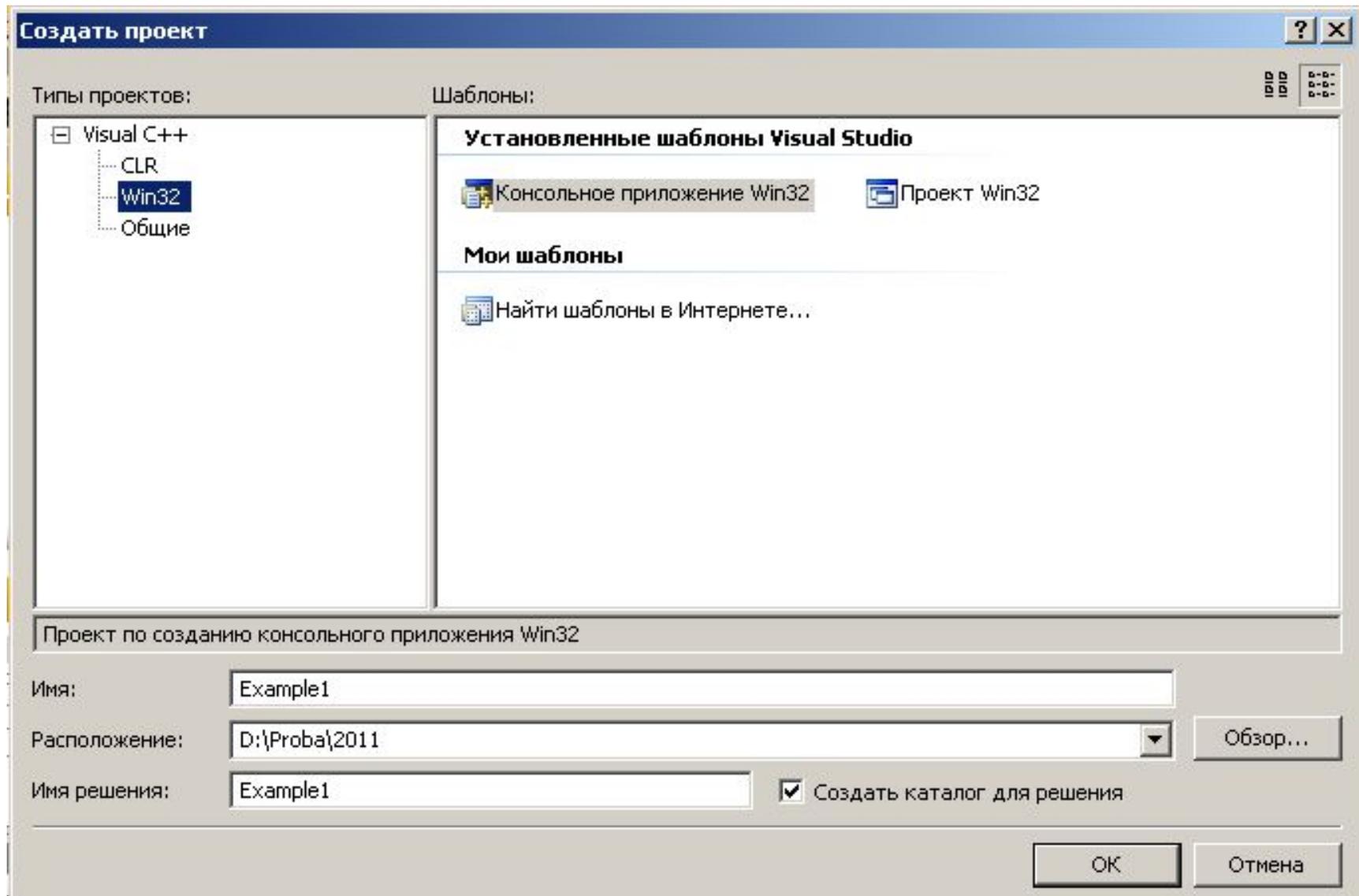


2. Создаем новый проект

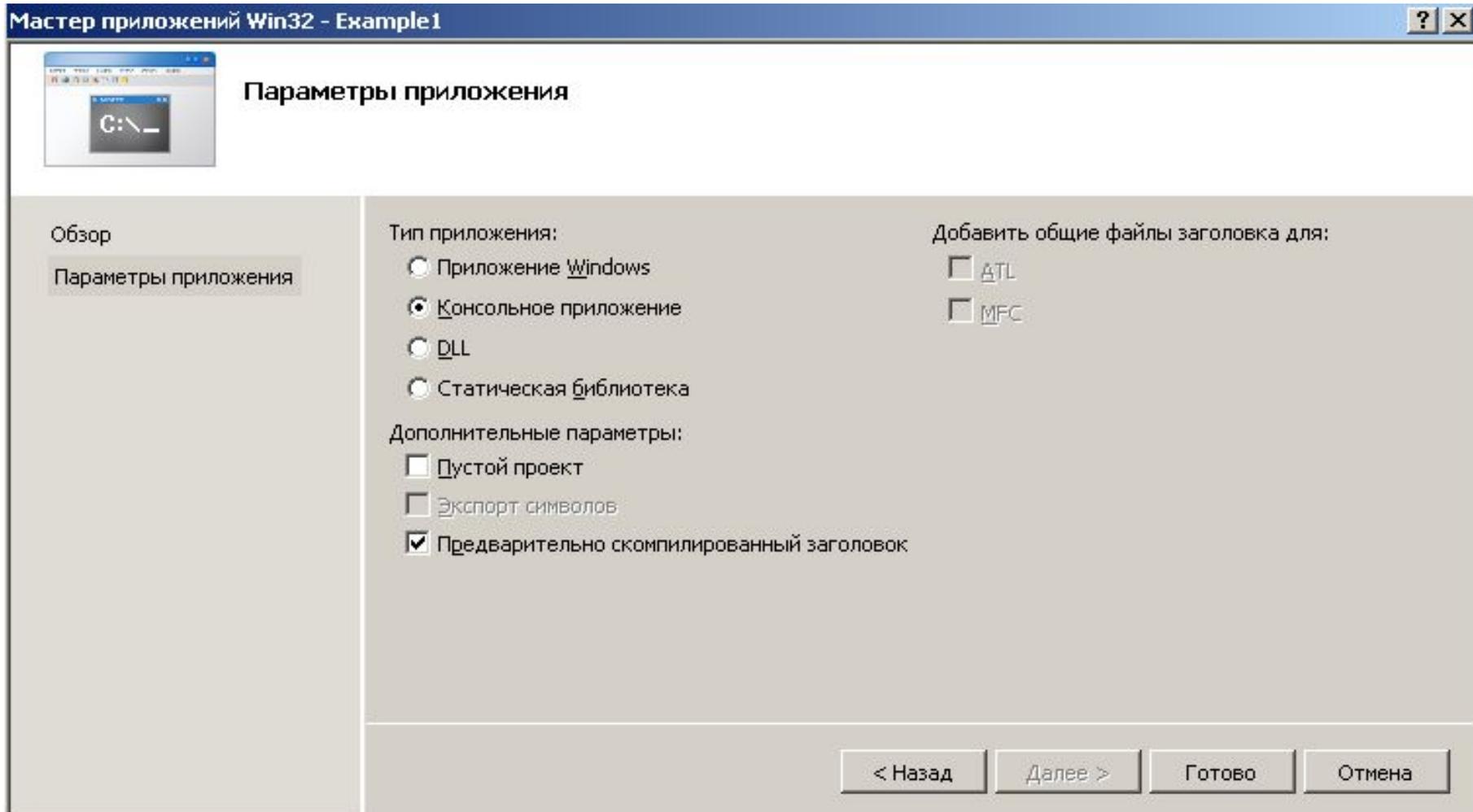


3. Задаем тип (Консольное приложение Win32), название (Example1) и каталог для размещения проекта.

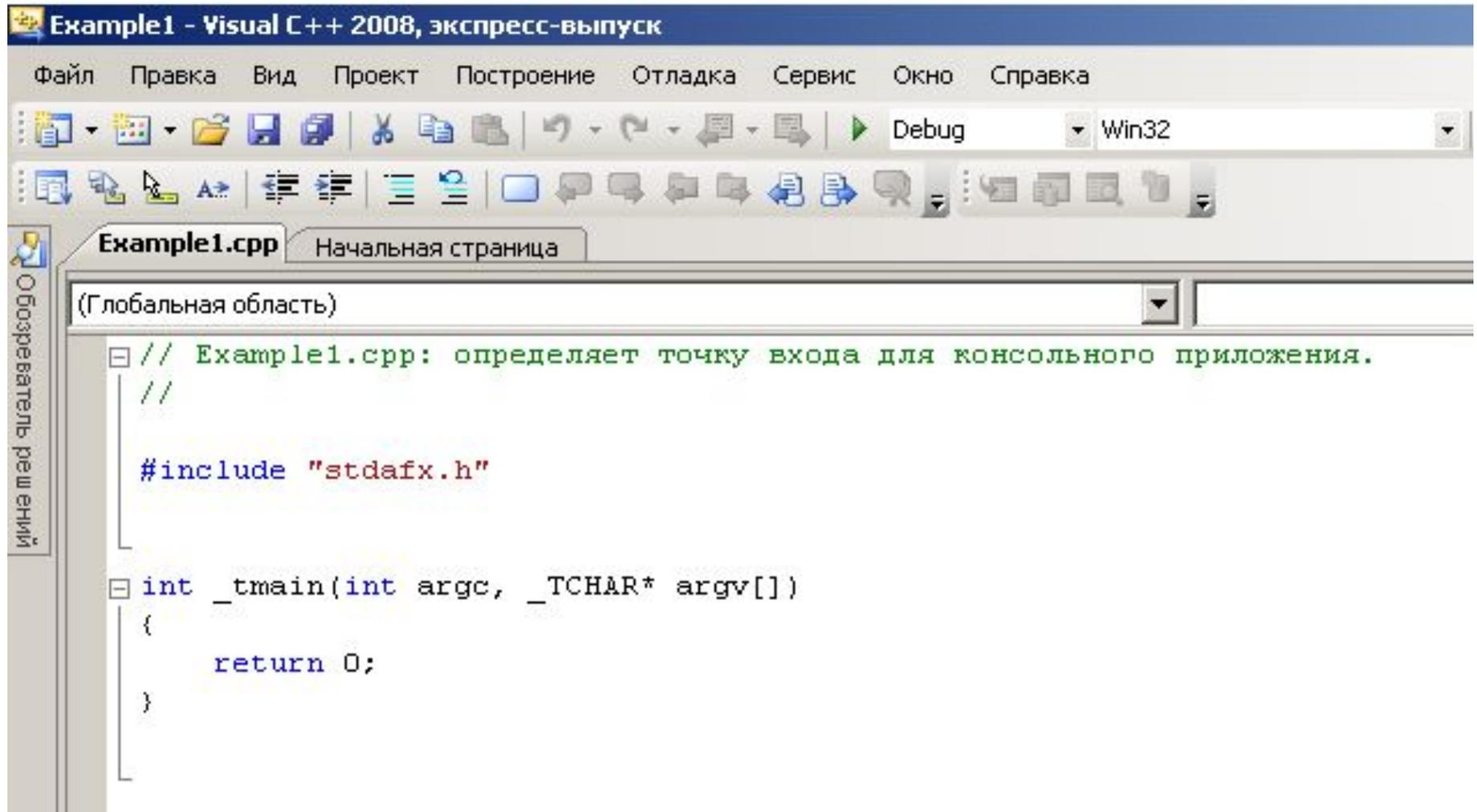
Нажимаем <Ok>



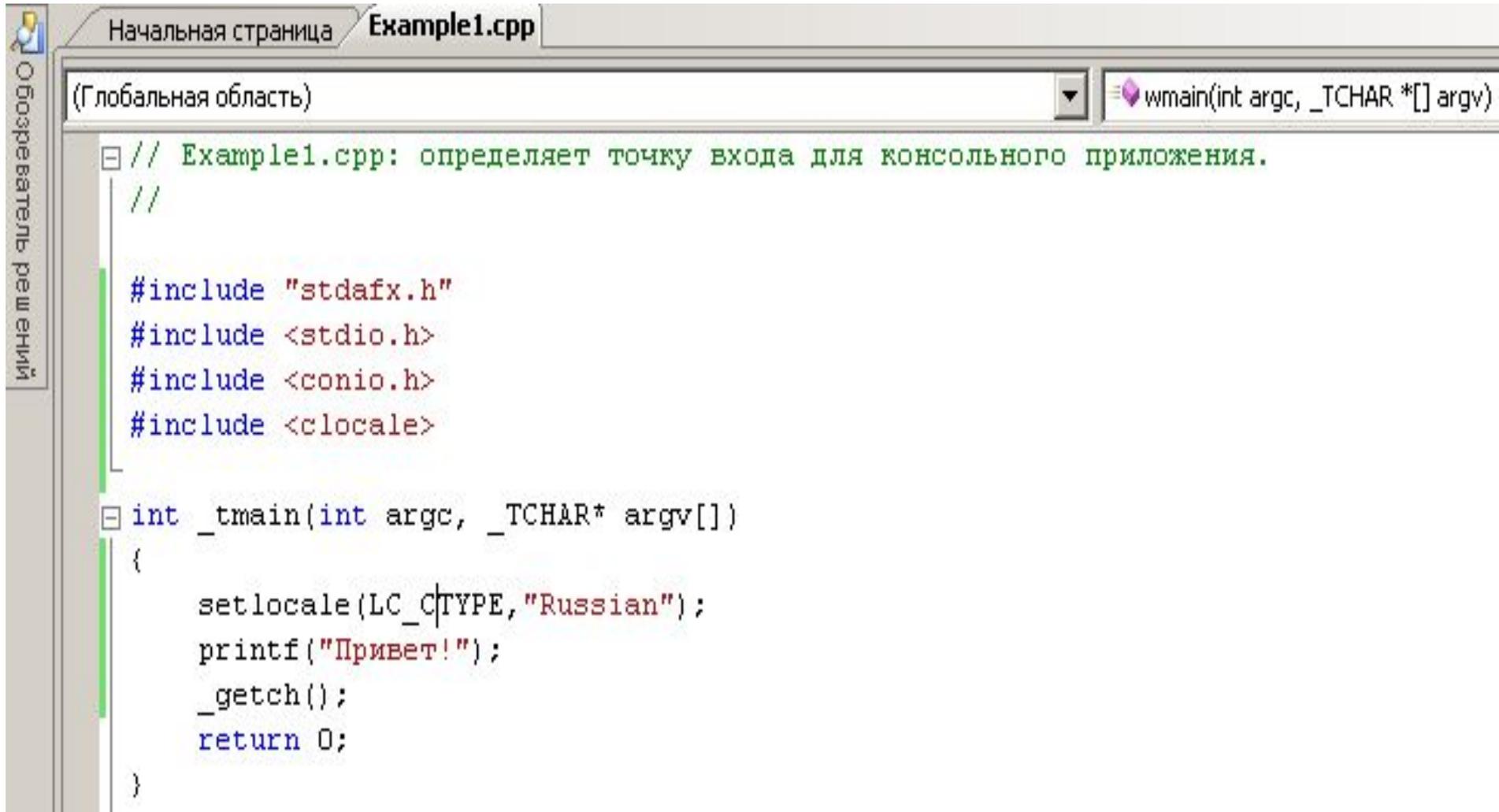
4. Следуя указаниям мастера, принимаем все параметры по умолчанию, нажимаем <Готово>



5. Получаем стандартную заготовку



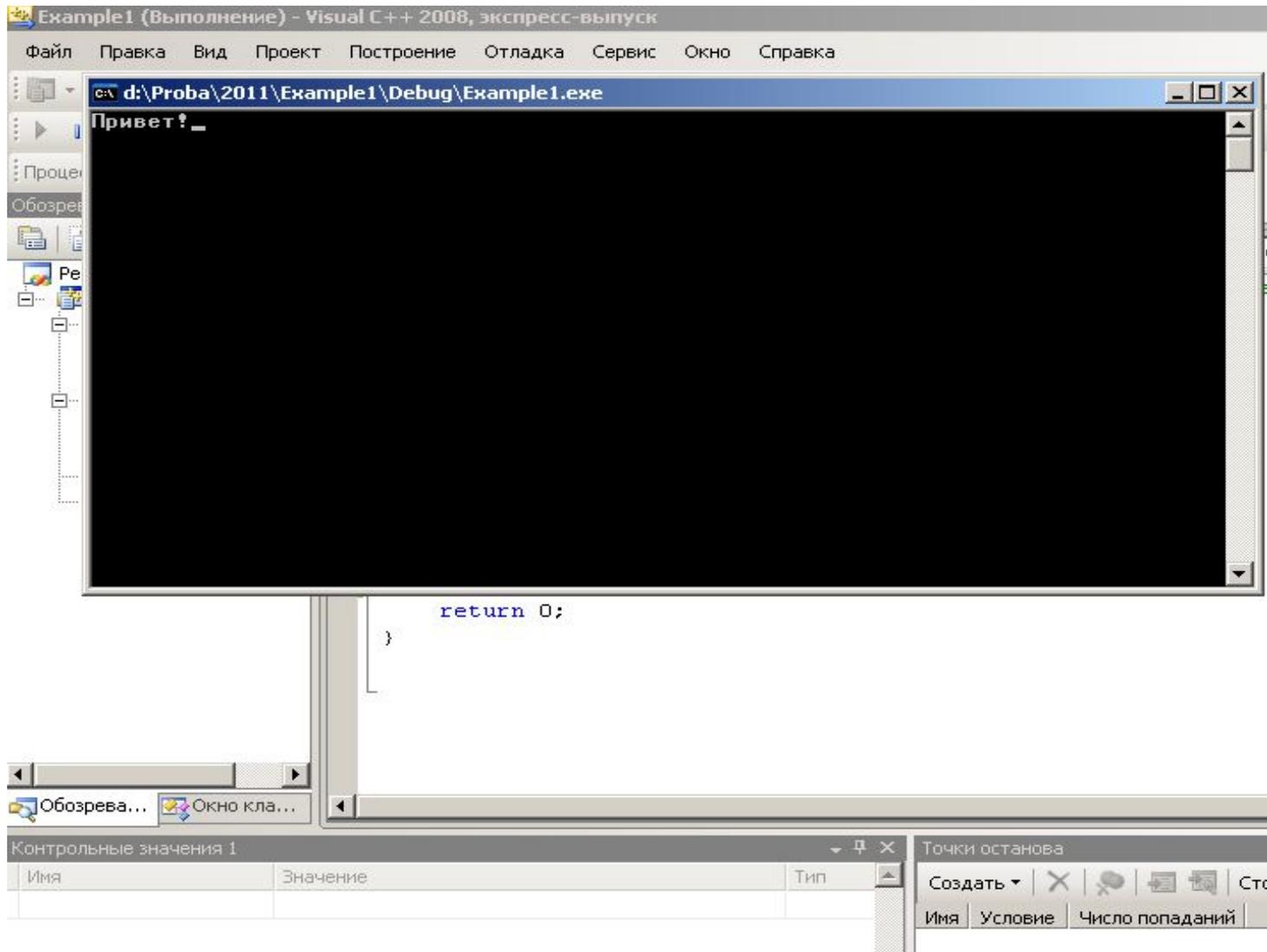
Которую дополняем так, что получится следующий вариант программы



```
Начальная страница Example1.cpp
(Глобальная область) wmain(int argc, _TCHAR *[] argv)
// Example1.cpp: определяет точку входа для консольного приложения.
//
#include "stdafx.h"
#include <stdio.h>
#include <conio.h>
#include <locale>

int _tmain(int argc, _TCHAR* argv[])
{
    setlocale(LC_CTYPE, "Russian");
    printf("Привет!");
    _getch();
    return 0;
}
```

Запускаем программу на выполнение



Примечания:

1. Вместо функции `main()` здесь используется

```
int _tmain(int argc, _TCHAR *argv[])
```

2. Библиотеки `<stdio.h>` и `<conio.h>` необходимо добавить вручную.

