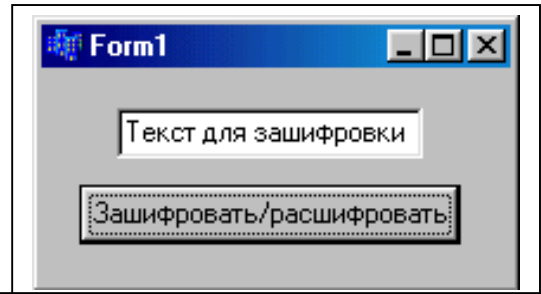


```
#include "MDLL.h"
Button1Click
{Edit1->Text=Code_Dec(Edit1->Text.c_str(),'A'); }
```

DLL: MDLL.cpp

MDLL.h



```
#include <vcl.h>
#include <windows.h>
#pragma hdrstop
//-----

#pragma argsused
#include "MDLL.h"
int WINAPI DllEntr-----

char * Code_Dec(char *s, char Key)
{ for (int i = 0; ; i++)
  { if (s[i] == '\0') break;
    s[i] = s[i] ^ Key;
  } return s;
}
```

```
#ifndef _MDLL_H
#define _MDLL_H

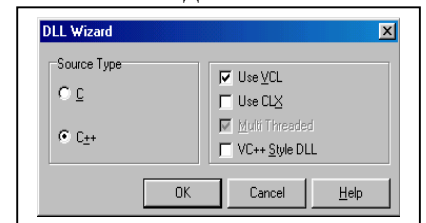
#ifdef __DLL__
#define DLL_EI __declspec(dllexport)
#else
#define DLL_EI __declspec(dllimport)
#endif

extern "C" char * DLL_EI Code_Dec(char *s, char
Key);

#endif
```

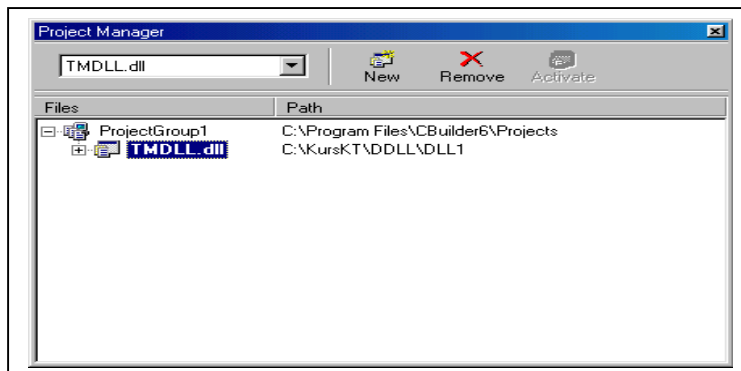
Последовательность создания DLL для статической загрузки:

1. Запустить C++builder, выполнить File->Close All, тем самым убрать форму и модуль.
2. Выполнить File->New->Other...--в диалоговом окне New выбрать DLL Wizard (мастер создания DLL), настроить диалоговое окно (ничего не изменять) – оставить выбор языка C++, можно оставить, а можно и убрать выбор включения библиотеки компонентов <vcl.h> (если будет использоваться какой-либо компонент или его класс), индикатор MC++ Style DLL обеспечивает создание DLL в стиле Visual C++, после чего нажать «OK»



3. Убрать все комментарии, после чего записать программную реализацию одной или нескольких методов (функций), которые должны выполняться DLL (пример в файле MDLL.cpp).
4. Перед строкой int WINAPI DllEntr----- вставить имя предполагаемого заголовочного файла, например: #include "MDLL.h" .
5. Выполнить File->New->Other...-- в диалоговом окне New выбрать значек Header file и приступить к написанию заголовочного файла (пример в MDLL.h).
6. Сохранить оба файла – один из них с расширением cpp, а другой – h.
7. Сохранить созданные файлы в виде проекта, для чего выполнить File->Save Project As .. и при сохранении **обязательно** измените имя файла, например, вместо Project1 рекомендуем записать: **PMDLL**.
8. Построить созданное приложение для чего выполнить команду из главного меню Project->Build MDLL. Если ошибок нет то в строке **Done** -> **Build All**, т.е. все в порядке.
9. Закрываются все созданные файлы и создается файл приложения, которое использует созданную библиотеку.
10. После создания формы и записи исполняемой программы, обязательно необходимо подключить библиотеку DLL: MDLL.h.
11. На заключительном этапе необходимо подключить библиотеку с расширением **.lib**, которое создается на 8 шаге. Для подключения библиотеки необходимо выполнить View->Project Manager щелкнуть на исполняемом файле правой кнопкой и в диалоговом окне выполнить команду Add, после чего в диалоговом окне в списке файла найти строку Library file (*.lib). указать свой файл и нажать кнопку открыть.
12. После проделанных шагов можно запускать свое приложение и работать, если отсутствуют ошибки.

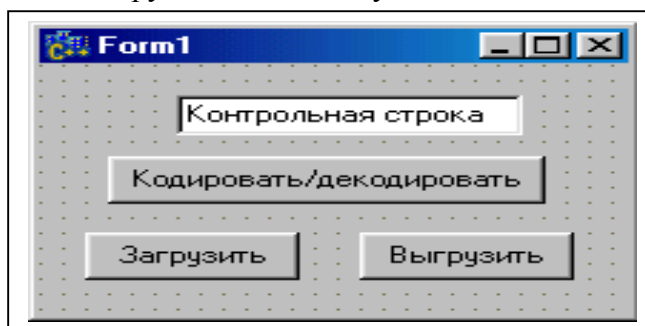
Примечание: Пункт 11 можно и не выполнять, но подключить файл с расширением **lib** обязательно, для чего для приложения выполнить Project->View Source. Появляется программа исполняемого файла, в котором имеется следующая строчка **USEFORM("Unit1.cpp", Form1);** После этой строчки добавить самостоятельно строку: **USELIB("TMDLL.lib");** Можно работать.



Последовательность создания DLL для динамической загрузки:

- 1) Первые 9 пунктов предыдущих шагов остаются без изменения
- 2) Закрываем созданные файлы и строим исполняемое приложение, т.е. File->New->Application;
- 3) На форме устанавливаем необходимые компоненты и добавляем две кнопки – одна из них должна загрузить созданную библиотеку, другая – выгрузить библиотеку.

- 4) Для примера рассмотрим вышеизложенную задачу: -> (см. рис.)
- 5) В заголовочном файле приложения в описание класса формы вносим след. дополнения:



```
private:        // User declarations
//объявление типа указателя на DLL
HINSTANCE dllInstance;
//объявление типа указателя на
функцию
typedef char (__import * FType(char *,
char));
FType * C_D;
public:
```

- 6) Записываем коды соответствующих обработчиков:

```
void __fastcall TForm1::Button2Click(TObject
*Sender)
{// загрузка DLL
dllInstance = LoadLibrary("PUMyDLL.dll");
if (dllInstance)
//получение указателя на функцию
C_D = (FType *)GetProcAddress(dllInstance,
"_Code_Dec");
else ShowMessage("Не удалось загрузить
'UMyDLL'");
}
```

```
void __fastcall
TForm1::Button1Click(TObject
*Sender)
{
if (C_D) Edit1->Text=C_D(Edit1-
>Text.c_str(),'A');
else
ShowMessage("Функция 'Code_Dec'
не доступна!");
}
```

```
void __fastcall TForm1::Button3Click(TObject
*Sender)
{//выгрузка DLL
FreeLibrary(dllInstance);
C_D=NULL;
}
```