УДК 004.432

Степович-Цветкова Г.С.
кандидат экономических наук, доцент,
ФГБОУ ВО «Ивановский государственный университет»,
Россия, г. Иваново

СОЗДАНИЕ МНОГОМОДУЛЬНОГО ПРИЛОЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕКИ ДИНАМИЧЕСКОЙ КОМПОНОВКИ НА ЯЗЫКЕ С++

Рассматриваются особенности создания многомодульного приложения, содержащего библиотеку динамической компоновки, на примере решения задачи о реализации нового типа данных.

Ключевые слова: многомодульные приложения, библиотеки функций, С++, DLL.

Структурированность программ повышает читабельность и возможность их последующего использования в решении новых задач. Использование многомодульных приложений, с одной стороны, приносит некоторые дополнительные сложности в создании таких приложений, но, с другой стороны, каждый разработанный модуль может быть использован многократно в других приложениях.

Для повторного использования созданного кода в других программах существует возможность создания библиотек функций и типов данных. Библиотеки подразделяются на библиотеки статической компоновки (LIB-файл) и библиотеки динамической компоновки (DLL). Разница между ними в том, что код, подключенный из статической библиотеки, становится частью нового приложения, а на код в библиотеке DLL необходимо ссылаться в новом приложении, что позволяет экономить место в каждом приложении, которое ссылается на этод код, и обновлять DLL-библиотеку без перекомпиляции всех приложений.

Рассмотрим процесс создания библиотеки динамической компоновки на примере решения задачи о создании типа данных «Трехмерный вектор» и перегрузке операций сложения и умножения. Определение нового типа данных осуществим с помощью структуры, операцию умножения положим эквивалентной скалярному произведению двух векторов. Определение типа и

реализацию перегруженных операций поместим в библиотеку динамической компоновки.

Необходимо создать новый проект типа консольное приложение Win32. При создании зададим проекту имя Vector, а решению имя DynamicLibrary. В параметрах приложения при создании в качестве типа приложения выберем DLL. В проекте автоматически будет создан и открыт файл Vector.cpp, в котором мы будем описывать реализацию функций. Однако еще требуется соответствующий заголовочный файл Vector.h, который нужно добавить в проект (меню Проект -> Добавить новый элемент -> в разделе Visual C++ выбираем Код, в центральной области – Заголовочный файл).

В файл Vector.h помещаем:

```
// Vector.h
#ifdef Vector_EXPORTS
#define Vector_API __declspec(dllexport)
#else
#define Vector_API __declspec(dllimport)
#endif
namespace MyVector
{
struct vector{  // объявление типа «Трехмерный вектор»
                float x,y,z; // координаты вектора
};
// перегрузка операции сложения
Vector_API vector operator+(vector a,vector b);
// перегрузка операции умножения
Vector_API float operator*(vector a,vector b);
}
```

Определяется пространство имен MyVector, которое содержит описание типа данных и объявления перегруженных функций. Vector_API применяется для осуществления экспорта/импорта из библиотеки DLL.

В файл Vector.cpp помещаем:

```
#include "stdafx.h"
#include "Vector.h"
#include <iostream>
using namespace std;
namespace MyVector
{
// перегрузка операции сложения
vector MyVector::operator+(vector a,vector b){
        vector result;
        result.x=a.x+b.x;
        result.y=a.y+b.y;
        result.z=a.z+b.z;
        return result;
}
// перегрузка операции умножения
float MyVector::operator*(vector a,vector b){
        return a.x*b.x+a.y*b.y+a.z*b.z;
}
}
```

Компилируем библиотеку динамической компоновки командой Собрать решение. Для работы с созданной библиотекой, на ее основе необходимо создать многомодульное приложение.

Создадим новый проект типа консольное приложение Win32 с именем Using_Vector. Данный проект добавим в существующее решение DynamicLibrary, содержащее DLL-библиотеку, указав «Добавить в решение» при создании. В процессе создания в параметрах приложения в разделе Дополнительные параметры необходимо снять флажок «Предкомпилированный заголовок». Автоматически будет создан Using_Vector.cpp.

Созданное приложение должно ссылаться на библиотеку статической компоновки. Для создания такой ссылки для проекта Using_Vector в контекстном меню в области Обозревателя решений необходимо выбрать Ссылки, в открывшемся диалоговом окне для узла Общие свойства -> .NET Framework и ссылки нужно выбрать Добавить новую ссылку, выбрав библиотеку Vector в качестве ссылки. Для создания ссылки на файл заголовка Vector.h необходимо изменить путь к каталогам включения. Для этого в диалоговом окне Окна свойств библиотеки Vector для узла Свойства конфигурации -> C/C++ -> Общие в поле Дополнительные каталоги включаемых файлов необходимо указать путь к каталогу Vector.

Таким образом, ссылка установлена, библиотека Vector доступна в приложении. В файл Using_Vector.cpp помещаем:

```
#include "stdafx.h"
#include "Vector.h"
#include <iostream>
using namespace std;
using namespace MyVector;
int main()
{
	vector a,b,c;
	cout<<"Vvedite koordinati vectora: ";
	cin>>a.x>>a.y>>a.z;
	cout<<"Vvedite koordinati vectora: ";
	cin>>b.x>>b.y>>b.z;
	c=a+b; // вызов перегруженной операции сложения
	cout<<"a+b=("<<c.x<<','<<c.y<<','<<c.z<<')';
	cout<<"\na*b="<<a*b; // вызов перегруженного умножения
	return 0;
}
```

Собираем решение (Сборка -> Собрать решение), и, назначив проект Using_Vector запускаемым по умолчанию, запускаем проект, результат работы программы будет следующим:

Vvedite koordinati vectora: 2 5 4

Vvedite koordinati vectora: 1 -3 5

a+b=(3,2,9)

a*b=7

Таким образом, при создании пользовательских типов данных удобно объявления и реализацию допустимых операций для работы с ними помещать в библиотеки функций, которые возможно использовать в многомодульных приложениях.

Список литературы

1. Степович-Цветкова, Г. С. Языки и технологии программирования. Ч. I : Структурное программирование на языке C++ [Электронный ресурс] : учебное пособие для студентов бакалавриата направлений "Математика", "Математика и компьютерные науки", "Фундаментальная информатика и информационные технологии", "Информационная безопасность" / Г.С. Степович-Цветкова. – Электрон. дан. – Иваново : Иван. гос. ун-т, 2016. – 1 электрон. опт. диск (CD-ROM).

2. Шилдт Г. Искусство программирования на C++. – СПб. : БХВ-Петербург, 2005. – 496 с.