3. Функция `_getch()`, как мы видим, теперь слева дополнена символом подчеркивания. Однако, для совместимости, она будет работать и в прежнем виде, без символа подчеркивания, но выдаст предупреждение (Warning)

4. Для правильного отображения кириллицы при исполнении программы необходимо применить функцию

```
setlocale(LC_STYPE, "Russian");
```

прототип которой подключается через

```
#include <locale>
```

Пример 1

Применение **Console::WriteLine** вместо **printf**

```
#include "stdafx.h"
```

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
using namespace System;
```

```
// Нужно добавить
```

```
using namespace System::IO;
```

```
//using namespace std;
```

```
int main(array<System::String ^> ^args)
```

```
{
```

```
return 0;
```

```
}
```

Пример 1 (продолжение)

```
int main(array<System::String ^> ^args)
{
    int x,y,z,u;
```

```
    Console::WriteLine(L"Введите x,y,z");
```

```
    //printf("Введите x,y,z"); // не печатает по русски
```

```
    scanf ("%d%d%d", &x, &y, &z) ;
```

```
    if (x>y)
```

```
        if (x>z)
```

```
            u=x;
```

```
        else
```

```
            u=z;
```

```
    else
```

```
        if (y>z)
```

```
            u=y;
```

```
        else
```

```
            u=z;
```

```
    Console::WriteLine ("x={0} \ny={1} \nz={2} \nu={3}", x, y, z, u) ;
```

```
    Console::ReadKey() ; // вместо getch();
```

```
    return 0;
```

```
}
```

{0} – ОКНО для x

{1} – ОКНО для y

{2} – ОКНО для z

{3} – ОКНО для u

Пример 1 (продолжение)

Формат окна:

{номер_аргумента_в_списке,[-]ширина,тип}

[-] означает выравнивание влево

Тип:

D(d)12345

D8 00012345

E 1.234568E+004

E10 1.2345678900E+004

E4 1.2346E+004

F(f) 17843.00

F3 29541.000

F0 18934

X

C

Внимание!!!

WriteLine умеет выводить только символьные строки в расширенной UNICODE кодировке, которую обеспечивает входящий в MSVS класс String.

Обработка стандартных char* строк не поддерживается, однако, их можно перекодировать в нужный вид.

Пример 2

```
using namespace System::Threading;
ConsoleKeyInfo cki;
while(Console::KeyAvailable==false)
    {
    //  аналог      while (!kbhit())
    }
cki = Console::ReadKey(true);
true – нажатая клавиша не отображается
false или пусто - клавиша отображается
Console::WriteLine("Вы нажали '{0}' ", cki.Key);

Thread::Sleep(250);
Thread::Abort(); //  аналог exit()
```

Turbo C	MSVS 2008
Установить размер окна. Убрать курсор.	<pre>Console::BufferHeight=Console::WindowHeight; Console::BufferWidth=Console::WindowWidth; Console::CursorVisible::set(false);</pre>
<pre>textcolor(YELLOW);</pre>	<pre>Console::ForegroundColor=ConsoleColor::Yellow;</pre>
<pre>textbackground(CYAN);</pre>	<pre>Console::BackgroundColor=ConsoleColor::Cyan;</pre>
<pre>clrscr();</pre>	<pre>Console::Clear();</pre>
<pre>gotoxy(10,16);</pre>	<pre>Console::CursorLeft=10; Console::CursorTop=16;</pre>
<p style="text-align: center;">Отображение строк меню в окне</p> <pre>window(10,7,70,15); textbackground(LIGHTGRAY); clrscr(); textcolor(BLACK); cprintf("\n"); for(i=0; i<7; i++) cprintf(" %s\n\r",dan[i]);</pre>	<pre>char BlankLine[]=" "; Console::CursorLeft=10; Console::CursorTop=4; printf(BlankLine); for(i=0;i<7;i++) { Console::CursorLeft=10; Console::CursorTop=i+5; printf(" %s ",dan[i]); } Console::CursorLeft=10; Console::CursorTop=12; printf(BlankLine);</pre>

Пример выполнения
курсовой работы
в среде
Microsoft Visual Studio 2008
Express

В качестве файла с исходными данными используем

D:\Proba\2011\VC_ACY\VC_ACY\VKLAD.DAT

12

Иванов_А_А Срочный 15000 2002-09-23

Сидоров_И_А Юбилейный 7150 1991-03-08

Петров_В_Н Пенсионный 38876 1999-12-16

Сидоров_И_А Сберегательный 12860 2008-06-23

Юдин_О_В Особый 25000 2006-12-13

Федоров_М_К До_востребования 8905 2005-11-14

Иванов_А_А Пенсионный 75980 2003-08-04

Сидоров_И_А Особый 34760 2004-02-07

Андреев_В_Н До_востребования 13908 2007-04-09

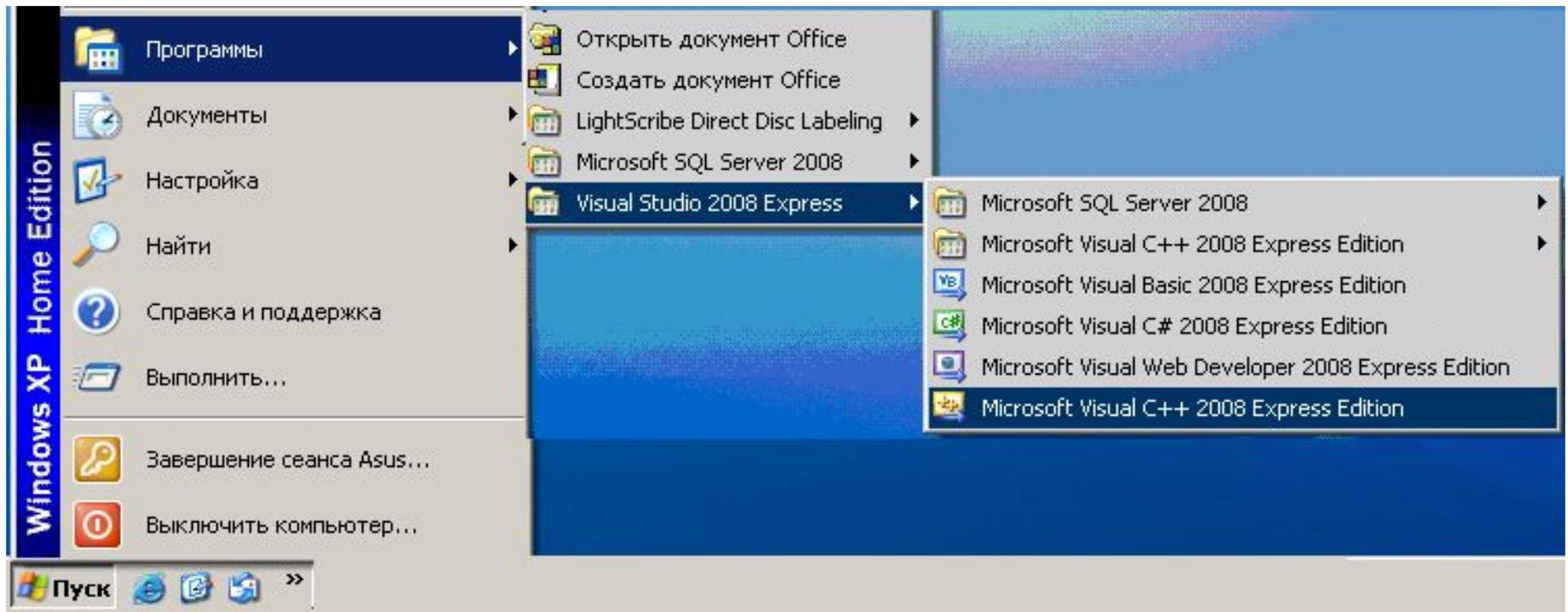
Астафьева_Н_О Срочный 14500 2006-11-15

Иванов_А_А До_востребования 23900 2005-05-05

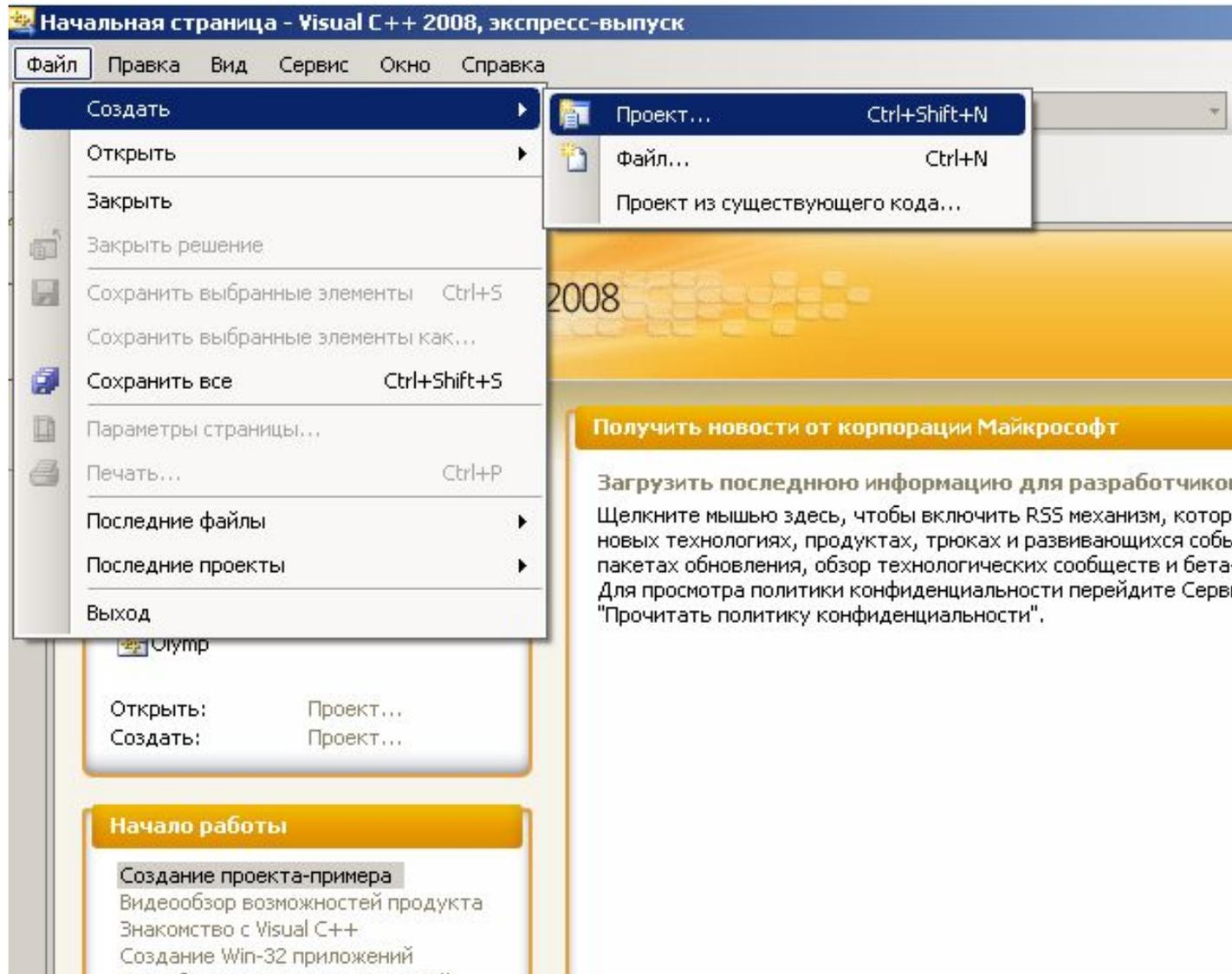
Лукьянова_А_В Срочный 8700 1998-09-08

1. Запускаем среду

Microsoft Visual C++ Express Edition

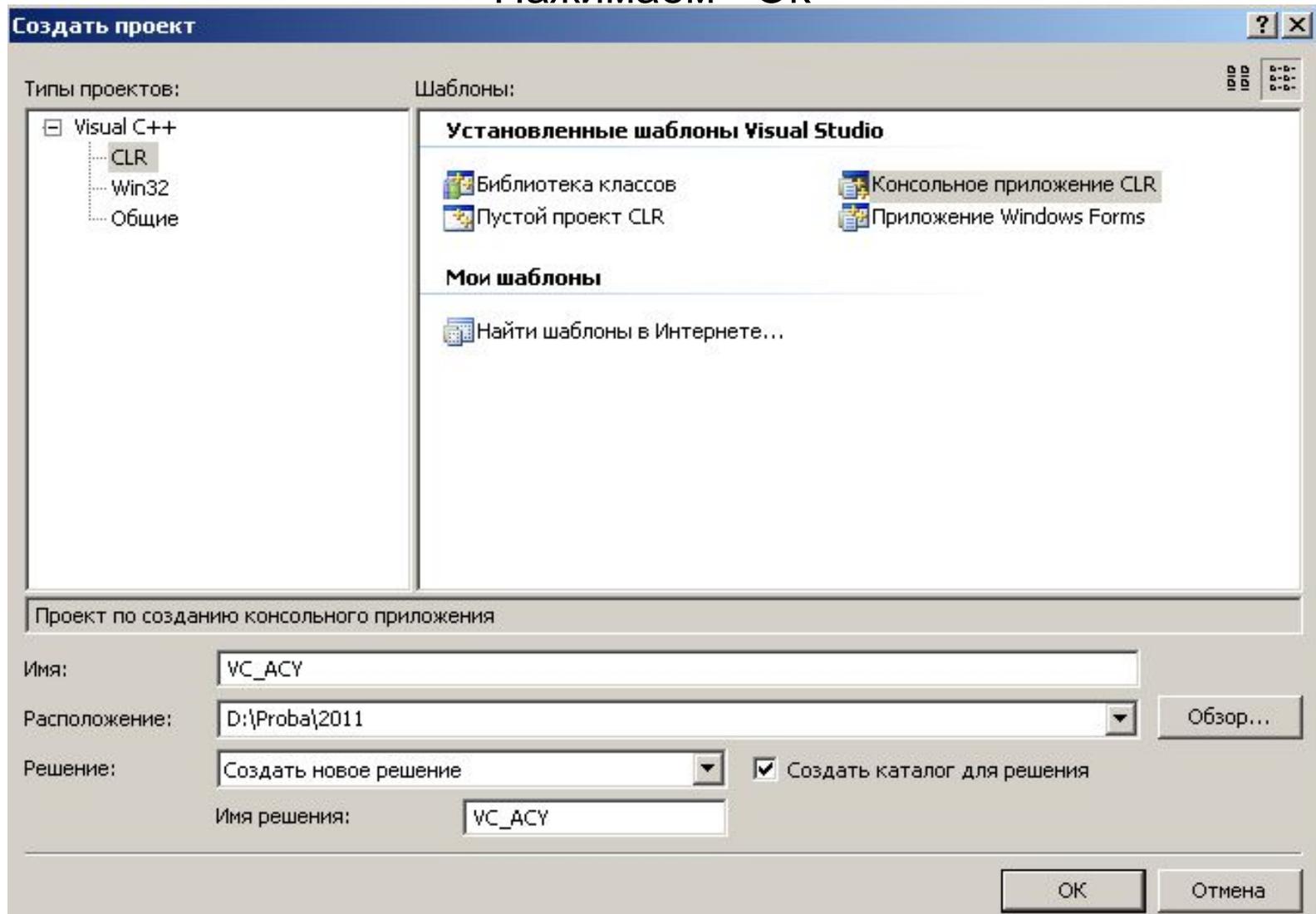


2. Создаем новый проект

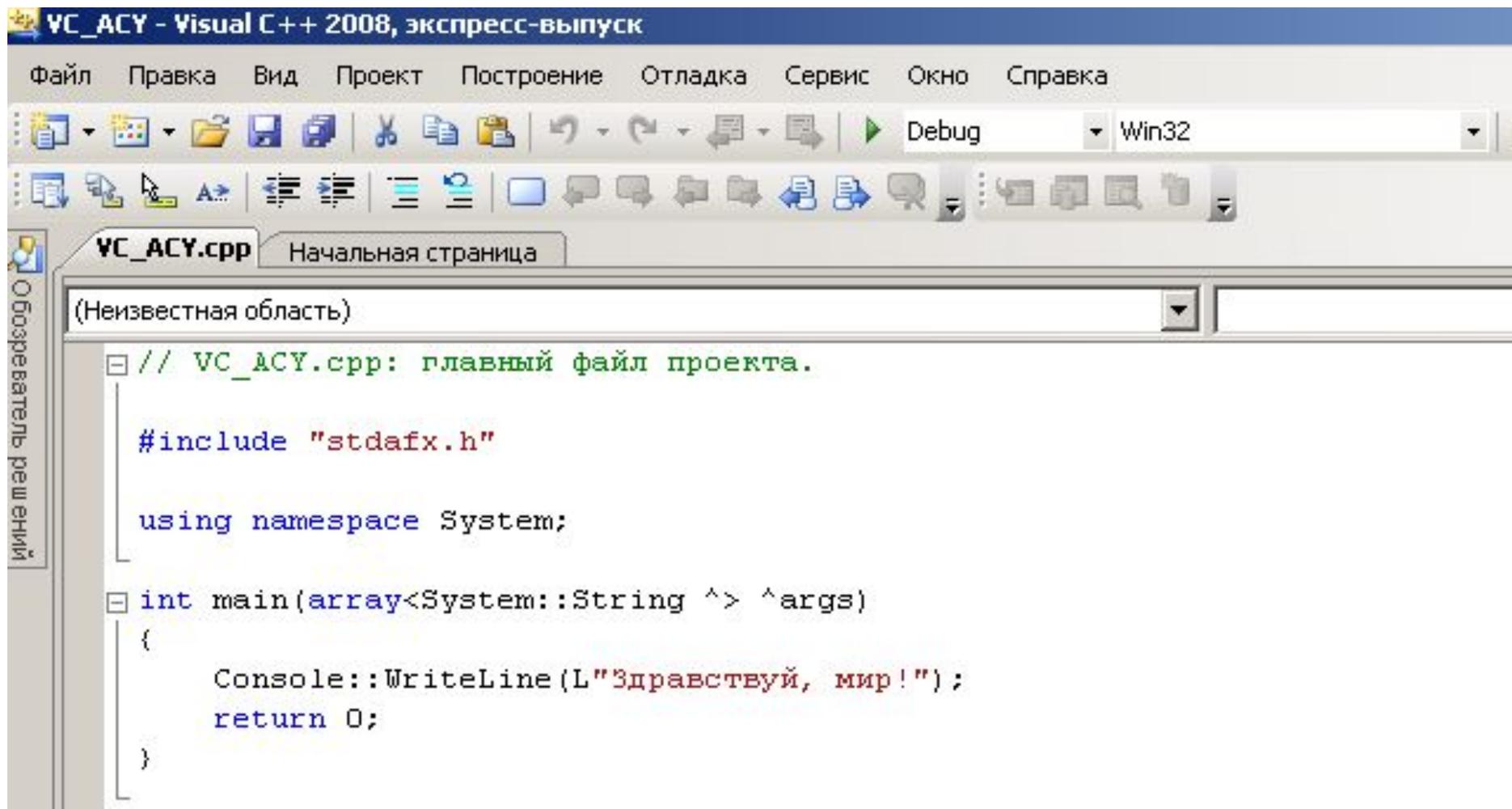


3. Задаем тип (Консольное приложение CLR), название (VC_ACY) и каталог для размещения проекта.

Нажимаем <Ok>



В результате мастер создает шаблон, который мы будем дополнять текстом нашей программы.



The image shows a screenshot of the Visual Studio 2008 IDE. The title bar reads "VC_ACU - Visual C++ 2008, экспресс-выпуск". The menu bar includes "Файл", "Правка", "Вид", "Проект", "Построение", "Отладка", "Сервис", "Окно", and "Справка". The toolbar contains various icons for file operations and debugging. The main editor window shows the file "VC_ACU.cpp" with the following code:

```
(Неизвестная область)

// VC_ACU.cpp: главный файл проекта.

#include "stdafx.h"

using namespace System;

int main(array<System::String ^> ^args)
{
    Console::WriteLine(L"Здравствуй, мир!");
    return 0;
}
```

VC_ACY.cpp: главный файл проекта

```
#include "stdafx.h"  
#include <stdio.h>  
#include <stdlib.h>  
#include <conio.h>  
#include <string.h>  
#include <malloc.h>  
#include <locale>  
  
using namespace std;  
using namespace System;  
using namespace System::IO;  
  
#define ENTER 13  
#define ESC 27  
#define UP 72  
#define DOWN 80
```

```

char dan[7][55]={
    "Какой вкладчик имеет на счету наибольшую сумму денег ?",
    "Какой вклад был открыт раньше всех ?           ",
    "Список \"Срочных\" вкладов с суммой свыше 10 000 р.  ",
    "Алфавитный список всех вкладчиков                ",
    "Количество вкладов \"До востребования\"             ",
    "Диаграмма. Процентное соотношение вложенных денег ",
    "Выход                                             "
};
char BlankLine[ ]="                                     ";
// BlankLine состоит из 56 пробелов
// или так: char BlankLine[80]; memset (BlankLine,' ',56);
BlankLine[56]=0;
int NC;

```

```
struct z {  
    char name[20];  
    char vid[20];  
    long summa;  
    char data[11];  
};
```

```
struct sp {  
    char fio[20];  
    long summa;  
    struct sp* sled;  
    } *spisok;
```

```
int menu(int) ;
void maxim(struct z*) ;
void first(struct z*) ;
void text_data(char *,char *) ;
void kolvo(struct z *) ;
void alfalist(struct z*) ;
void vstavka(struct z*,char*) ;
void listing(struct z*) ;
void diagram(struct z*) ;
```

```
int main(array<System::String ^> ^args)
{
int i,n;
FILE *in;
struct z *clients;
    setlocale(LC_CTYPE,"Russian");
    Console::CursorVisible::set(false);
    Console::BufferHeight=Console::WindowHeight;
    Console::BufferWidth=Console::WindowWidth;
    if((in=fopen("Vclad.dat","r"))==NULL)
    {
        printf("\nФайл Vclad.dat не открыт!");
        getch(); exit(1);
    }
}
```

// Ввод из файла и контрольная печать

```
fscanf(in, "%d", &NC) ;
clients=(struct z*)malloc(NC*sizeof(struct z)) ;

for (i=0 ; i<NC ; i++)
    fscanf(in, "%s%s%ld%s", clients[i].name,
           clients[i].vid, &clients[i].summa,
           clients[i].data) ;

for (i=0 ; i<NC ; i++)
    printf("\n%-20s  %-20s  %7ld  %s",
clients[i].name,
           clients[i].vid, clients[i].summa,
           clients[i].data) ;
getch() ;
```

```
while(1)
{
    Console::ForegroundColor=ConsoleColor::Gray;
    Console::BackgroundColor=ConsoleColor::Black;
    Console::Clear();
    Console::ForegroundColor=ConsoleColor::Black;
    Console::BackgroundColor=ConsoleColor::Gray;
    Console::CursorLeft=10;
    Console::CursorTop=4;
    printf(BlankLine);

    for(i=0;i<7;i++)
    {
        Console::CursorLeft=10;
        Console::CursorTop=i+5;
        printf(" %s ",dan[i]);
    }
    Console::CursorLeft=10;
    Console::CursorTop=12;
    printf(BlankLine);
```

```
n=menu(7);
switch(n) {
    case 1: maxim(clients); break;
    case 2: first(clients); break;
    case 3: listing(clients); break;
    case 4: alfalist(clients); break;
    case 5: kolvo(clients); break;
    case 6: diagram(clients); break;
    case 7: exit(0);
}
} // конец while(1)...
return 0;
} // конец main()
```

// программирование выбора из меню

