**Создание и использование DLL на языке C++.**

В данном документе на конкретном примере рассмотрено создание динамической библиотеке и использование ее в консольном приложении C++ при использовании компилятора Microsoft Visual Studio 10.0.

**1.Создание DLL.**

Для создания библиотеки динамического вызова(**DLL**) необходимо создать проект типа “Библиотека DLL MFC”. При этом автоматически будет сгенерирован текст основной программы. Данный текст лучше не изменять. Для написания функции, которая будет помещена в библиотеку нужно включить в проект новый файл CPP – для этого выберите “ПРОЕКТ” – “Добавить новый элемент” –“ Файл C++(.cpp)” , указать его имя в соответствующем поле и выбрать “Добавить”. На экране появится окно для набора текста вашей функции. Обратите внимание, что первым оператором должен быть #include "stdafx.h".

Ниже приведен пример функции, вычисляющей одну из функций на указанном интервале и выводящей результат в виде таблицы:

#include "stdafx.h"

#include <math.h>

#include <stdio.h>

#define Sin 0

#define Cos 1

int SinCos(int Regime,double xn,double xk,double dx)

{

char function[2][20]={" Sin(x) "," Cos(x) "};

double x;

if(Regime != Cos && Regime != Sin) return 1;

if(xn >xk) return 2;

printf("\n------------------");

printf("\n| x |%s|",function[Regime]);

printf("\n--------------------");

for(x=xn; x <=xk+0.001; x+=dx)

{

if(Regime == Sin)

printf("\n|%7.2lf |%7.3lf |",x,sin(x));

else

printf("\n|%7.2lf |%7.3lf |",x,cos(x));

}

printf("\n--------------------");

return 0;

}

Далее необходимо открыть файл с расширением .def и в секцию EXPORTS добавить имя помещаемой в библиотеку функции и после ее имени указать ее номер, используя значок @.

Ниже приведен пример такого файла для функции SinCos, текст которой был приведен выше:

; SinCos.def: объявляет параметры модуля для DLL.

LIBRARY

EXPORTS

; Сюда можно направлять явные операции экспорта

SinCos @1

В дальнейшем эта функция может вызываться либо по имени, либо по номеру. В нашем примере будет рассматриваться вызов по имени.

Выберите пункт Построение – Построить.

Все. Библиотека создана (если не было ошибок).

**1.Использование DLL.**

Создайте проект “Консольное приложение” и напишите свою главную функцию.

Ниже приведен текст для рассматриваемого примера:

// UseSinCosDLL.cpp: определяет точку входа для консольного приложения.

//

#include "stdafx.h"

#include <math.h>

#include <stdio.h>

#include <windows.h> // для использования функций и типов данных для работы с DLL

#define Sin 0

#define Cos 1

typedef int (\_\_cdecl \*MyProc)(int Regime,double xn,double xk,double dx);

// определяем тип MyProc для дальнейшего использования

// int вначале означает тип возвращаемого значения

// (int Regime,double xn,double xk,double dx); определяет количество и типы параметров

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

{ double xn,xk,dx;

HINSTANCE hDLL; // Дескриптор копии DLL

MyProc SinCos; // адрес точки входа в вызываемую функцию

hDLL = LoadLibrary("SinCos.DLL");

// если DLL находится не в директории вызова приложения – необходимо указать полный путь (например, d:\\mydll\\SinCos.DLL”).

if(!hDLL) // если ошибка

{

printf("\n Library not loaded\n");

return 1;

}

SinCos=(MyProc)GetProcAddress(hDLL,"SinCos");

// получаем адрес загружаемой из DLL функции. Вторым параметром указано имя из файла // .def(см. выше).

while(1)

{

printf("\n Input xn,xk,dx\n");

if(scanf("%lf%lf%lf",&xn,&xk,&dx) == -1) break;

printf("\n Compute Sinus");

if(SinCos(Sin,xn,xk,dx)) //вызов нужной нам функции

{printf("\n Wrong calling\n");continue;}

printf("\n Compute Cosinus");

if(SinCos(Cos,xn,xk,dx)) {printf("\n Wrong calling\n");continue;}

}

return 0;

}

Внимание! Текст приведен для проекта с использованием многобайтовой кодировки. При использовании Юникода необходимо соответственно задавать текстовые константы для функций LoadLibrary и GetProcAddress.