Лекция 08
БГТУ, ФИТ, ПОИТ, 2 семестр, Языки программирования

Структура языка программирования. Статическая библиотека

План лекции:

- назначение и классификация библиотек;
- преимущества и недостатки статических библиотек;
- создание статических библиотек;
- использование статических библиотек.

1. Назначение библиотек:

предоставить стандартный простой и надёжный механизм повторного использования кода.

1.1. Использование:

- для использования функций из библиотеки в разных программах;
- при разработке большого проекта отлаженные функции помещают в библиотеку (время трансляции уменьшается).

1.2. Классификация библиотек:

- библиотеки на языках программирования (библиотеки классов, шаблонов, функций и т. п.). Компилируются вместе с исходными файлами проекта;
- библиотеки объектных модулей (статические библиотеки). Компилируются вместе с объектными файлами проекта;
- библиотеки исполняемых модулей (динамические библиотеки). Загружаются в память в момент запуска программы или во время ее исполнения, по мере надобности.

1.3. Статические библиотеки. Библиотеки для компилируемых языков

Статическая библиотека — файл с исходным кодом или объектный файл, предназначенный для вставки в программу на этапе компоновки.

Библиотеки, распространяемые в виде исходного кода, преобразуются компилятором в объектные файлы. Затем компоновщик соединяет объектные файлы библиотек и объектные файлы программы в один исполняемый файл.

Например, в исходных текстах распространяются:

- библиотеки для языка Fortran;
- библиотека Boost для языка C++.

Справка.
Библиотека BOOST — это набор C++ библиотек, созданных независимыми разработчиками, которые используются для создания проектов, предназначенных для реализации для различных платформ.

В стандарт C++ включена библиотека файловой системы, основанная на boost::filesystem. Спецификации для стандарта C++17 были опубликованы в декабре 2017 года.

Расширения объектных файлов модулей статических библиотек:

для ОС Microsoft Windows	**.lib**;
для ОС UNIX	**.a**.

1.4. Библиотеки для интерпретируемых языков

Библиотека — файл, содержащий либо код на интерпретируемом языке, либо байт-код для виртуальной машины.

Например, библиотеки для языка Python могут распространяться либо в виде файлов с исходным кодом (расширение «ру»), либо в виде файлов с байт-кодом (расширение «рус», (пайк) «ру» + буква «с» от англ. *compiled*).

2. **Статическая библиотека:**
файл (обычно с расширением **lib**), содержащий объектные модули;
входной файл для компоновщика (**linker**).

Достоинства:

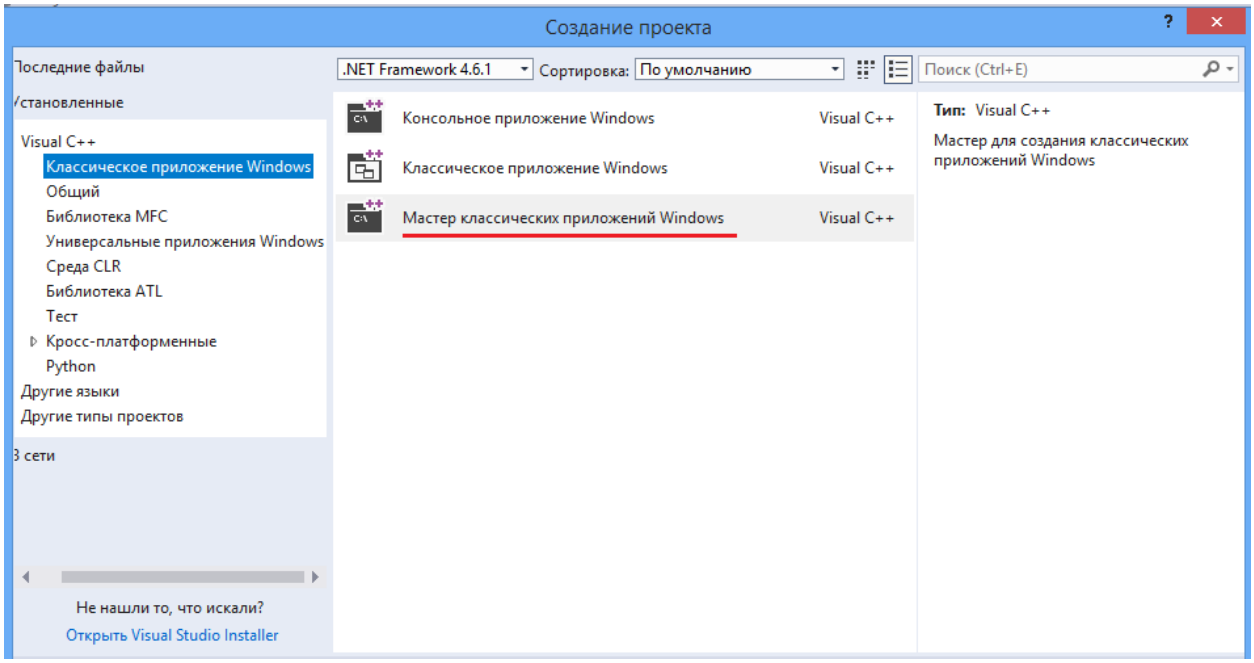
- просто использовать;
- не требуется наличие самой библиотеки;
- исполняемый файл один (расширение .exe).