```
int menu(int n)
{
int y1=0,y2=n-1;
char c=1;
while (c!=ESC)
    {
    switch(c) {
        case DOWN: y2=y1; y1++; break;
        case UP: y2=y1; y1--; break;
        case ENTER: return y1+1;
        }
    if(y1>n-1){y2=n-1;y1=0;}
    if(y1<0) {y2=0;y1=n-1;}
    Console::ForegroundColor=ConsoleColor::White;
    Console::BackgroundColor=ConsoleColor::Blue;
    Console::CursorLeft=11;
    Console::CursorTop=y1+5;
    printf("%s",dan[y1]);
```

```
Console::ForegroundColor=ConsoleColor::Black;
Console::BackgroundColor=ConsoleColor::Gray;
Console::CursorLeft=11;
Console::CursorTop=y2+5;
printf("%s",dan[y2]);
c=getch();
} // конец while(c!=ESC)...
exit(0);
}
```

// функция поиска максимального размера вклада

```
void maxim(struct z* client)
{
int i=0; struct z best;
strcpy(best.name,client[0].name);
best.summa=client[0].summa;
for(i=1;i<NC;i++)
    if (client[i].summa>best.summa)
        {
            strcpy(best.name,client[i].name);
            best.summa=client[i].summa;
        }
Console::ForegroundColor=ConsoleColor::Yellow;
Console::BackgroundColor=ConsoleColor::Black;
Console::CursorLeft=10;
Console::CursorTop=15;
printf("Максимальный вклад %ld рублей",best.summa);
Console::CursorLeft=10;
Console::CursorTop=16;
printf("имеет вкладчик %s",best.name);
getch();
}
```

// функция преобразования даты
2011-02-23 (sd) => 23 февраля 2011 (s)

```
void text_data(char *s, char *sd)
{
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;
strcat(s, month[atoi(s0)-1]);
strcat(s, " ");
strncat(s, sd, 4);
return;
}
```

// функция поиска самой ранней даты вклада

```
void first(struct z* client)
{
int i;
char s[17];
struct z* best=client;
for(i=1;i<NC;i++)
if (strcmp(client[i].data,best->data)<0)
    best=&client[i];
    text_data(s,best->data);
```

```
Console::ForegroundColor=ConsoleColor::Yellow;
Console::BackgroundColor=ConsoleColor::Black;
Console::CursorLeft=10;
Console::CursorTop=15;
printf("Самый \"старый\" вклад %s на %ld р.",
        best->vid,best->summa);
Console::CursorLeft=10;
Console::CursorTop=16;
printf("имеет вкладчик %s",best->name);
Console::CursorLeft=10;
Console::CursorTop=17;
printf("Открыт %s ",s);
getch();
}
```

// функция подсчета вкладов «До востребования»

```
void kolvo(struct z *client)
{
    int i,k=0;
    for (i=0;i<NC;i++)
        if (strcmp(client[i].vid,"До_востребования")==0)
            k++;
    Console::ForegroundColor=ConsoleColor::Yellow;
    Console::BackgroundColor=ConsoleColor::Black;
    Console::CursorLeft=10;
    Console::CursorTop=15;
    printf("Вклад \"До востребования\");
    printf("Всего : %d",k);
    getch();
}
```

// Составление алфавитного списка вкладчиков

```
void alfalist(struct z* client)
{
    int i;
    struct sp* nt;
    Console::ForegroundColor=ConsoleColor::Black;
    Console::BackgroundColor=ConsoleColor::Gray;
    Console::Clear();
    if(!spisok)
        for(i=0;i<NC;i++)
            vstavka(client,client[i].name);

    Console::Clear();
    printf("\n  Алфавитный список вкладчиков СБ");
    printf("\n  =====\n");
    for(nt=spisok; nt!=0; nt=nt->sled)
        printf("\n %-20s %ld",nt->fio,nt->summa);
    getch();
}
```

// Вставка вкладчика в алфавитный список

```
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;
}
```

// Вывод алфавитного списка вкладчиков

```
void listing(struct z* client)
{
int i;
struct z* nt;
Console::ForegroundColor=ConsoleColor::Black;
Console::BackgroundColor=ConsoleColor::Gray;
Console::Clear();
printf("\n\r Список \"Срочных\" вкладов с суммой свыше 10 тыс.
рублей");
printf("\n\r
=====\  

n\r");
for(i=0,nt=client;i<NC;nt++,i++)
    if (nt->summa>10000 && strcmp(nt->vid,"Срочный")==0)
        printf("\n\r %-20s %ld p.",nt->name,nt->summa);
getch();
}
```

```
// Диаграмму строим на основе псевдографики
// Столбик формируется цветной полоской из N
// пробелов
```

```
void diagram(struct z *client)
{
    struct sp *nt;
    int len,i,NCOLOR;
    long sum = 0 ;
    char str1[20];
    char str2[20];
    System::ConsoleColor Color;
    Console::ForegroundColor=ConsoleColor::Black;
    Console::BackgroundColor=ConsoleColor::White;
    Console::Clear();
    for(i=0;i<NC;i++) sum = sum+client[i].summa ;
    if(!spisok)
        for(i=0;i<NC;i++)
            vstavka(client,client[i].name);
    Color=ConsoleColor::Black; NCOLOR=0;
```

```

for(nt=spisok,i=0; nt!=0; nt=nt->sled,i++)
{
    sprintf(str1,"%s",nt->fio);
    sprintf(str2,"%3.1f%%",(nt->summa*100./sum));
    Console::ForegroundColor=ConsoleColor::Black;
    Console::BackgroundColor= ConsoleColor::White;
    Console::CursorLeft=5;    Console::CursorTop=i+1;
    printf(str1);
    Console::CursorLeft=20;
    printf("%s",str2);
    Console::BackgroundColor=++Color; NColor++;
    Console::CursorLeft=30;
    for(len=0; len<nt->summa*100/sum; len++) printf(" ");
    if(NColor==14)
        { Color=ConsoleColor::Black; NColor=0; }
}
getch();
return ;
}

```

Запускаем на выполнение



```
D:\F\STUDENT\VC2008Express\VC_ACY\Debug\VC_ACY_Con.exe
```

Иванов_А_А	Срочный	15000	2002-09-23
Сидоров_И_А	Юбилейный	7150	1991-03-08
Петров_В_Н	Пенсионный	38876	1999-12-16
Сидоров_И_А	Сберегательный	12860	2008-06-23
Юдин_О_В	Особый	25000	2006-12-13
Федоров_М_К	До_востребования	8905	2005-11-14
Иванов_А_А	Пенсионный	75980	2003-08-04
Сидоров_И_А	Особый	34760	2004-02-07
Андреев_В_Н	До_востребования	13908	2007-04-09
Астафьева_Н_О	Срочный	14500	2006-11-15
Иванов_А_А	До_востребования	23900	2005-05-05
Лукьянова_А_В	Срочный	8700	1998-09-08

D:\F\STUDENT\VC2008Express\VC_ACY\Debug\VC_ACY_Con.exe



Какой вкладчик имеет на счету наибольшую сумму денег ?

Какой вклад был открыт раньше всех ?

Список "Срочных" вкладов с суммой свыше 10 000 р.

Алфавитный список всех вкладчиков

Количество вкладов "До востребования"

Диаграмма. Процентное соотношение вложенных денег

Выход

Самый "старый" вклад Юбилейный на 7150 р.

имеет вкладчик Сидоров_И_А

Открыт 08 марта 1991

D:\F\STUDENT\VC2008Express\VC_ASY\Debug\VC_ASY_Con.exe



Алфавитный список вкладчиков СБ

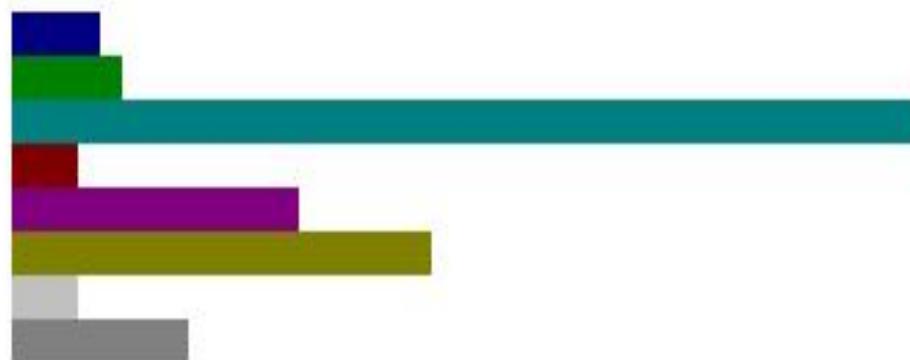
=====

Андреев_В_Н	13908
Астафьева_Н_О	14500
Иванов_А_А	114880
Лукьянова_А_В	8700
Петров_В_Н	38876
Сидоров_И_А	54770
Федоров_М_К	8905
Юдин_О_В	25000

C:\ d:\F\STUDENT\VC2008Express\VC_ACY\Debug\VC_ACY_Con.exe



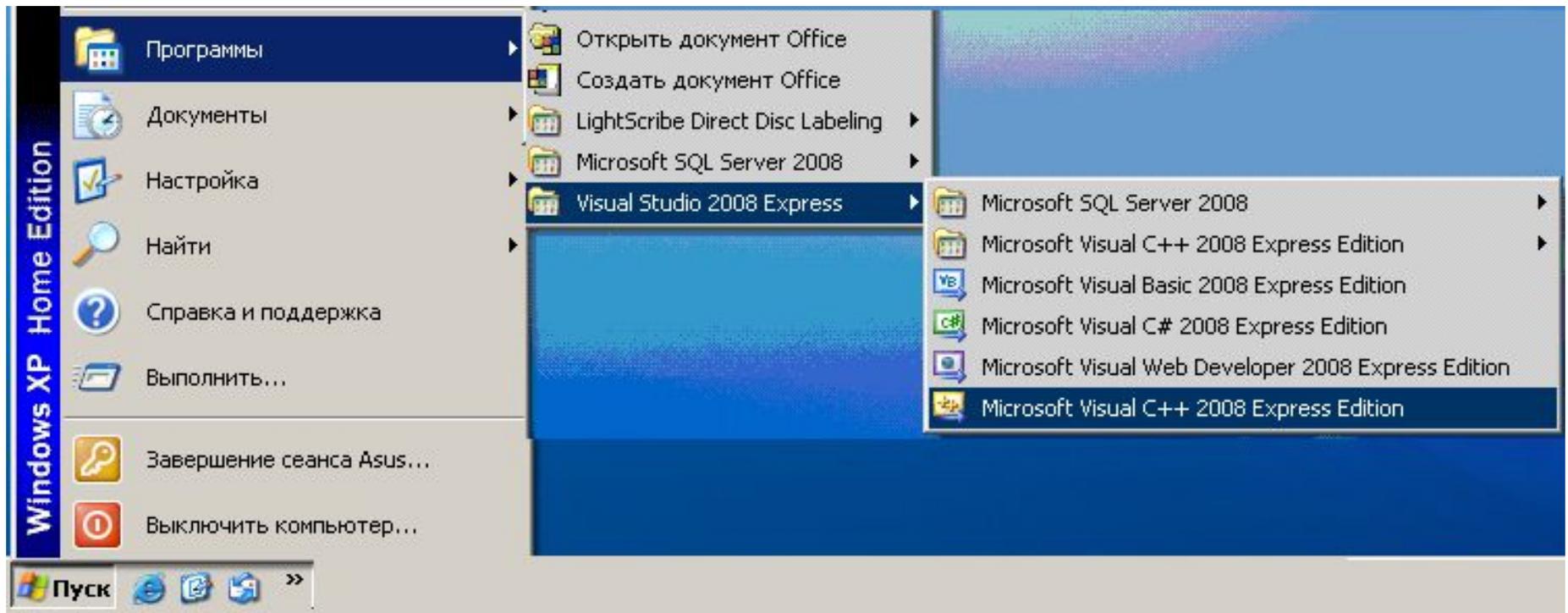
Андреев_В_Н	5.0%
Астафьева_Н_О	5.2%
Иванов_А_А	41.1%
Лукьянова_А_В	3.1%
Петров_В_Н	13.9%
Сидоров_И_А	19.6%
Федоров_М_К	3.2%
Юдин_О_В	8.9%



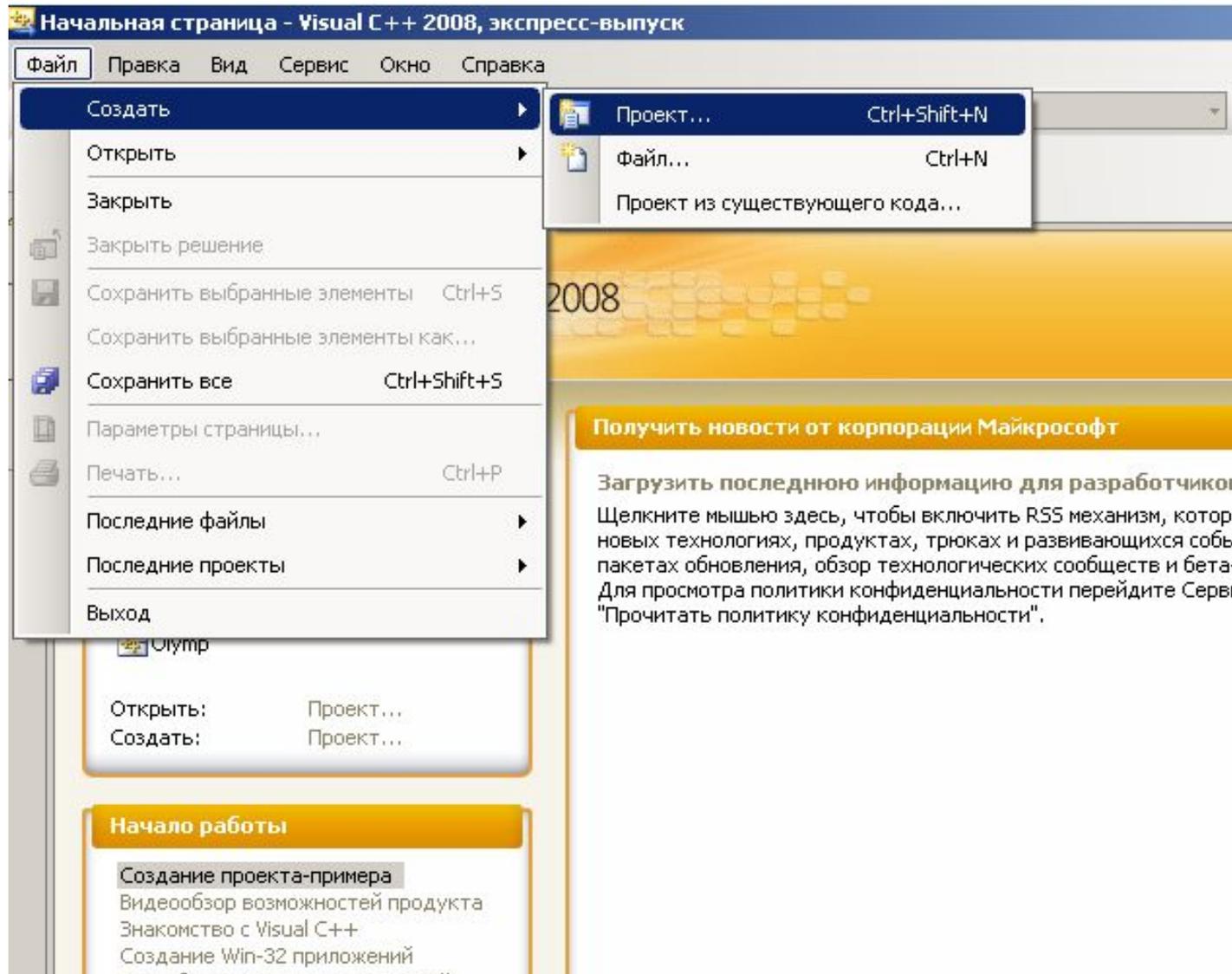
Графический режим
в среде
Microsoft Visual Studio 2008
Express

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>

Создать проект

Типы проектов:

- Visual C++
 - CLR
 - Win32
 - Общие

Шаблоны:

Установленные шаблоны Visual Studio

- Библиотека классов
- Пустой проект CLR
- Консольное приложение CLR
- Приложение Windows Forms

Мои шаблоны

- Найти шаблоны в Интернете...

Проект приложения с пользовательским интерфейсом Windows

Имя:

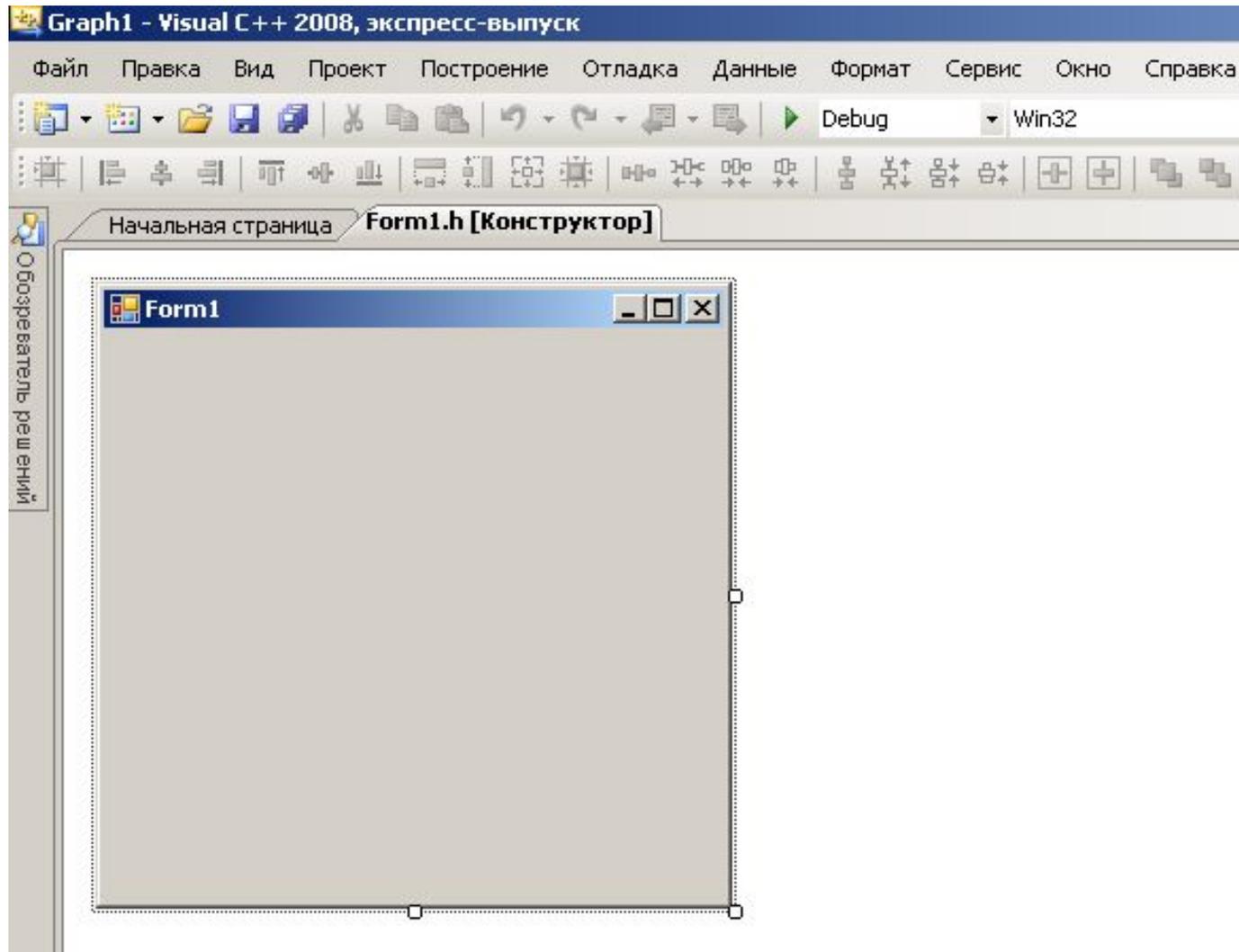
Расположение: Обзор...

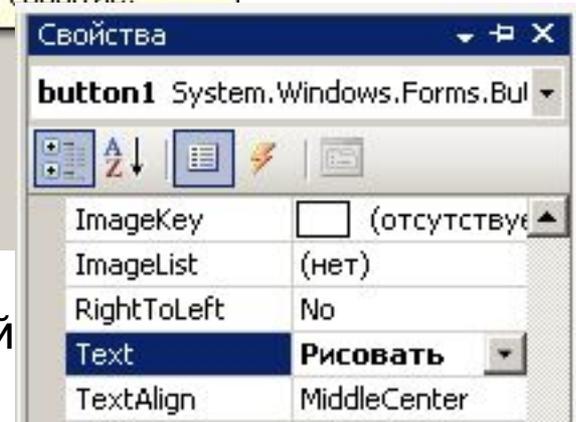
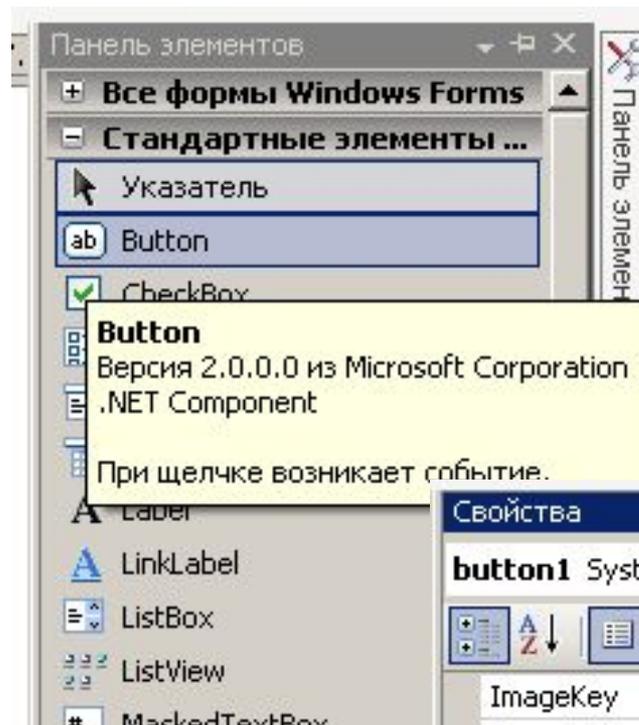
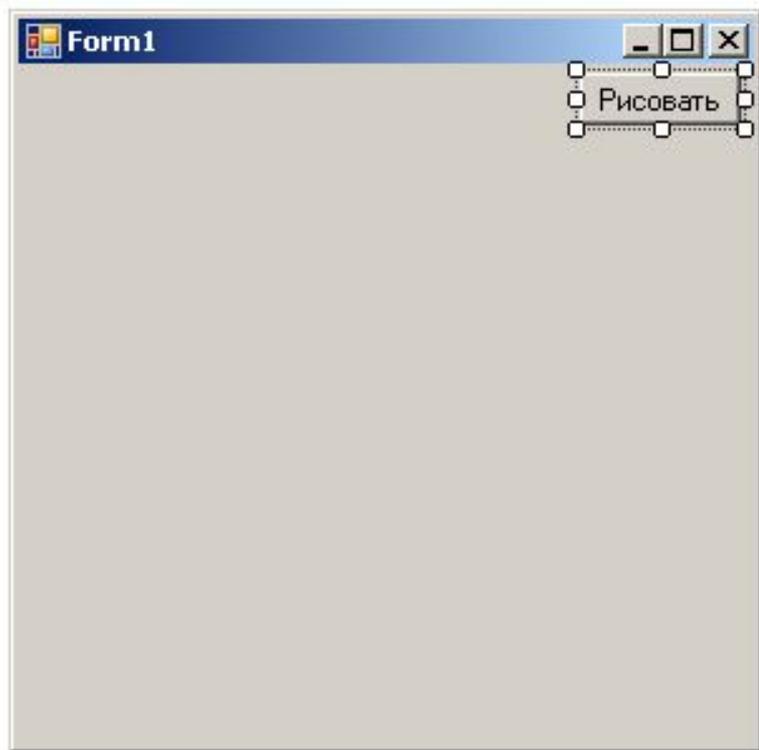
Решение: Создать каталог для решения

Имя решения:

OK Отмена

4. Мастер создает заготовку формы



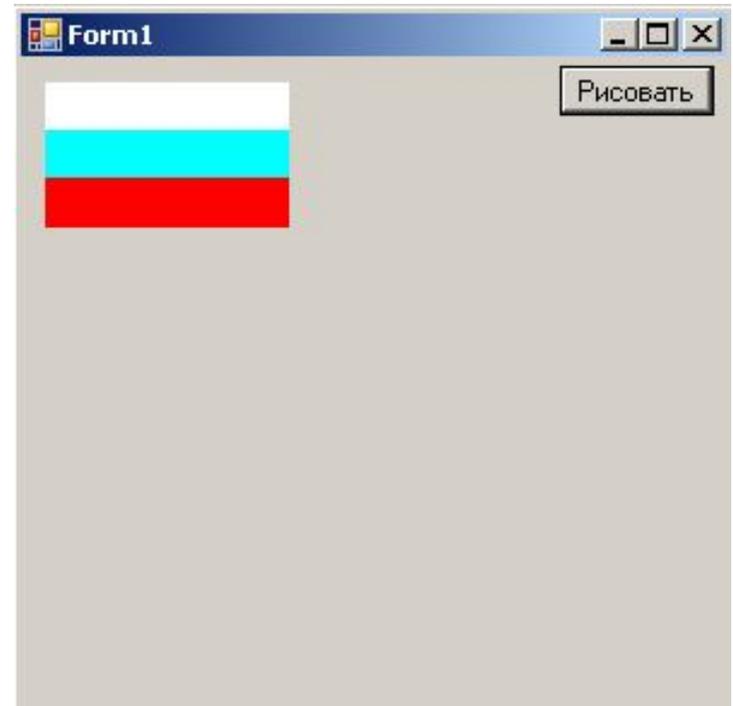


На форме размещаем командную кнопку, для которой устанавливаем свойство Text="Рисовать"

Далее делаем двойной щелчок по кнопке и в раскрывшемся окне набираем следующий код

```
private: System::Void button1_Click(  
    System::Object^ sender, System::EventArgs^ e)
```

```
{  
    Graphics ^g = this -> CreateGraphics();  
    g->Clear(this->BackColor); // Фон закрашиваем стандартным цветом  
    Pen ^myPen= gnew Pen(Color::White,1);  
    SolidBrush ^myBrush = gnew SolidBrush(Color::White);  
    g->DrawRectangle(myPen,10,10,100,20);  
    g->FillRectangle(myBrush,10,10,100,20);  
    myPen->Color::set(Color::Cyan);  
    myBrush->Color::set(Color::Cyan);  
    g->DrawRectangle(myPen,10,30,100,20);  
    g->FillRectangle(myBrush,10,30,100,20);  
    myPen->Color::set(Color::Red);  
    myBrush->Color::set(Color::Red);  
    g->DrawRectangle(myPen,10,50,100,20);  
    g->FillRectangle(myBrush,10,50,100,20);  
}
```

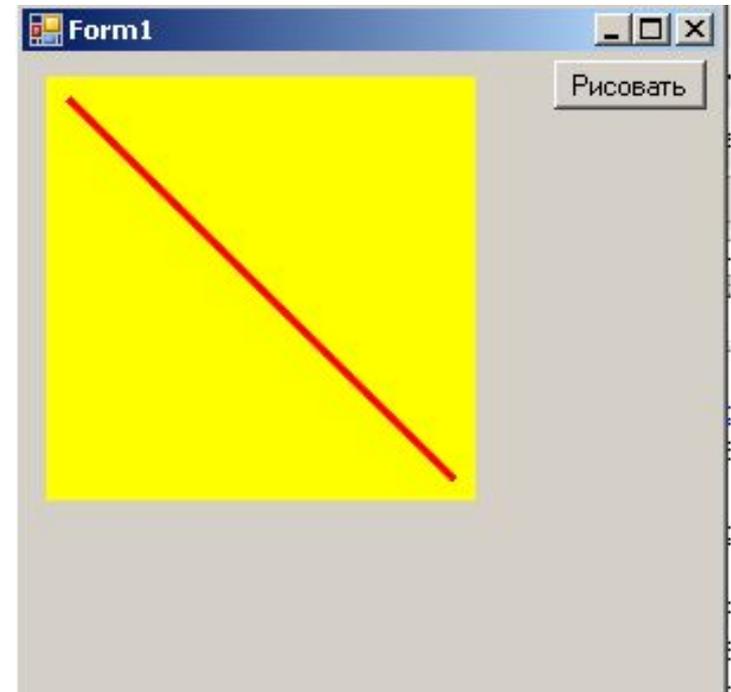


Изображение сохраним в файле IMAGE2.BMP

```
private: System::Void button1_Click(  
    System::Object^ sender, System::EventArgs^ e)  
{  
    Graphics ^g = this -> CreateGraphics();  
    g->Clear(this->BackColor);  
    Pen ^myPen= gcnew Pen(System::Drawing::Color::Red,3);  
    Bitmap ^Image2=gcnew Bitmap(200,200);  
    Graphics ^g2=Graphics::FromImage(Image2);  
    g2->Clear(Color::Yellow);  
    g2->DrawLine(myPen,10,10,190,190);  
    Image2->Save("Image2.bmp");  
    g->DrawImage(Image2,10,10,180,180);  
}
```

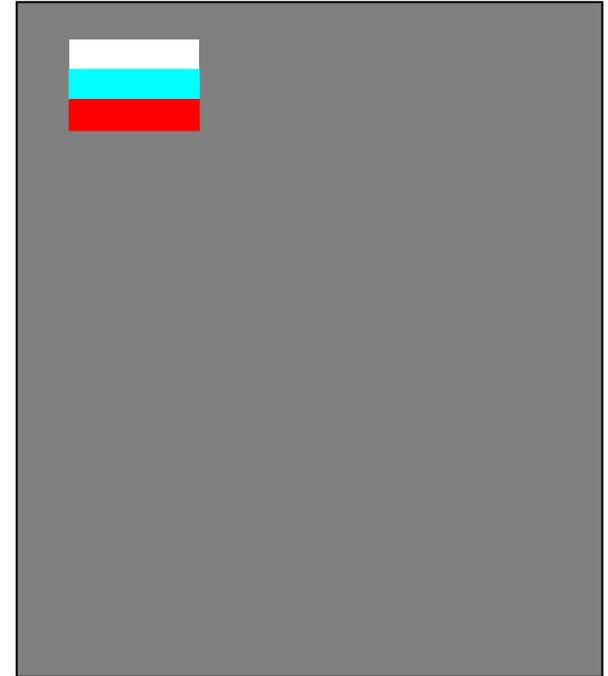
А вот так картинку из файла можно воспроизвести в произвольной точке (x,y)

```
Bitmap ^Image1=gcnew Bitmap("Image2.bmp");  
g->DrawImage(Image1,x,y,180,180);
```

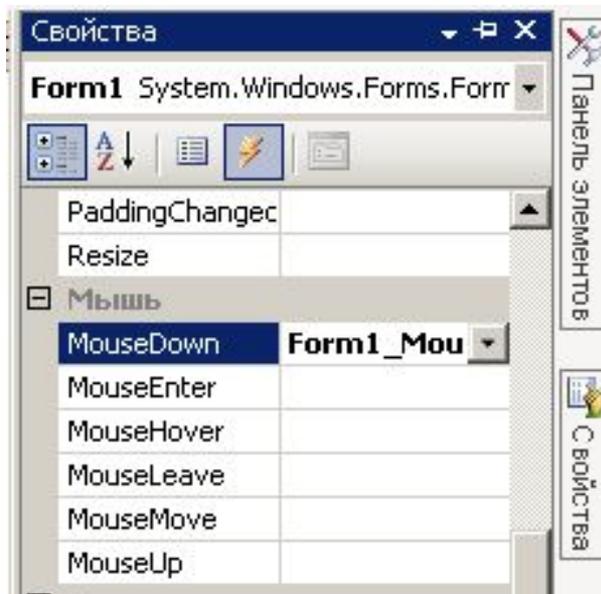


А так можно воспроизвести движение изображения сверху вниз

```
Graphics ^g = this -> CreateGraphics();  
Bitmap ^Image1=gcnnew Bitmap("Image1.bmp");  
g->DrawImage(Image1,10,10,100,50);  
  
int y;  
for(y=10;y<200;y++)  
{  
    g->Clear(System::Drawing::Color::DarkGray);  
    g->DrawImage(Image1,10,y,100,50);  
    _sleep(10); // прототип в <stdlib.h>  
}
```

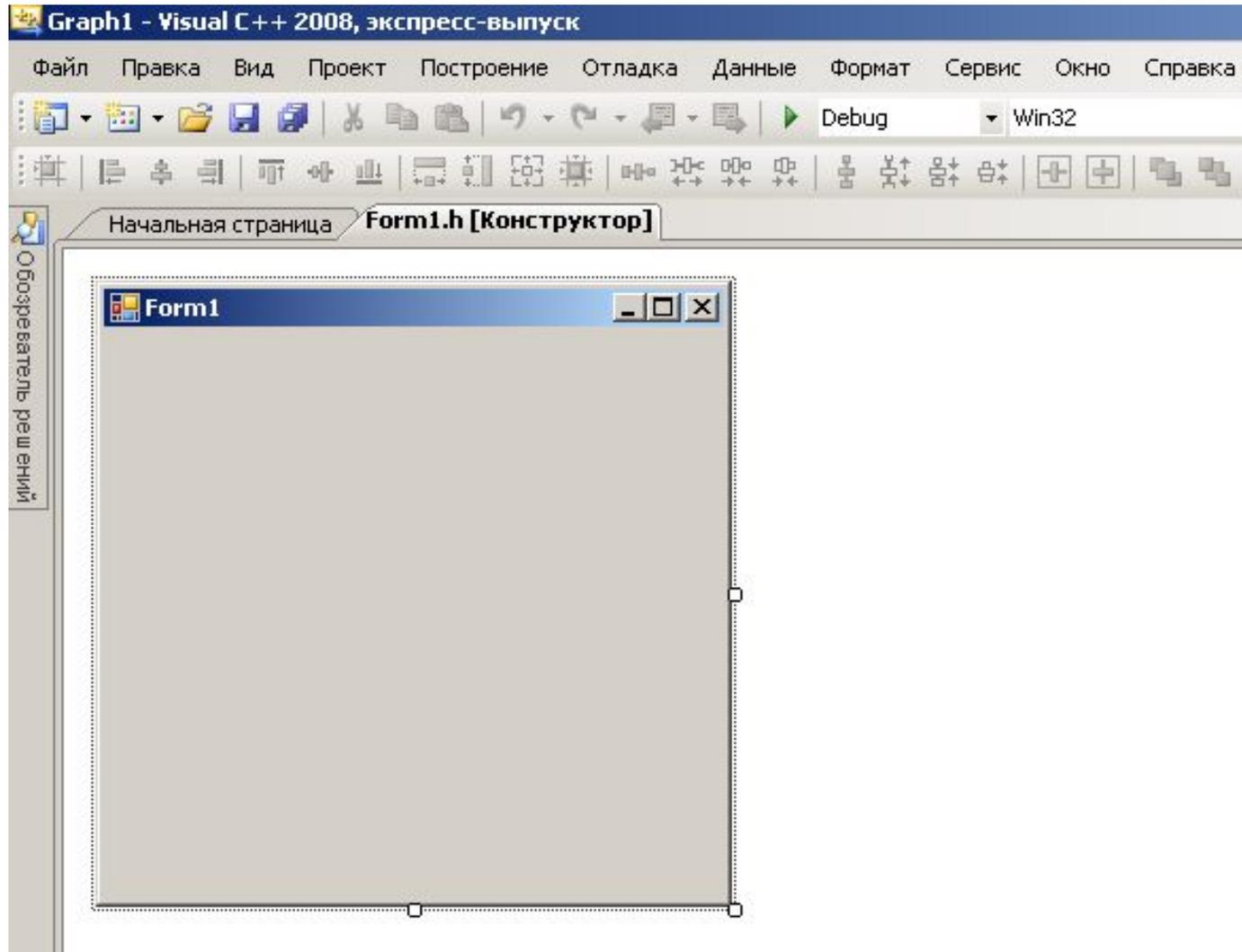


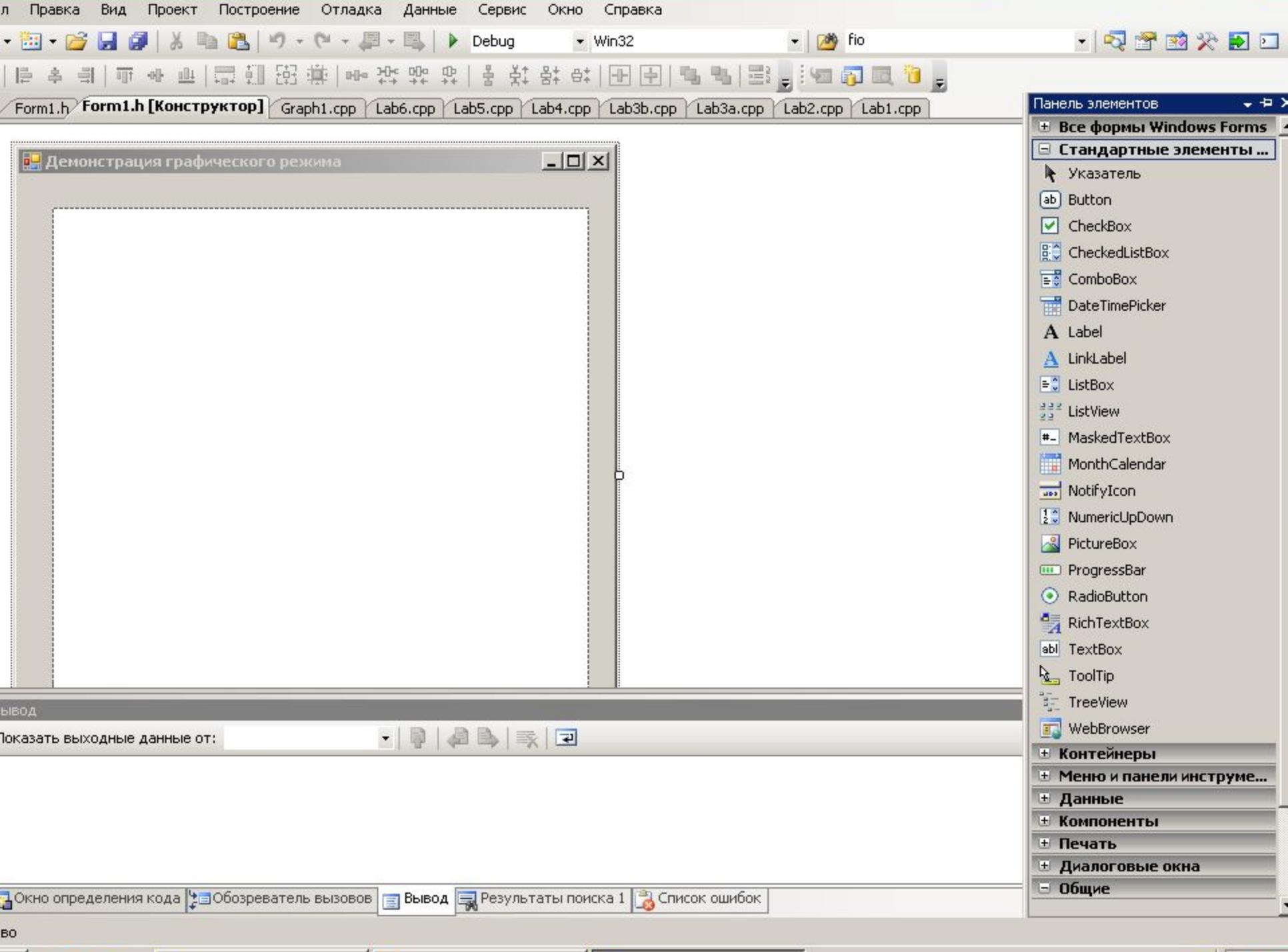
*Кроме того, можно рисовать в той точке, где пользователь щелкнул
МЫШЬЮ*



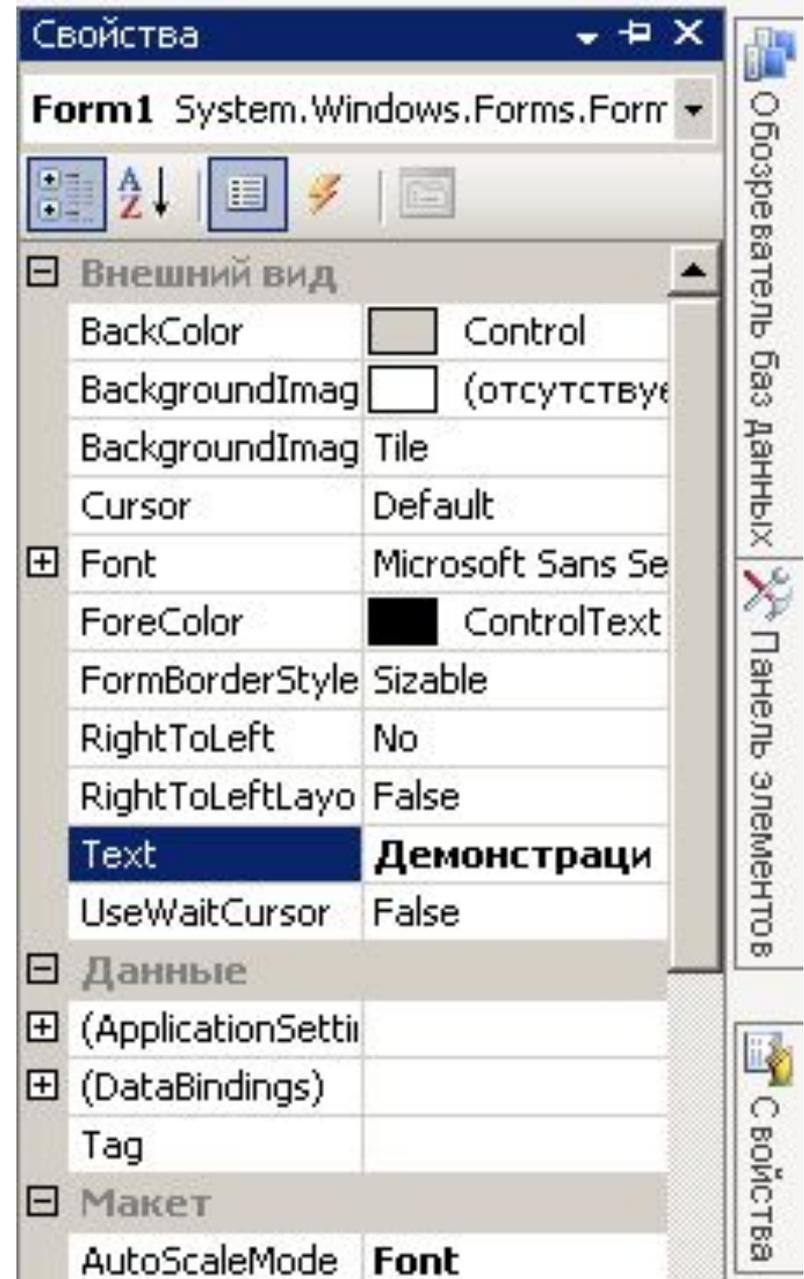
```
private: System::Void Form1_MouseDown(  
    System::Object^ sender,  
    System::Windows::Forms::MouseEventArgs^ e)  
{  
    Graphics ^g = this -> CreateGraphics();  
    g->Clear(this->BackColor);  
    g->DrawRectangle(Pens::Red,e->X,e->Y,100,20);  
    // Pens::Цвет наряду с Brushes::Цвет можно применять вместо ссылок на  
    // Pen myPen и Brush myBrush, соответственно.  
    // Правда, толщина линии пера тогда может быть равна только 1.  
}
```

4. Мастер создает заготовку формы

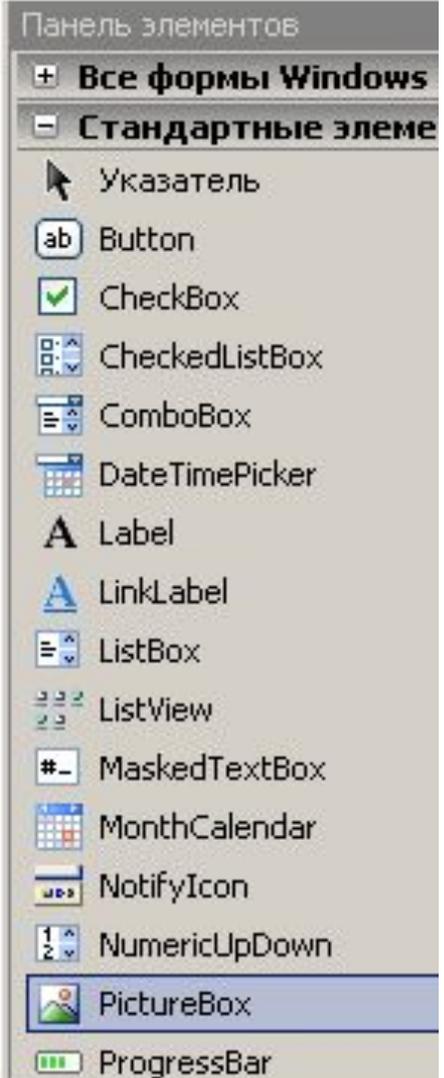
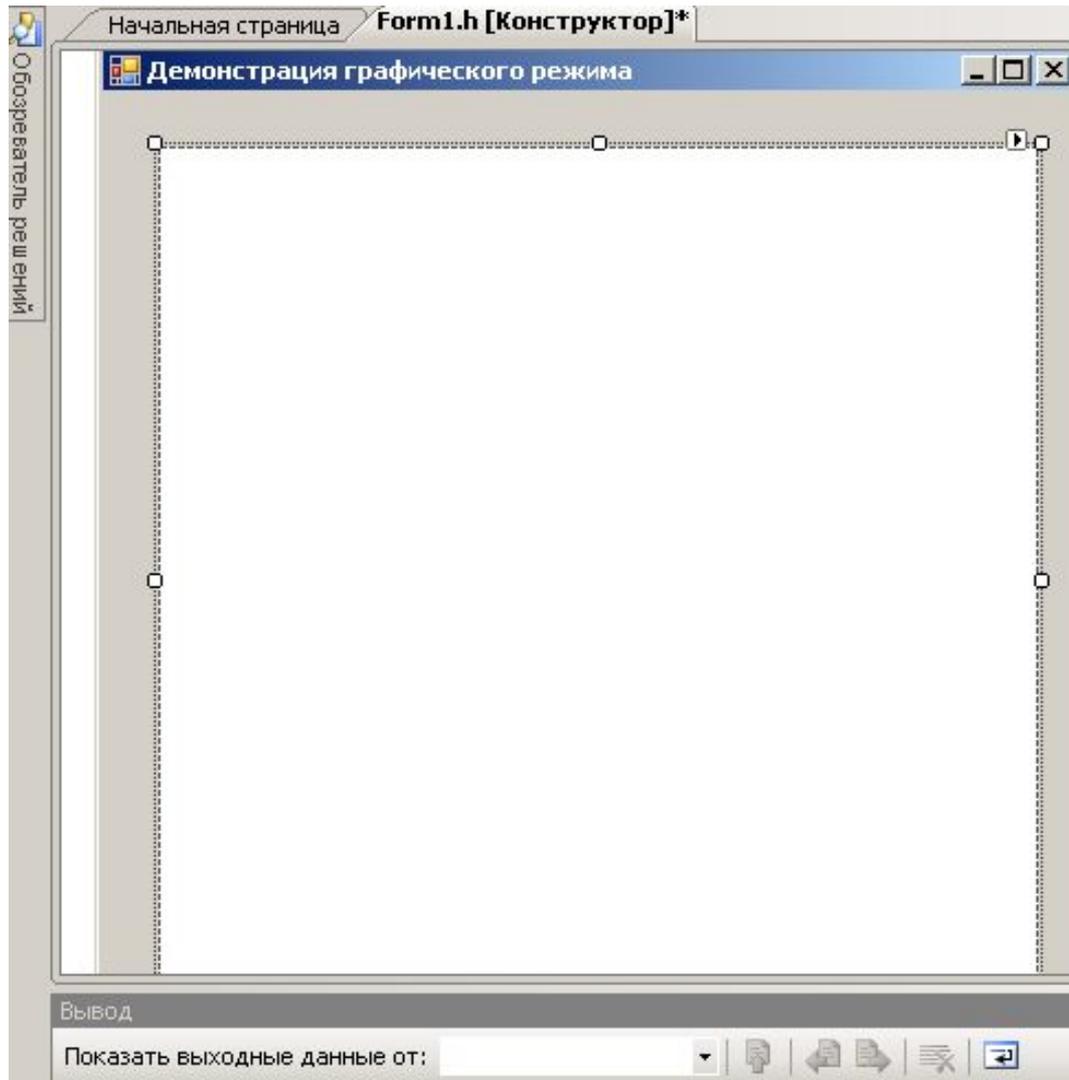




- а) Изменяем свойство Text:
Демонстрация графического режима. Данный текст тут же появится в заголовке окна формы.
- б) Изменяем размер формы —
Size = 400;500 и положение на экране
StartPosition = CenterScreen
(форма будет запускаться в центре экрана).
- в) Размещаем на форме компонент pictureBox, который перетаскиваем из панели элементов на нашу форму.



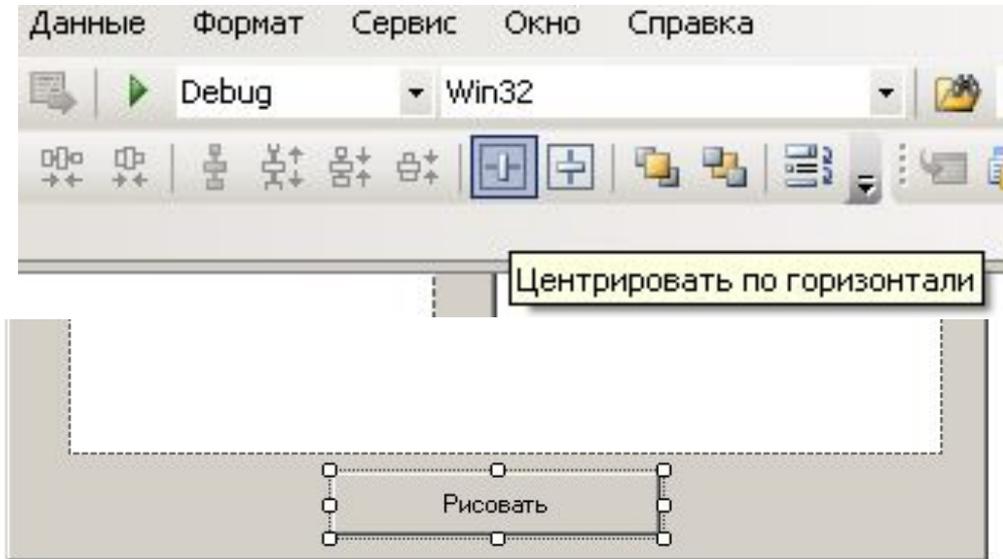
в) Размещаем на форме компонент `pictureBox`, который перетаскиваем из панели элементов на нашу форму.



PictureBox
Версия 2.0.0.0 из Microsoft Corp
.NET Component
Определяет поддержку для и
транзакции.

г) Устанавливаем параметры pictureBox1 (так этот элемент теперь именуется в нашей программе):

Size = 400;500, Location = 25;25, BackColor=ActiveCaptionText.



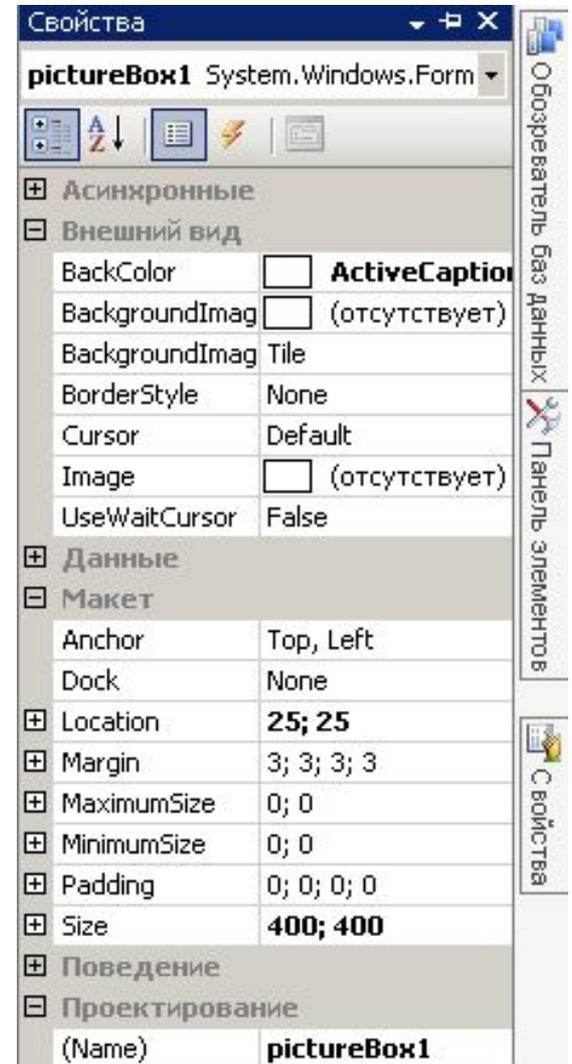
Перетаскиваем с панели элементов командную кнопку button1, для которой задаем такие свойства:

Text=Рисовать,

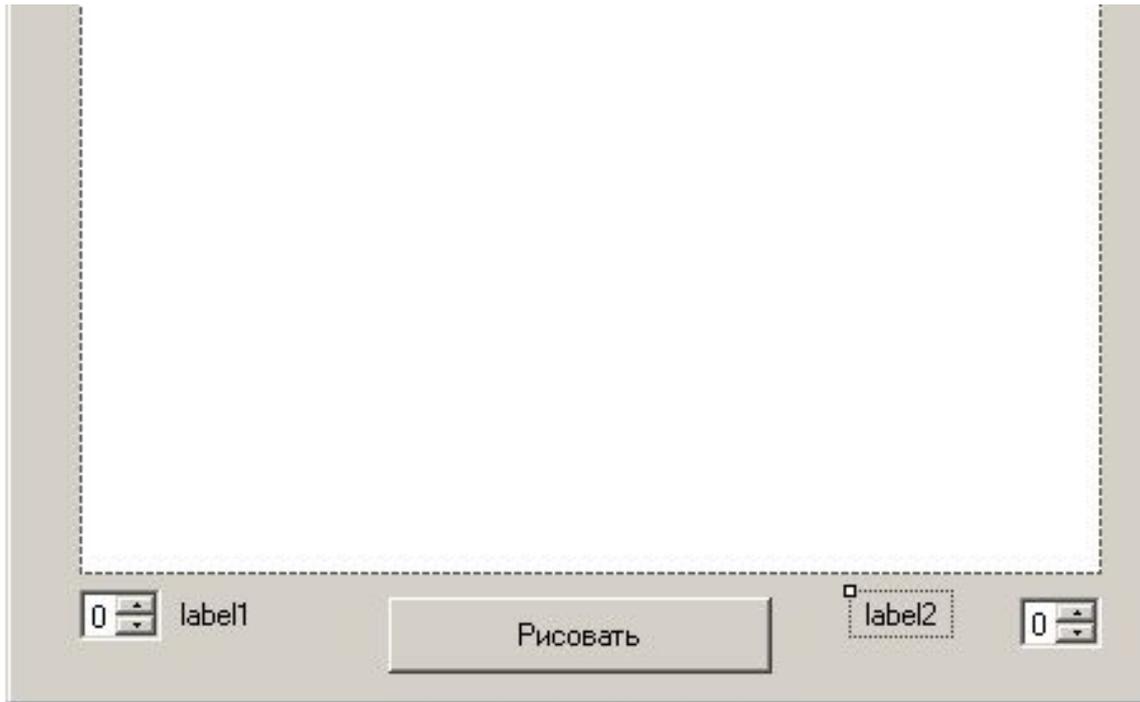
Size = 150;30,

Location = 146;434,

для расположения точно по центру можно воспользоваться соответствующей кнопкой меню

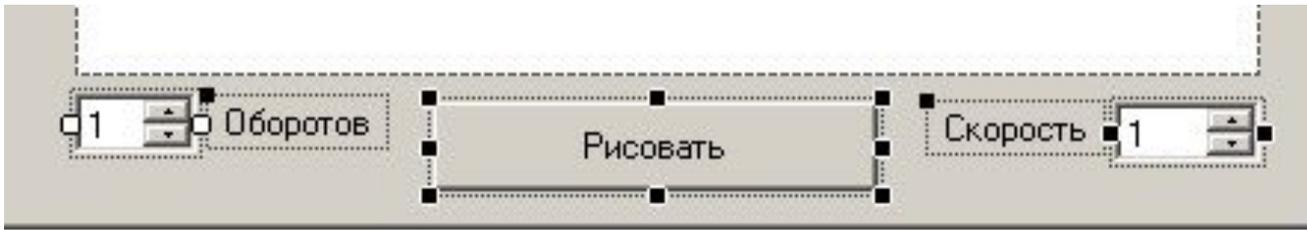


д) Разместим дополнительно на форме по два элемента Label и NumericUpDown

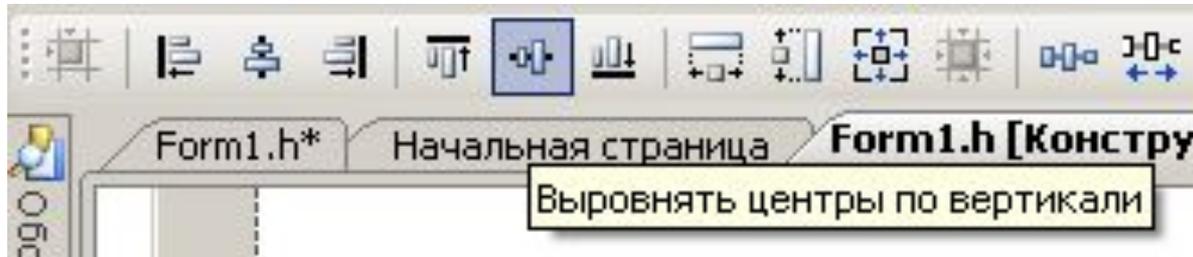


для которых установим следующие свойства:
label1: Text=Оборотов
label2: Text=Скорость
numericUpDown1: Size = 150;30, Minimum=1, Maximum=10
numericUpDown2: Size = 150;30, Minimum=-5, Maximum=5

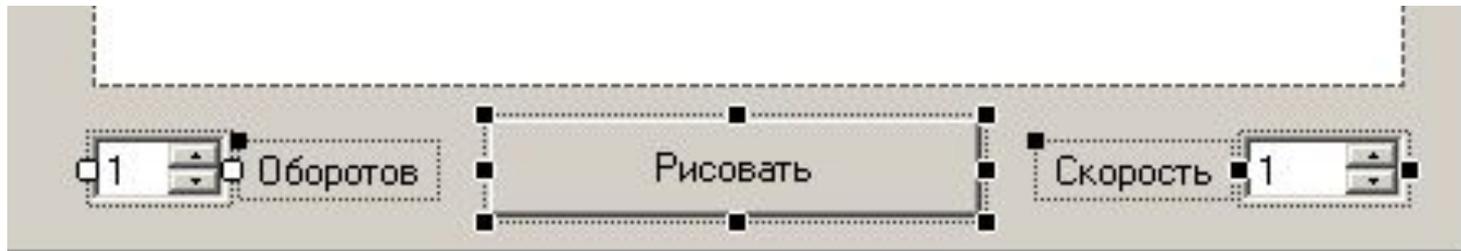
Выделим (используя SHIFT) группу элементов



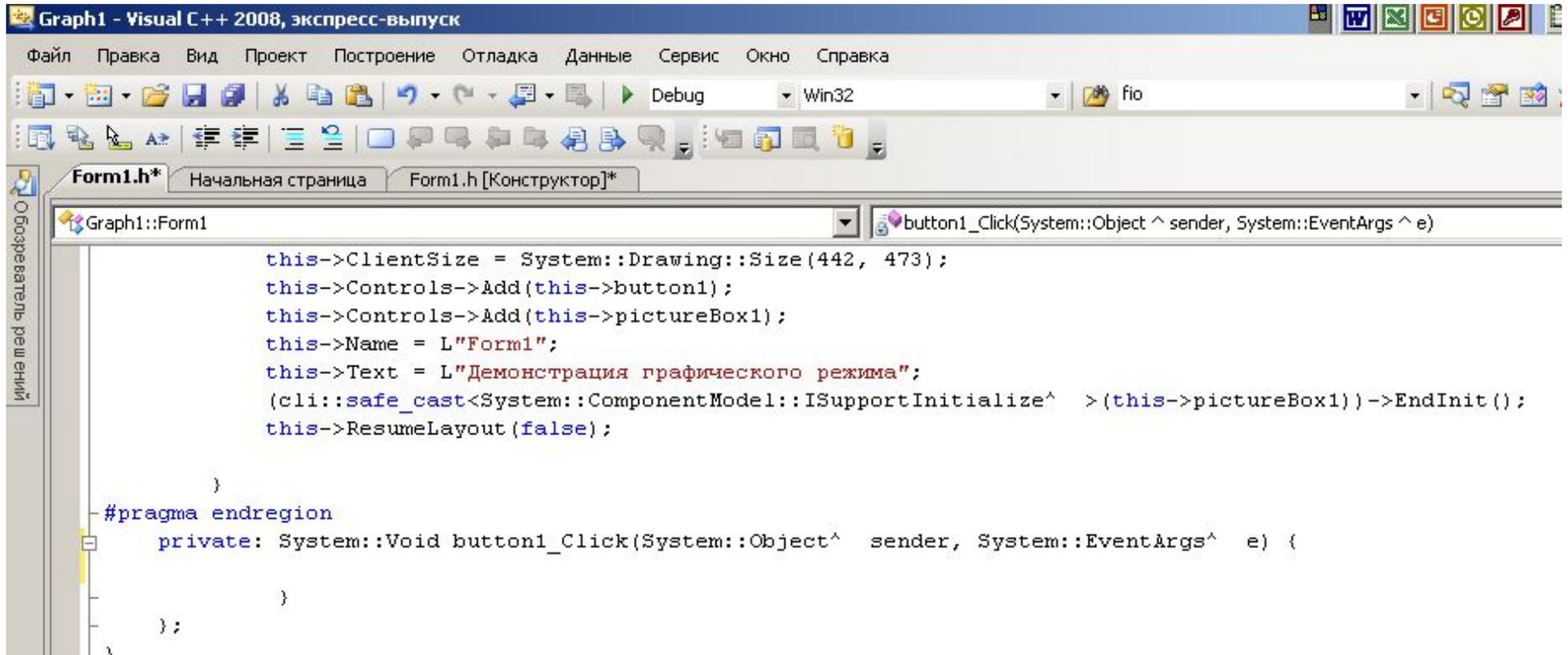
применим для неё выравнивание центров по вертикали



Получаем геометрически правильное расположение элементов на форме



е) Делаем двойной щелчок по кнопке рисовать и в открывшемся окне, набиваем необходимый программный код.



The screenshot shows the Visual C++ 2008 Express Edition IDE. The title bar reads "Graph1 - Visual C++ 2008, экспресс-выпуск". The menu bar includes "Файл", "Правка", "Вид", "Проект", "Построение", "Отладка", "Данные", "Сервис", "Окно", and "Справка". The toolbar shows various icons for file operations, editing, and debugging. The "Debug" dropdown is set to "Win32". The active window is "Form1.h*" with a sub-tab "Form1.h [Конструктор]*". The code editor shows the following code:

```
Graph1::Form1
button1_Click(System::Object ^ sender, System::EventArgs ^ e)

    this->ClientSize = System::Drawing::Size(442, 473);
    this->Controls->Add(this->button1);
    this->Controls->Add(this->pictureBox1);
    this->Name = L"Form1";
    this->Text = L"Демонстрация графического режима";
    (cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this->pictureBox1))->EndInit();
    this->ResumeLayout(false);

}

#pragma endregion
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {

}

};
```

```

private: System::Void button1_Click(
System::Object^ sender, System::EventArgs^ e)
{
int N=3;           // N=3 задает треугольник
PointF point1=PointF(100,150); // Задаем координаты
PointF point2=PointF(100,250); // вершин
PointF point3=PointF(300,250); // треугольника
int xC=200,yC=200; // Середина гипотенузы
int i,Angle,k,Tempo; // Локальные
float x,y,X,Y;     // переменные
// Создаем ссылку на графическую область
Graphics ^g=pictureBox1->CreateGraphics();
g->Clear(System::Drawing::Color::White);
Pen ^myPen= gcnew Pen(System::Drawing::Color::Black,3);
// При создании пера задаем цвет(BLACK) и толщину линии (3)
array <PointF> ^PointsXY={point1,point2,point3};
// Массив понадобится для функции DrawPolygon()
g->DrawPolygon(myPen,PointsXY);

```

// Число оборотов

```
k=(int)this->numericUpDown1->Value;
```

// Скорость вращения, отрицательное значение Tempo

// означает вращение против часовой стрелки

```
Tempo=(int)this->numericUpDown2->Value;
```

```
if(!Tempo)
```

```
{
```

```
MessageBox::Show("Скорость не должна равняться нулю!",  
"Внимание!");
```

```
return;
```

```
}
```

```
do{
```

```
for(Angle=0;Angle<360;Angle+=Math::Abs(Tempo))
```

```
{
```

//Белое перо стирает изображение треугольника

```
myPen->Color::set(Color::White);
```

```
g->DrawPolygon(myPen,PointsXY);
```

```

for(i=0;i<N;i++)
{
    x=PointsXY[i].X; y=PointsXY[i].Y;
// Функция turn поворачивает каждую вершину (x,y) на угол Tempo
// вокруг центра (xC,yc)
    turn(xC,yC,-Tempo,x,y,&X,&Y);
    PointsXY[i].X=X; PointsXY[i].Y=Y;
}
//Черное перо рисует треугольник в новом (после поворота) положении
myPen->Color::set(Color::Black);
g->DrawPolygon(myPen,PointsXY);
// Можно изменить толщину линии:
// myPen->Width::set(3);
myPen->Color::set(Color::Red);
g->DrawEllipse(myPen,xC-3,yC-3,7,7);
_sleep(50); //задержка на 50 мс, прототип в <stdlib.h>
}
} while (--k);
this->numericUpDown1->Value=1;
}

```

Прокручиваем экран в самое начало и после

```
#pragma once
```

вставляем такой текст:

```
#include <stdlib.h>
```

```
#include <math.h>
```

```
void turn(int x0, int y0, int alfa, float x, float y, float* x1, float* y1)
```

```
{
```

```
float t;
```

```
t=alfa*3.14/180;
```

```
*x1=x0+(x-x0)*cos(t)+(y-y0)*sin(t);
```

```
*y1=y0-(x-x0)*sin(t)+(y-y0)*cos(t);
```

```
return;
```

```
}
```

```
#pragma once

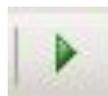
#include <stdlib.h>
#include <math.h>

void turn(int x0, int y0, int alfa, float x, float y, float* x1, float* y1)
{
    float t;
    t=alfa*3.14/180;
    *x1=x0+(x-x0)*cos(t)+(y-y0)*sin(t);
    *y1=y0-(x-x0)*sin(t)+(y-y0)*cos(t);
    return;
}

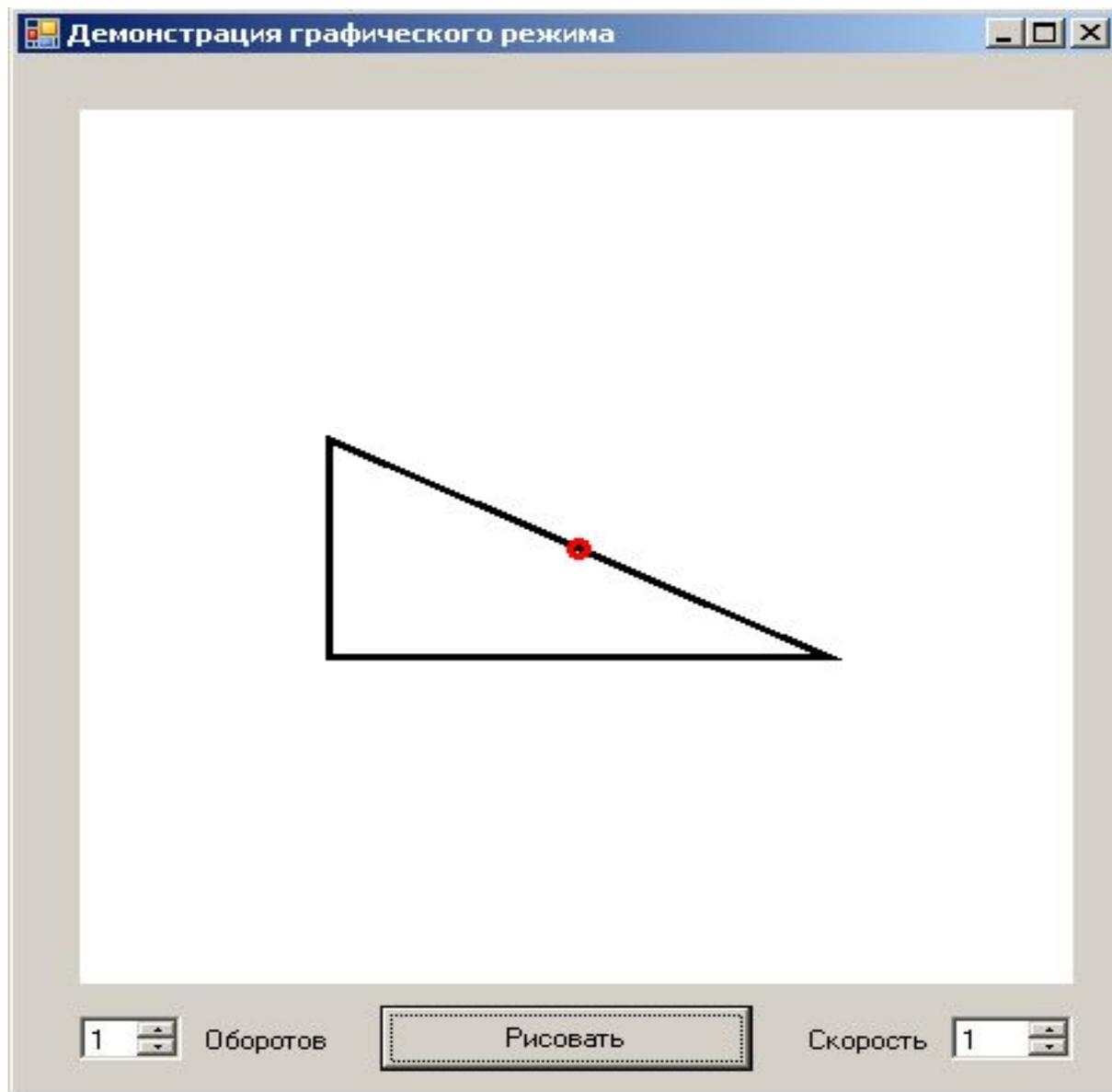
namespace Graph1 {

    using namespace System;
```

Теперь нажимаем

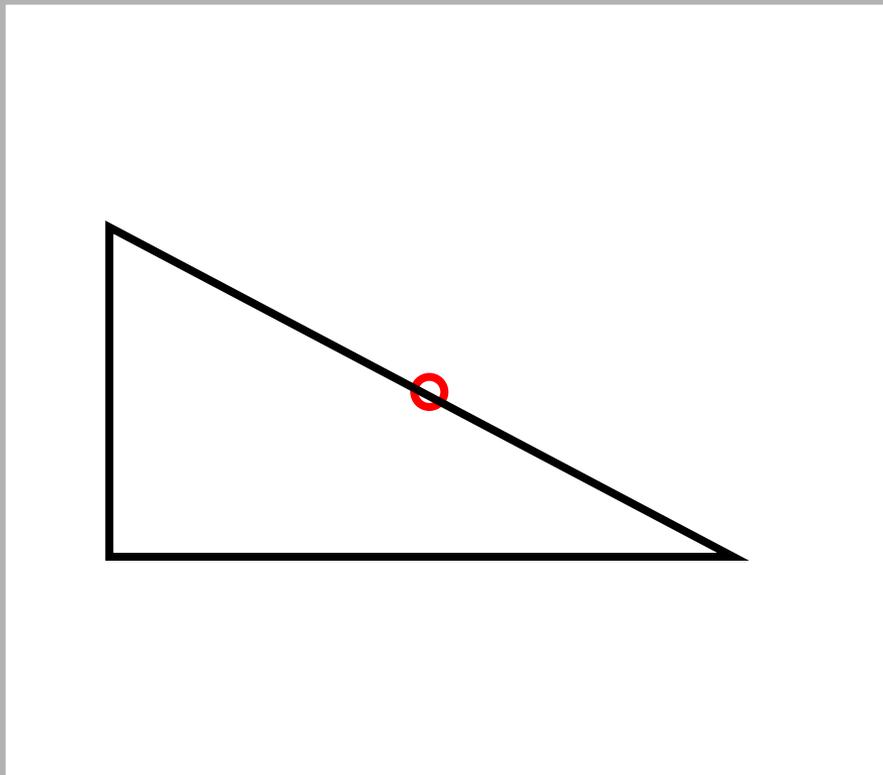


программа начинает работать!





Демонстрация графического режима



1

Оборотов

Рисовать

Скорость

1

Основные графические функции:

1. Очистка области (закрашивание цветом).

```
g->Clear(цвет);
```

Цвет задается, например, так

```
System::Drawing::Color::Red
```

Или через функцию класса System::Drawing::Color::

```
FromArgb(Прозрачность,iRed,iGreen,iBlue);
```

Каждый из параметров принимает значения от 0 до 255.

Максимум (255) означает полностью непрозрачное образование, а для красного, зеленого и синего цветов максимальную насыщенность.

Перо задается через конструктор с указанием цвета и толщины линии, например:

```
Pen ^myPen= gcnnew Pen(System::Drawing::Color::Black,3);
```

Цвет или толщину можно в дальнейшем изменить, например,

```
myPen->Color::set(Color::Red); myPen->Width::set(1);
```

2. Нарисовать дугу эллипса.

```
g->DrawArc(перо,float x, float y,float width, float height,float startAngle,  
float sweepAngle);
```

Задается габаритная рамка для эллипса (координаты левого верхнего угла, ширина и высота, начальный угол и длина дуги в градусах).

3. Нарисовать эллипс.

```
g->DrawEllipse(перо, float x, float y, float width, float height);
```

4. Нарисовать отрезок.

```
g->DrawLine(перо, float x1, float y1, float x2, float y2);
```

5. Нарисовать прямоугольник.

```
g->DrawRectangle(перо, float x1, float y1, float width, float height);
```

6. Нарисовать многоугольник.

```
g->DrawPolygon(. . .);
```

Пример рассмотрен выше.

7. Отобразить текст.

```
g->DrawString(System::String ^s, System::Drawing::Font ^font,  
              System::Drawing::Brush ^brush, float x, float y);
```

Пара (x,y) задает левый верхний угол текста. Шрифт и кисть можно предварительно задать как стандартные, например,

```
System::Drawing::Font^ myFont=this->Font;
```

или, используя конструктор, например:

```
Font ^myFont=gcnnew System::Drawing::Font("Arial",8);  
SolidBrush ^myBrush = SolidBrush(System::Drawing::Color::Blue);
```

Разобрать предложение по словам (решение проблемы русификации консоли)

```
#include <stdafx.h>
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <windows.h>
#include <locale>

using namespace System;

int main(array<System::String ^> ^args)
{
    char s[80];
    char Znak[]=" , . () ";
    char *Temp;

    setlocale(LC_CTYPE,"Russian");
    printf("Введите предложение:");
    // не работает при SetConsoleCP(1251);
    // GetConsoleCP() выдает код страницы
```

```
SetConsoleCP(1251);
gets(s);
SetConsoleCP(866);

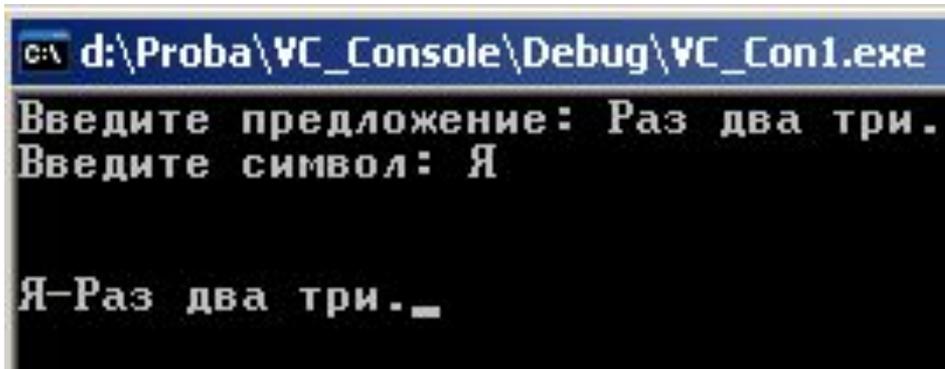
printf("Введено предложение:\n
      %s\nСостоит из слов:\n",s);

Temp=strtok(s,Znak);
do
{
    printf("%s\n",Temp);
    //Console::WriteLine(gcnew
String(Temp));

    Temp=strtok(NULL,Znak);
} while(Temp);
getch();
return 0;
}
```

Разобрать предложение по словам (решение проблемы русификации консоли)

```
#include "stdafx.h "  
#include <stdio.h>  
#include <conio.h>  
#include <locale>  
#include <windows.h>  
  
using namespace std;
```



```
c:\ d:\Proba\VC_Console\Debug\VC_Con1.exe  
Введите предложение: Раз два три.  
Введите символ: Я  
  
Я-Раз два три._
```

```
int main()  
{  
    char c;  
    char s[80];  
  
    setlocale(LC_CTYPE,"Russian");  
  
    printf("Введите предложение: ");  
    SetConsoleCP(1251);  
    gets(s);  
    SetConsoleCP(866);  
  
    printf("Введите символ: ");  
    SetConsoleCP(1251);  
    scanf("%c",&c);  
  
    SetConsoleCP(866);  
    printf("\n\n%c-%s",c,s);  
    _getch();  
}
```

Разобрать предложение по словам (решение проблемы русификации консоли)

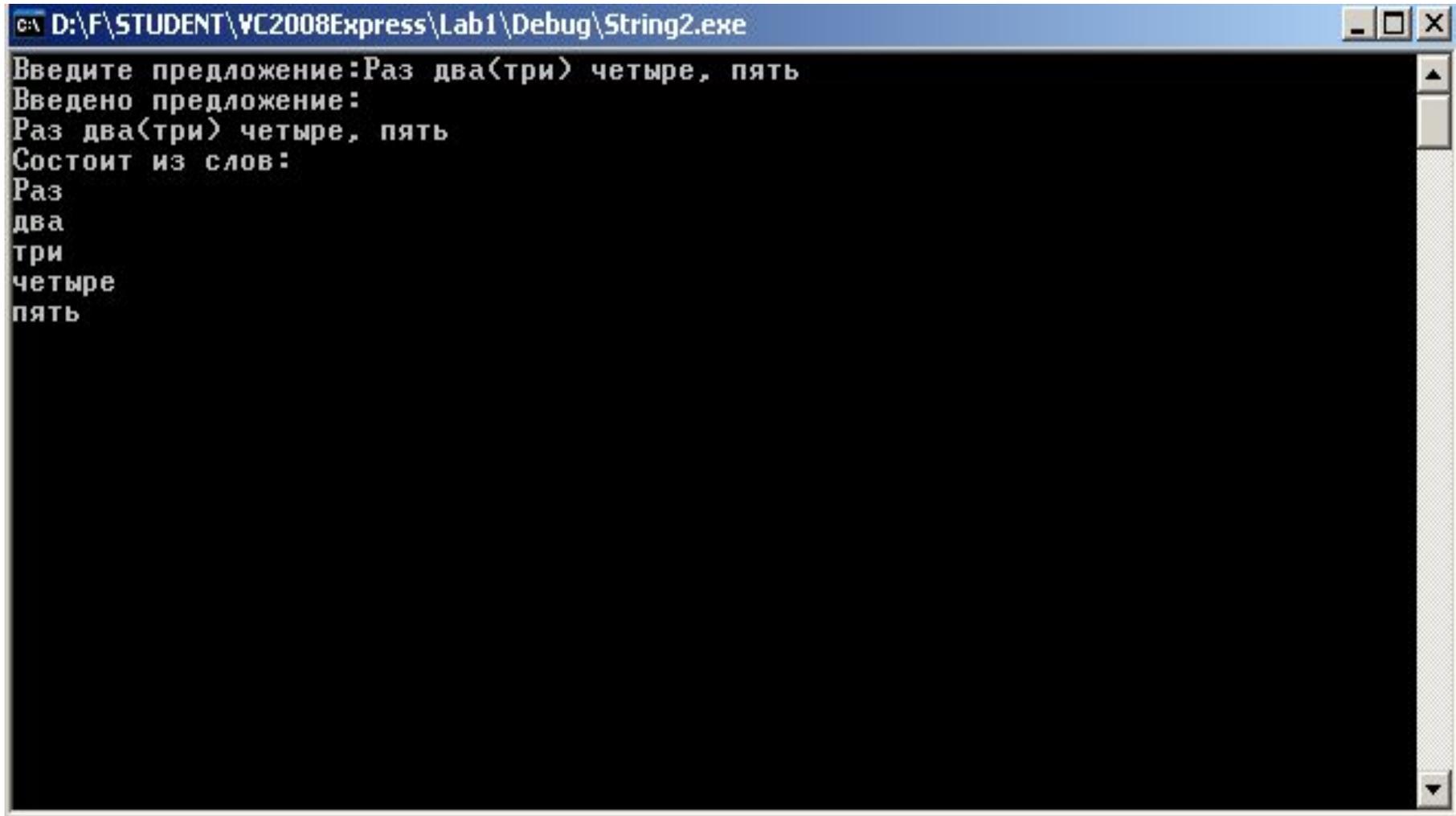
```
#include "stdafx.h"  
#include <stdio.h>  
#include <conio.h>  
#include <iostream>  
#include <locale>  
#include <windows.h>  
  
using namespace std;
```

```
int main()  
{  
    char s[80];  
    setlocale(LC_CTYPE,"Russian");  
    cout<<"введите S";  
    SetConsoleCP(1251);  
    cin>>s;  
    SetConsoleCP(866);  
    cout<<s;  
    _getch();  
}
```



```
c:\ d:\Proba\VC_Console\  
введите S Привет  
Привет
```

Разобрать предложение по словам (решение проблемы русификации консоли)



```
C:\> D:\F\STUDENT\VC2008Express\Lab1\Debug\String2.exe
Введите предложение:Раз два(три) четыре, пять
Введено предложение:
Раз два(три) четыре, пять
Состоит из слов:
Раз
два
три
четыре
пять
```

Разобрать предложение по словам (решение проблемы русификации консоли)

```
#include <stdafx.h>
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <iostream>
```

```
using namespace std;
```

```
using namespace System;
```

```
int main(array<System::String ^> ^args)
```

```
{
char s[80];
char Znak []=" , . ( ) ";
char *Temp;
```

```
Console::WriteLine(L"Введите предложение:");
gets (s) ;
```

```
Console::WriteLine(L"Введено предложение:");
cout<<s<<endl;
```

```
Console::WriteLine(L"Состоит из слов:");
```

```
Temp=strtok (s , Znak) ;
```

```
do
```

```
{
```

```
cout<<Temp<<endl;
```

```
Temp=strtok (NULL , Znak) ;
```

```
} while (Temp) ;
```

```
getch () ;
```

```
return 0 ;
```

Переписать предложение, заключив слова в скобки.

```
#include <stdafx.h>
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <iostream>

using namespace std;
using namespace System;

int main(array<System::String ^> ^args)
{
    char s[80],t[80];
    char Znak[]=" ,()";
    char *Temp;

    Console::WriteLine(
        L"Введите предложение:");
    gets(s);
```

```
Console::WriteLine(L"Введено предложение:");
cout<<s<<endl;
Console::WriteLine(L"Заключаем в собки:");

Temp=strtok (s ,Znak) ;
do
    {
        strcat (t, "(") ;
        strcat (t,Temp) ;
        strcat (t, ")" ) ;

        Temp=strtok (NULL, Znak) ;
    } while (Temp) ;
strcat (t, ".") ;
cout<<t;
getch () ;
return 0 ;
}
```

10 избранных отличий C++ от стандартного C

1. Комментарии

```
// комментарий  
int x; // комментарий  
x++;  
/*  
комментарий  
*/  
x*=2;
```

2. Новые возможности описания объектов

```
int n,k;  
scanf ("%d" , &n) ;  
k=n+1;  
float y;  
for(int i=0; i<n; i++) ;
```

Увеличивается перечень зарезервированных слов, которые нельзя использовать в качестве имён, например,

class, new, delete

и так далее.

Доступ к глобальным переменным, скрытым локальными переменными с тем же именем

```
int i=0; // глобальная
```

```
F()
```

```
{  
int i=0; //локальная  
i++; //локальная  
::i++; // глобальная  
}
```

3. Аргументы по умолчанию

// В прототипе функции можно задавать значения по умолчанию

```
void DrawCircle(int x=100, int y=75, int R=50);
```

// Правильный вызов

```
DrawCircle();           // Рисуем окружность R=50 с центром (100,75)
DrawCircle(200);       //окружность R=50 с центром (200,75)
DrawCircle(200,300);   //окружность R=50 с центром (200,300)
DrawCircle(200,300,150); //окружность R=150 с центром (200,300)
```

// Неправильный вызов

```
DrawCircle(, ,25);
```

// Разрешается опускать аргументы только справа!

// Неправильный прототип

```
void DrawCircle(int x, int y=75, int R=50);
```

// Разрешается задавать не все значения, но те, что не заданы, должны быть в списке правее заданных, а не наоборот!

4. Передача параметров по ссылке

```
void Swap(int& x, int &y)
// int& читается как ссылка на int
// конструкции int& x и int &x - равносильны
{
int z;
z=x; x=y; y=z;
}

main()
{
int x=1,y=2;
Swap(x,y);
. . .
}
```

5. ПОТОКОВЫЙ ВВОД/ВЫВОД

```
#include <iostream>

using namespace std;

main()
{
    int x;
    char s[80];
    cout<<"Введите x "<<endl;
    cin>>x;
    cout<<"x="<<x<<endl;
    cin>>s;
    cout<<"string= "<<s<<endl;

    . . .
```

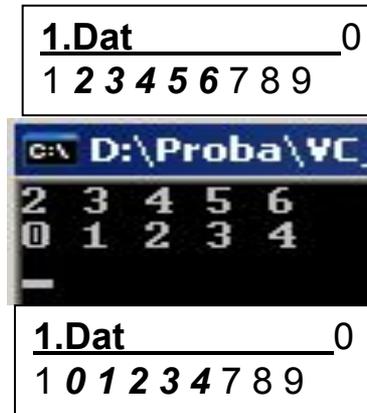
Манипуляторы

```
#include <iomanip>

cout<<setw(n); // int n - ширина поля
cout<<setprecision(n); // n – др.часть
cout<<fixed; // по формату %f
cout<<scientific; // по формату %e
//задать систему счисления
cout<<dec<<hex<<oct; // одно из ...
```

ПОТОКОВЫЙ ВВОД/ВЫВОД ИЗ ФАЙЛА (в файл)

```
#include <conio.h>           ios::beg // задает
#include <fstream>          ios::loc // позицию
using namespace std;       ios::end // в файле
main()
{
fstream f("1.dat",ios::in | ios::out | ios::trunc);
// ios::trunc очищает файл, если он существует,
// иначе создает новый файл.
// Без этой опции файл 1.dat не будет создан!
// Для создания файла можно использовать режим
// f.open("1.dat",ios::out);
//
int i,n;
for(i=0;i<10;i++)
    f<<i<<' ';
f.seekg(4l,ios::beg);
for(i=0;i<5;i++)
    {f>>n;cout<<n<<' ';}
cout<<endl;
```



```
1.Dat _____0
1 2 3 4 5 6 7 8 9

c:\ D:\Proba\VC_
2 3 4 5 6
0 1 2 3 4
_

1.Dat _____0
1 0 1 2 3 4 7 8 9
```

```
f.seekp(4l,ios::beg);
for(i=0;i<5;i++)
    f<<i<<' ';
f.seekg(4l,ios::beg);
for(i=0;i<5;i++)
    {f>>n;cout<<n<<' ';}
cout<<endl;
f.close();
getch();
}
// if(f.is_open()) {...}
// if(!f.eof()) {...}
// if(f.good()) {...}
// if(f.fail()) {...}
// f.clear() —сброс ошибки
// char ch,s[80],buf[100];
// f.get(ch);f.put(ch);
// f.getline(s,80);
// для режима ios::binary
// f.read(buf,100)
// f.write(buf,100)
// по умолчанию: ios::text
```

6. Операторы new и delete

Применяются вместо функций

`malloc()` и `free()`

Из библиотеки `malloc.h`

```
//*****
```

```
#include <iostream>
using namespace std;
```

```
main()
{
int n;
cin>>n;
int *y;
y= new int[n];
for(int i=0; i<n; i++)
    y[i]=0;
```

```
...
delete[] y;
```

```
#include <iostream>
```

```
using namespace std;
```

```
main()
{
int i,j,n,m;
cin>>n>>m;
int **a;
a = new int* [n];
for(i=0; i<n; i++)
    a[i]= new int [m];

for( i=0; i<n; i++)
    for( j=0; j<n; j++)
        cin>>a[i][j];
...
for( i=n-1; i>0; i--)
    delete[] a[i];
delete[] a;
```

7. Подставляемые (inline) функции

```
inline int xCube(int x)
{
    return x*x*x;
}
```

```
main()
{
    . . .
    cout<<xCube(2)<<endl;
    . . .
    cout<<xCube(3);
    _getch();
}
```

8. Перегрузка функций

```
void Print(int n)
{
printf("\nInt=%d",n);
}
```

```
void Print(float x)
{
printf("\nFloat=%f",x);
}
```

```
void Print(char* s)
{
printf("\nStr=%s",s);
}
```

```
main()
{
int i=5;
float x=3.14;
char s[]="МГСУ";
Print(i);
Print(x);
Print(s);
_getch();
}
```

9. Упрощенное описание структур

```
struct Z {  
    char name[20];  
    char vid[20];  
    long summa;  
    char data[11];  
};
```

```
main()  
{  
    struct Z A[10];  
    // можно короче!  
    Z *b,c,d[20];
```

10. Подключение математической библиотеки

```
#define _USE_MATH_DEFINES
#include <cmath>

main()
{
float x=M_PI;
cout<<x;
_getch();
}
```

При условии компиляции в режиме /clr
доступны функции класса System::Math::

Например,

Math::Cos(x); Math::PI; Math::Sqrt(x); Math::Pow(x,y) и др.

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);
// В 2, 8 или 16-ю систему счисления
s = Convert::ToString(число,основание);

// кроме того
s = String::Format("Pi={0:F3}",Math::PI);
// или так
Double X;
X=3.14;
s = X.ToString("F3"));

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

Операции для типа **String**.

```
// Пусть в программы определены  
// String ^s, ^s1, ^s2
```

Вместо библиотеки **<string.h>** для типа **String** применяются перегруженные операции и методы класса.

```
// = вместо strcpy  
// + или += вместо strcat
```

```
s="МГСУ" ;  
s+="-МИСИ" ;  
s1=s+" (1921) " ;
```

```
// метод CompareTo вместо strcmp  
if (s->CompareTo ("абракадабра") ==0) continue;
```

```
// можно применять == или != , но не < или >  
if (s=="абракадабра") continue;
```

Методы для типа **String**.

`s->Length` //определение длины строки

`s->IndexOf ("б")` //определение позиции подстроки или символа (отсчет от 0)

`s->LastIndexOf (L'б')` // кодировка символа UNICODE (wchar_t)

`s->Substring (pos)` //выделение подстроки, начиная с позиции pos до конца

`s->Substring (pos, len)` // здесь len задает длину вырезаемой строки

`s->Insert (pos, s1)` // вставка в строку s подстроки s1, начиная с позиции pos

`s->Remove (pos, len)` // удаление len символов из строки, начиная с позиции pos

`s->Replace (s1, s2)` // замена последовательности символов s1

// на последовательность символов s2

`s->ToUpper ()` // преобразование в верхний регистр

`s->ToLower ()` // преобразование в нижний регистр

`s->Split (s1, опция)` //разбивка строки на подстроки с разделителями из перечня s1

// в массив типа `cli:array <System::String^>`

// опция `System::StringSplitOptions::RemoveEmptyEntries`

//игнорирует пустые строки

//альтернатива - `System::StringSplitOptions::None`

```

#include "stdafx.h"
using namespace System;

int main()
{
String ^s, ^s1, ^s2;
s=Console::ReadLine();
Console::WriteLine(s);
Console::WriteLine(s->Length);
Console::WriteLine(s->IndexOf("б"));
Console::WriteLine(s->LastIndexOf(L'б'));
Console::WriteLine(s->Substring(2,3));
s1=s->Insert(4,"швабра");
Console::WriteLine(s1);
s2=s1->Remove(4,6);
Console::WriteLine(s2);
Console::WriteLine(s2->Replace("аб","АБ"));
Console::WriteLine(s2->ToUpper());
Console::WriteLine(s2->ToLower());

Console::ReadKey();
return 0;
}

```

Методы для типа **String**.

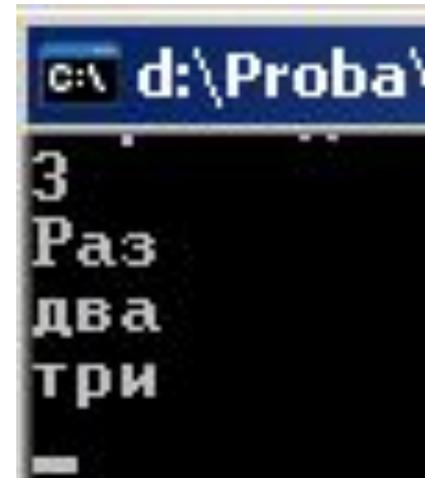
```

c:\ d:\Proba\VC_Console\
абракадабра
абракадабра
11
1
8
рак
абрашвабракадабра
абракадабра
АБракадАБра
АБРАКАДАБРА
абракадабра

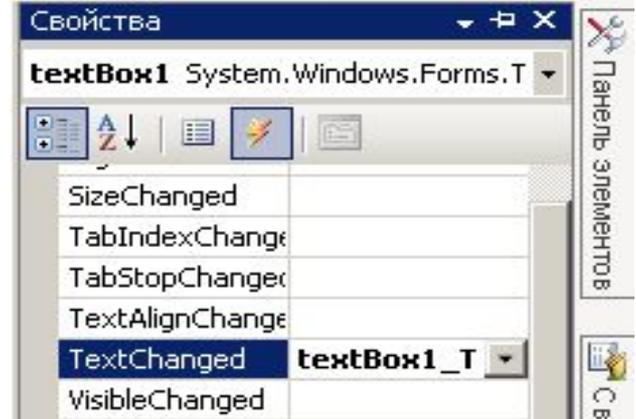
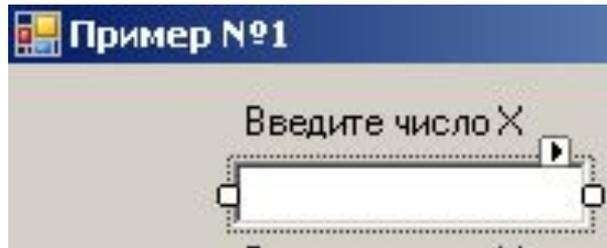
```

Пример использования метода **Split**. (аналог **strtok** для типа **char**)

```
array <System::String^> ^c = gcnew array <System::String^> (3);  
c[0]=L" ";  
c[1]=L",";  
c[2]=L".";  
s1="Раз, два      три";  
  
array <System::String^> ^ss;  
ss = s1->Split(c, System::StringSplitOptions::RemoveEmptyEntries);  
Console::WriteLine(ss->Length);  
int i, N;  
N=ss->Length;  
for(i=0; i<N; i++)  
    Console::WriteLine(ss[i]);
```



Как контролировать ввод числовых значений в поле типа TextBox



```
private: System::Void textBox1_TextChanged(System::Object^ sender, System::EventArgs^ e)
{
    if(this->textBox1->Text=="") return;
    try
    {
        System::Int32::Parse(this->textBox1->Text);
    }
    catch(...) //catch(Exception ^e)
    {
        MessageBox::Show("Введено не число","Ошибка!",
            System::Windows::Forms::MessageBoxButtons::OK,
            System::Windows::Forms::MessageBoxIcon::Error);
    }
    this->textBox1->Text=textBox1old;
    // String ^textBox1old – внешняя переменная,
    // где хранится прежнее значение
    textBox1old=this->textBox1->Text;
}
```

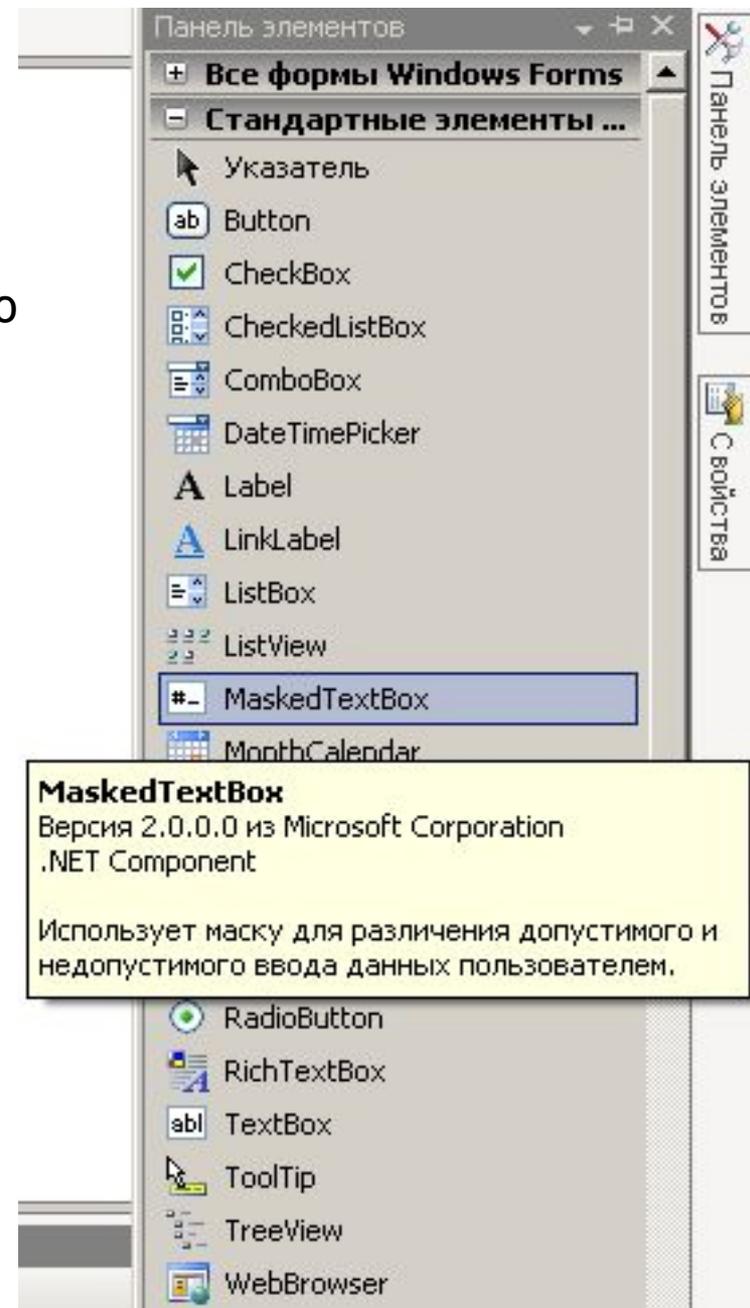
Еще проще эта проблема решается с помощью
компонента

MaskedTextBox

(свойство Mask)

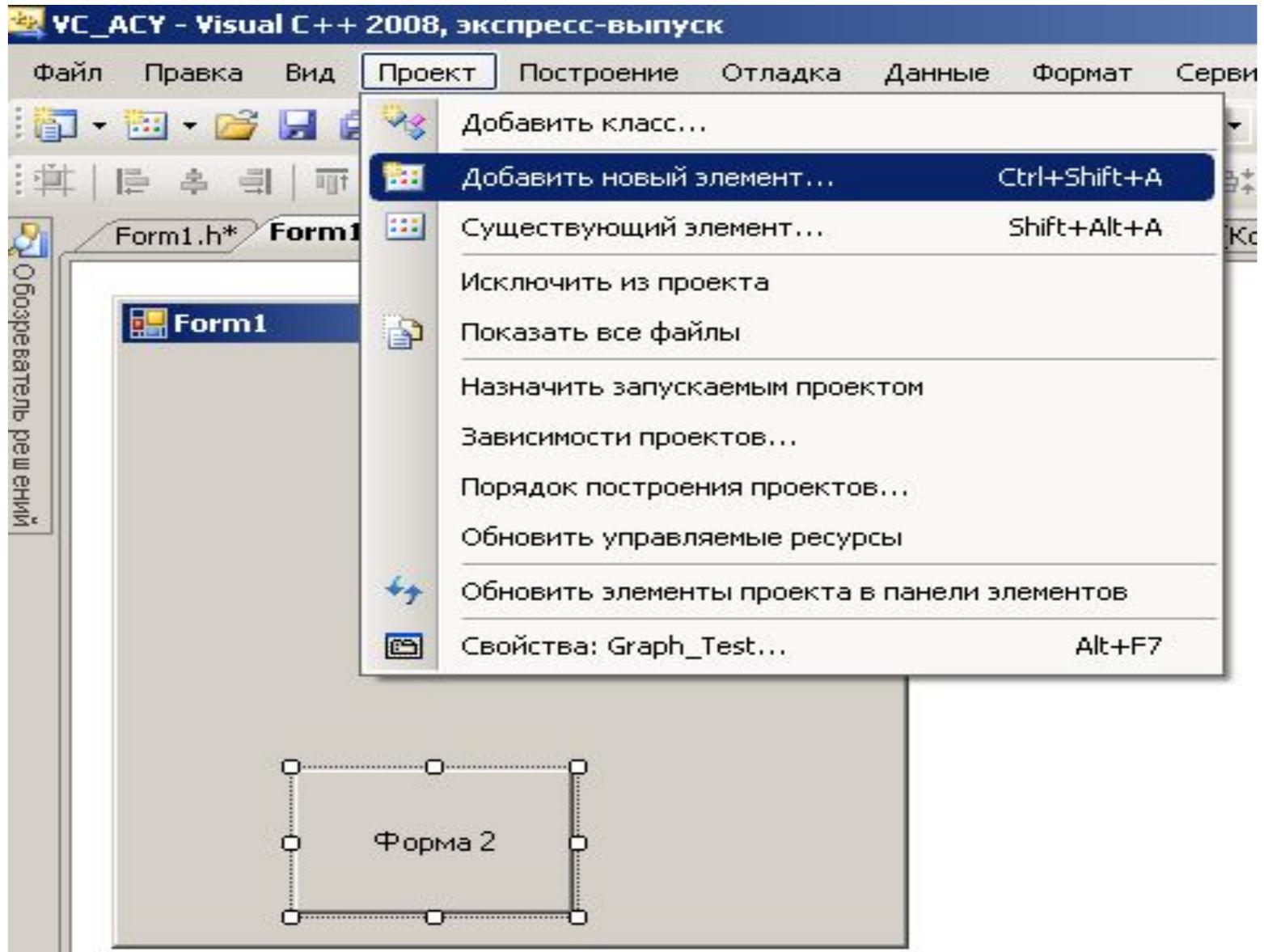
Предыдущий пример является хорошей
иллюстрацией применения средства

try ... catch ...



Пример использования нескольких форм

FORM2 вызывается из FORM1 при нажатии на кнопку «Форма2»

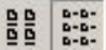


Добавление нового элемента - Graph_Test



Категории:

Шаблоны:



- Visual C++
 - UI
 - Код
 - Вкладки свойств

Установленные шаблоны Visual Studio

- Форма Windows Forms
- Заголовочный файл (.h)
- Класс компонента
- Файл C++ (.cpp)
- Вкладка свойств (.vsprops)

Мои шаблоны

- Найти шаблоны в Интернете...

Создает CLR из содержания других элементов управления Windows

Имя:

Form2

Расположение:

d:\F\STUDENT\VC2008Express\VC_ACY\Graph_Test

Обзор...

Добавить

Отмена

```
Form1.h  Form1.h [Конструктор]
(Глобальная область)
#pragma once
#include "Form2.h"
```

// Добавляем в самое
// начало кода Form1.h

// Програмируем метод CLICK для кнопки «Форма2», размещенной на FORM1

```
Form2 ^F2= gcnw Form2();
```

```
F2->Show(); // Немодальная форма
```

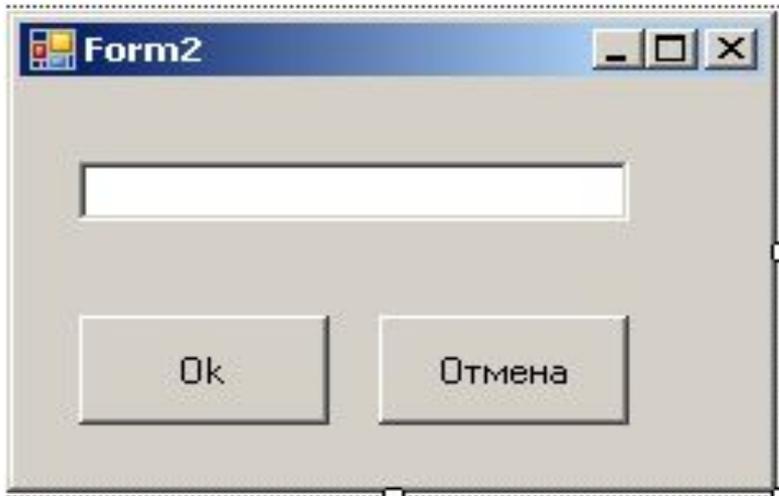
```
// Можно свободно переключаться между Form1 и Form2
```

```
// ИЛИ
```

```
F2->ShowDialog(); // Модальная форма
```

```
// Перейти на Form1 можно только, если закрыть Form2
```

Для обмена информацией между формами делаем следующее



```
Метод Form2_Load () {  
    textBox1->Text = a; //инициализация  
}
```

```
Метод button1_Click () {  
    a = textBox1->Text->ToString();  
    this->Close();  
}
```

```
Метод button2_Click () {  
    a = "Ввод отменен!";  
    this->Close();  
}
```

// На форме F2 размещаем две кнопки и текстовое поле

// Добавляем ОТКРЫТУЮ (public) переменную a.

// Из формы F1 к ней возможен свободный доступ

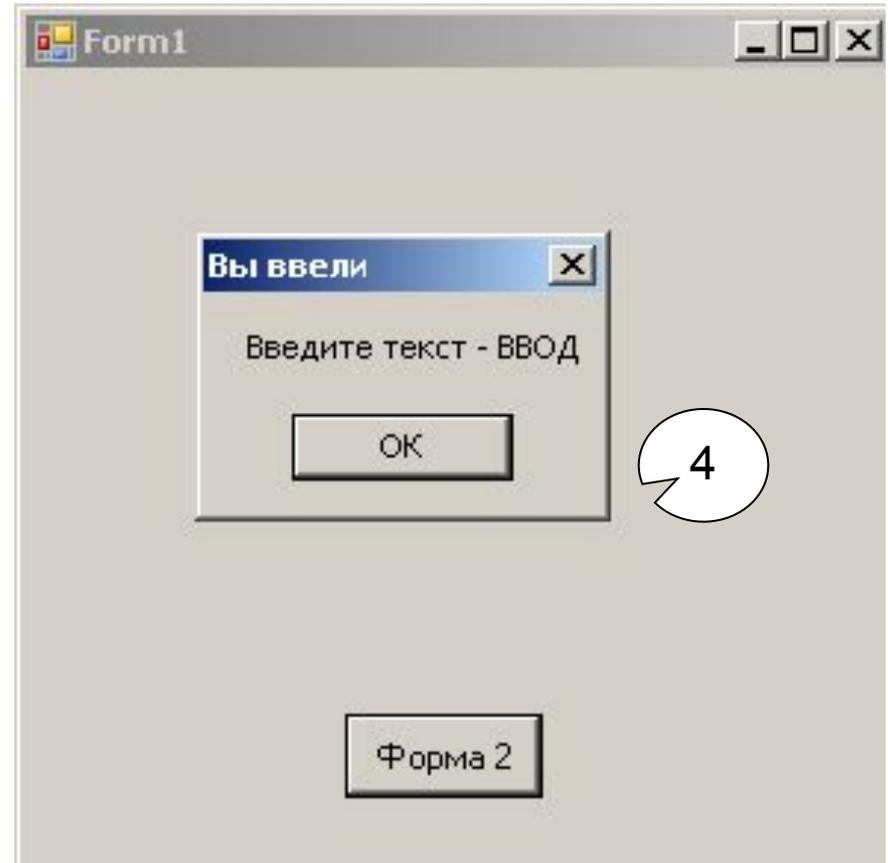
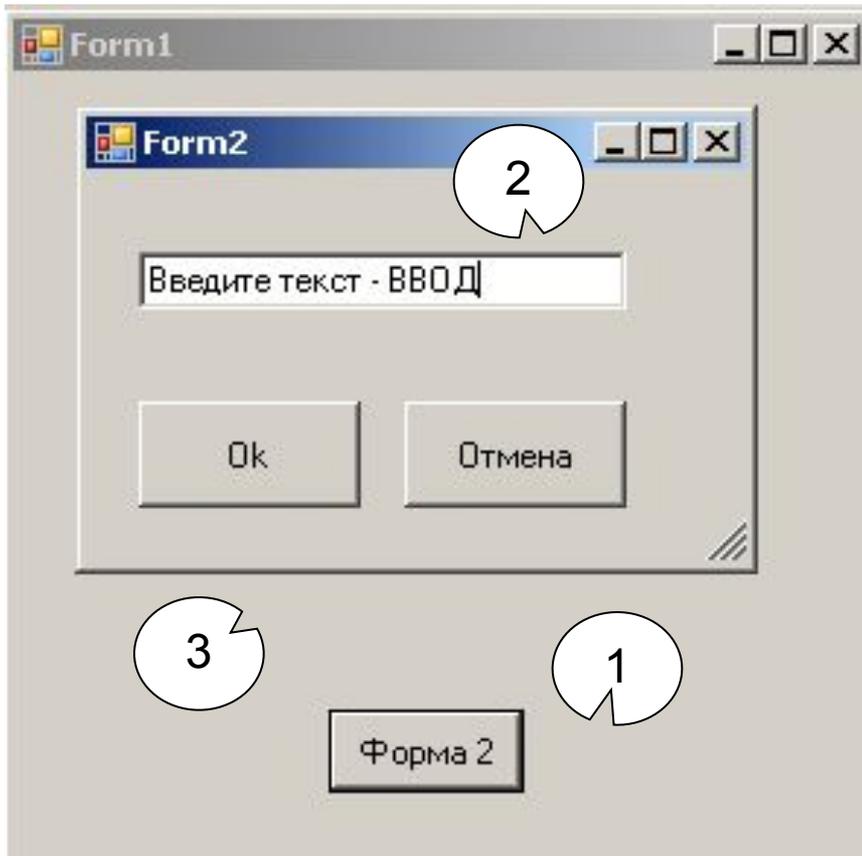
```
private: System::Windows::Forms::Button^ button1; //добавляется конструктором  
private: System::Windows::Forms::Button^ button2; //добавляется конструктором  
private: System::Windows::Forms::TextBox^ textBox1; //добавляется конструктором
```

```
public: String^ a; //добавляется вручную
```

// Програмуємо метод CLICK для кнопки «Форма2», розміщеної на FORM1

```
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e)
{
    Form2 ^F2=gcnew Form2();
    F2->a = "Введіть текст";
    F2->ShowDialog();
    MessageBox::Show(F2->a,"Вы ввели");
}
```

- 1 Последовательность
- 2 Действий
- 3 При запуске
- 4 Программы



Передача структуры из формы в форму

В проект добавляем
файл описания
шаблона структуры

StructZ.h

```
#ifndef STRUCTZ
#define STRUCTZ

struct z
{
    char Name[20];
    int Age;
};

#endif
```

Начало Form1.h должно быть таким

```
#pragma once
#include "StructZ.h"
#include "Form2.h"
#include <string.h>
```

```
struct z s;
```

Начало Form2.h должно быть таким

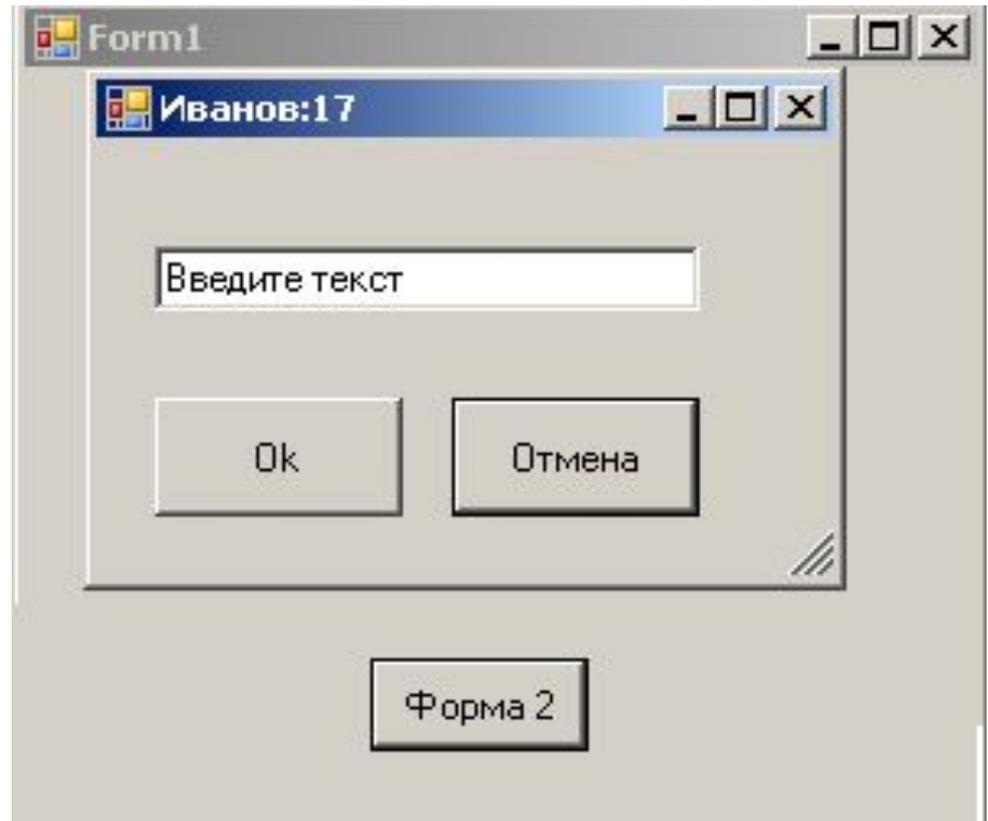
```
#pragma once
#include "StructZ.h"

extern struct z s;
```

Передача структуры из формы в форму

Процедура вызова формы F2
из формы F1

```
s.Age = 17;  
strcpy(s.Name, "Иванов");  
Form2 ^F2 = gcnew Form2();  
F2->a = "Введите текст";  
F2->ShowDialog();
```



Вывод полученного в заголовке формы F2 (Form2_Load)

```
String ^S = gcnew String(s.Name); // extern struct z s;  
this->Text = S + ":" + Convert::ToString(s.Age);  
textBox1->Text = a;
```

Передача аргументов функции main в форму

В главном (стартовом) файле проекта

```
int main(array<System::String ^> ^args)
{
    // Включение визуальных эффектов Windows XP до создания
    // каких-либо элементов управления
    Application::EnableVisualStyles();
    Application::SetCompatibleTextRenderingDefault(false);

    // Создание главного окна и его запуск
    // Вместо Application::Run(gcnew Form1());
    Application::Run(gcnew Form1(args));
    return 0;
}
```

В классе формы Form1.h

```
public ref class Form1 : public System::Windows::Forms::Form
{
public:
    ////////// Аргументы функции Main
    array<System::String ^> ^args;
    //// Вместо Form1(void)
    Form1(array<System::String ^> ^args)
    {
        InitializeComponent();
        //
        //TODO: добавьте код конструктора
        //
        this->args = args;
        if (args->Length>0)
            MessageBox::Show(args[0], args->Length.ToString());
    }
}
```

Как создать проект Windows Forms в VC-2013

1) Создать пустой проект CLR

Создать проект

Последние файлы

Установленные

Шаблоны

- Visual C++
 - Магазин Windows
 - ATL
 - CLR
 - Общие
 - MFC
 - Тест
 - Win32
 - TypeScript
- Другие языки
- Другие типы проектов
- Проекты моделирования
- Примеры

В сети

.NET Framework 4.5 Сортировать по: По умолчанию

	Библиотека классов	Visual C++
	Консольное приложение CLR	Visual C++
	Пустой проект CLR	Visual C++

Установлено: Шаблоны - поиск (Ctrl+P)

Тип: Visual C++

Пустой проект по созданию локального приложения

[Щелкните здесь для поиска шаблонов в Интернете.](#)

Имя: Project

Расположение: c:\users\экспрессмаг\documents\visual studio 2013\Projects

Имя решения: Project

Обзор...

Создать каталог для решения

Добавить в систему управления версиями

OK Отмена

2) Добавить в него форму, например, MyForm.h

Меню Проект / Добавить новый элемент

Добавление нового элемента - Project

Установлено: Шаблоны - поиск (Ctrl+ )

Сортировать по: По умолчанию

Установленные

- Visual C++
 - UI
 - HLSL
 - Код
 - Данные
 - Ресурс
 - Web
 - Службная программа
 - Вкладки свойств
 - Test
- В сети

	Форма Windows Forms	Visual C++
	Пользовательский элемент управления CLR	Visual C++
	XML-файл определения ленты MFC	Visual C++

Тип: Visual C++

Создает CLR из содержания других элементов управления Windows

[Щелкните здесь для поиска шаблонов в Интернете.](#)

Имя: MyForm.h

Расположение: c:\Users\экспрессмар\documents\visual studio 2013\Projects\Project\Project\ 

Обзор...

Добавить Отмена

3) Не помешает открыть
обозреватель решений
И в стартовом файле проекта
(в нашем случае это MyForm.cpp)
прописать код

Microsoft Visual Studio

ВИД	ПРОЕКТ	СБОРКА	ОТЛАДКА	КОМАНДА	ФОРМ
<>	Код			Ctrl+Alt+0	
🏗	Конструктор			Shift+F7	
📊	Обозреватель решений			Ctrl+Alt+L	
👤	Командный обозреватель			Ctrl+\, Ctrl+M	
🏗	Обозреватель архитектуры			Ctrl+\, Ctrl+R	
🏗	Обозреватель объектов SQL Server			Ctrl+\, Ctrl+S	

The screenshot shows the Microsoft Visual Studio interface. The top menu bar includes: ФАЙЛ, ПРАВКА, ВИД, ПРОЕКТ, СБОРКА, ОТЛАДКА, КОМАНДА, СЕРВИС, ТЕСТ, АРХИТЕКТУРА. The toolbar contains icons for navigation and development, along with a dropdown for the Windows Local Debugger set to 'Авто'. The Solution Explorer on the left shows a project named 'Project' with a tree structure: 'Project' (circled in red), 'Внешние зависимости', 'Заголовочные файлы', 'MyForm.h', 'Файлы исходного кода', 'MyForm.cpp', and 'Файлы ресурсов'. The code editor on the right shows the 'MyForm.h' file in 'Конструктор' mode, with the following code in the global scope:

```
#include "MyForm.h"
```

```

#include "MyForm.h"

using namespace Project; // это имя нашего проекта

[STAThreadAttribute]
int main(array <System::String ^> ^args)
{
    // Включение визуальных эффектов Windows XP
    Application::EnableVisualStyles();
    Application::SetCompatibleTextRenderingDefault(false);

    // Создание главного окна и его запуск
    Application::Run(gcnew MyForm());
    return 0;
}

```

4) Установить свойства конфигурации проекта.
(Меню ПРОЕКТ / Свойства)

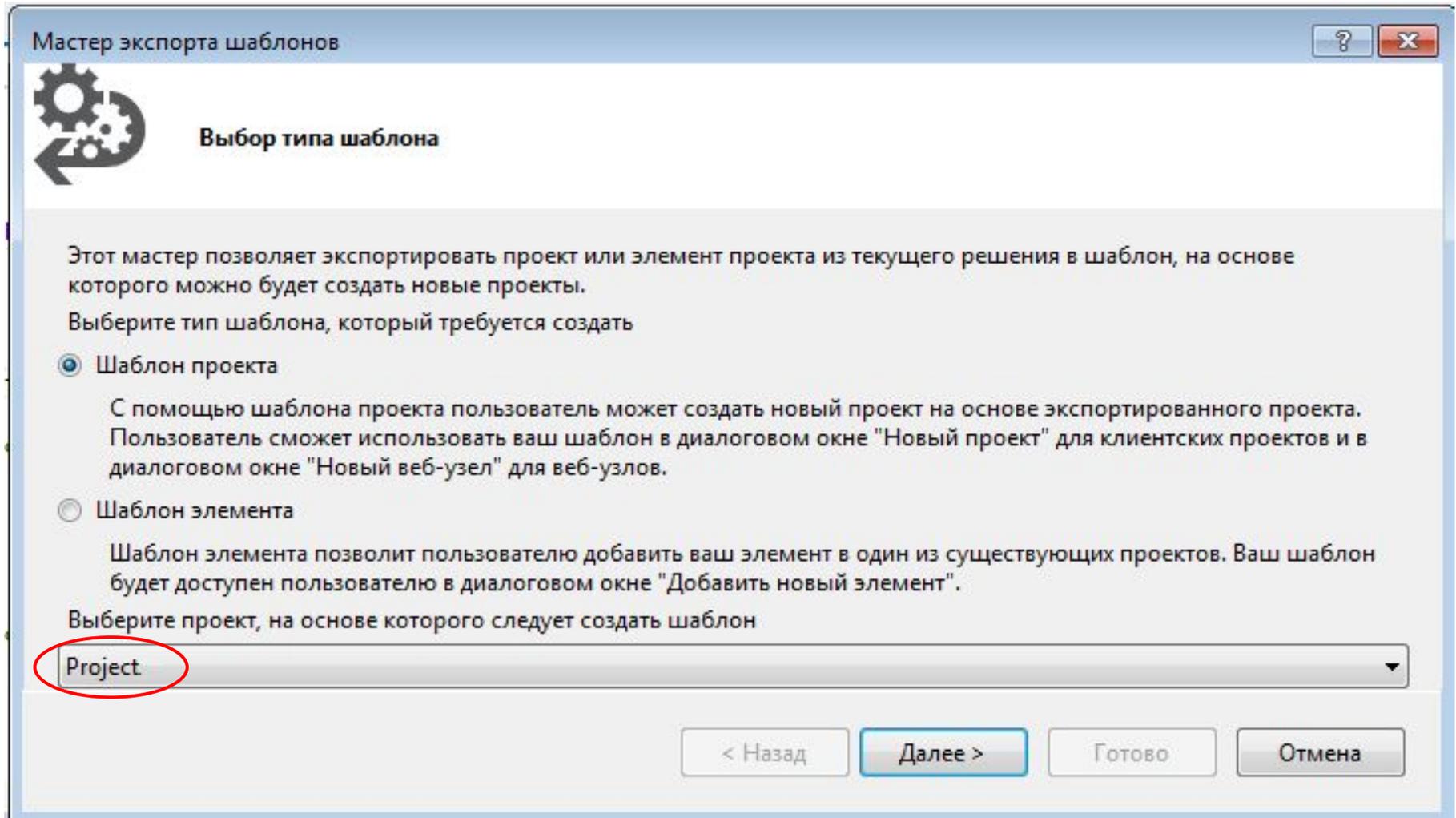
Компоновщик / Система / Подсистема =
Windows (/SUBSYSTEM:WINDOWS).

и

Компоновщик / Дополнительно / Точка входа = **main**
И ВСЁ !!!

Шаблон данного проекта можно сохранить для повторного использования

5) Выбрать пункт меню ФАЙЛ / ЭКСПОРТ ШАБЛОНА



6) Нажать ДАЛЕЕ и заполнить поля формы ИМЯ ШАБЛОНА и ОПИСАНИЕ ШАБЛОНА (не обязательно).

Мастер экспорта шаблонов

 Выбор параметров шаблона

Имя шаблона:
WFApp

Описание шаблона:
Приложение Windows Forms

Изображение значка:
 Обзор...

Просмотр изображения:
 Обзор...

Местоположение выходных файлов:
C:\Users\Master\Documents\Visual Studio 2013\My Exported Templates\WFApp.zip

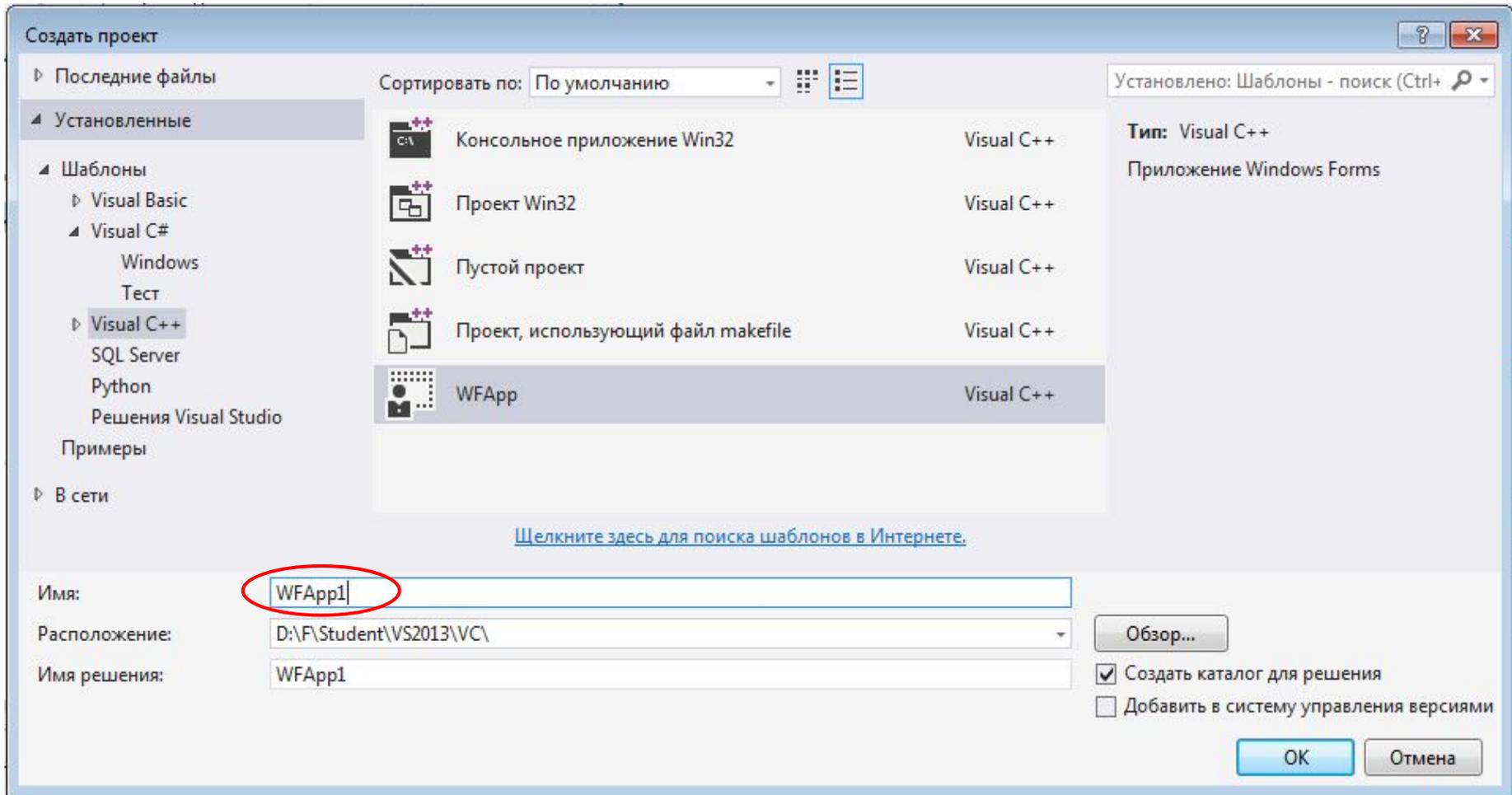
Автоматически импортировать шаблон в Visual Studio

Открыть окно проводника в папке выходных файлов

< Назад Далее > Готово Отмена

7) Нажать ГОТОВО.

Теперь при создании нового проекта можно использовать свой шаблон



В созданных мастером файлах проекта **cpp** (стартовый) и **h** (класс формы) необходимо поменять название пространства имен с **Project1** (наследовано из шаблона) на имя текущего проекта (например, **WFApp1**, как в данном примере). В файле **cpp** изменению подлежит оператор **using namespace Project1**. В файле **h** - объявление пространства имен **namespace Project1**