Недостатки:

- платформа-зависима;
- загружается в память с каждым экземпляром запущенного приложения;
- при изменении кода библиотеки необходима компоновка всех приложений, которые используют библиотеку.

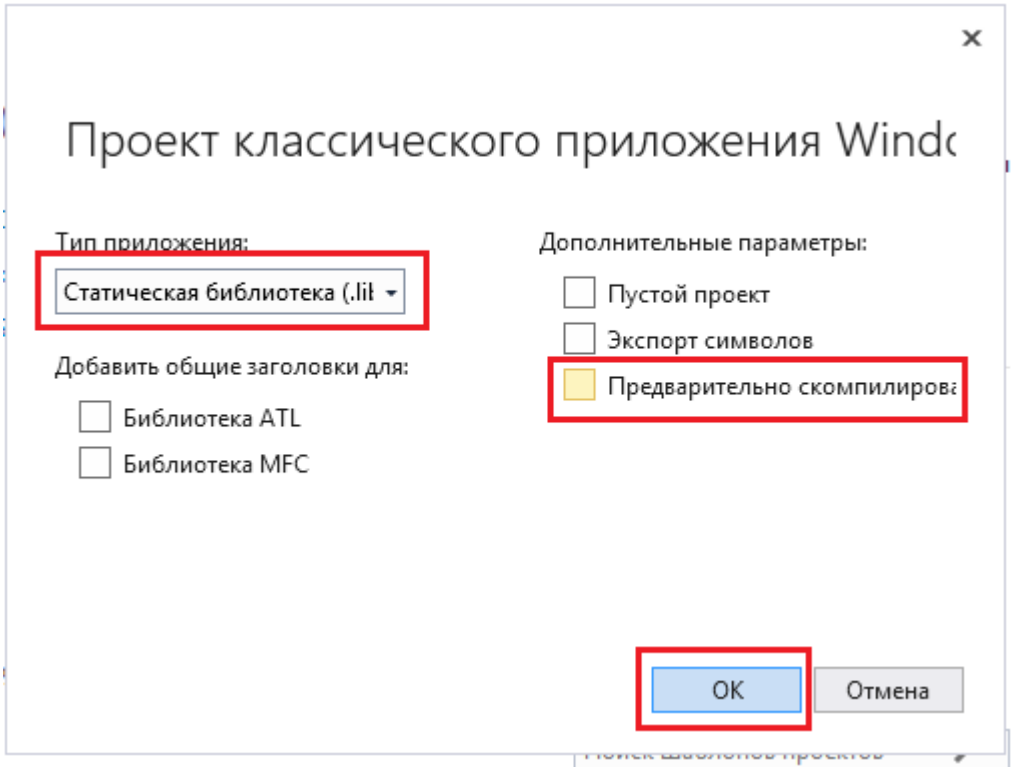
3. **Статическая библиотека Microsoft:**
файл с расширением **lib.**
Для работы с библиотекой предназначена утилита LIB.

Создание статической библиотеки с помощью Мастера классических приложений Windows:



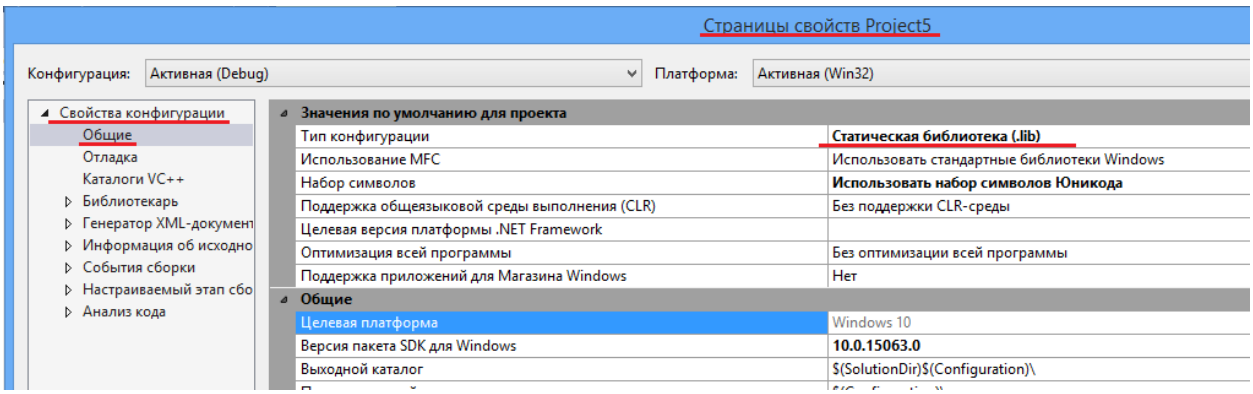
Не забываем определить имя решения, имя проекта и выбираем место размещения на диске.

Выбираем тип приложения «Статическая библиотека» (снимаем флажок «Предварительно скомпилированные заголовки» при необходимости):



В проект добавляем один или несколько файлов, содержащих реализации функций библиотеки.

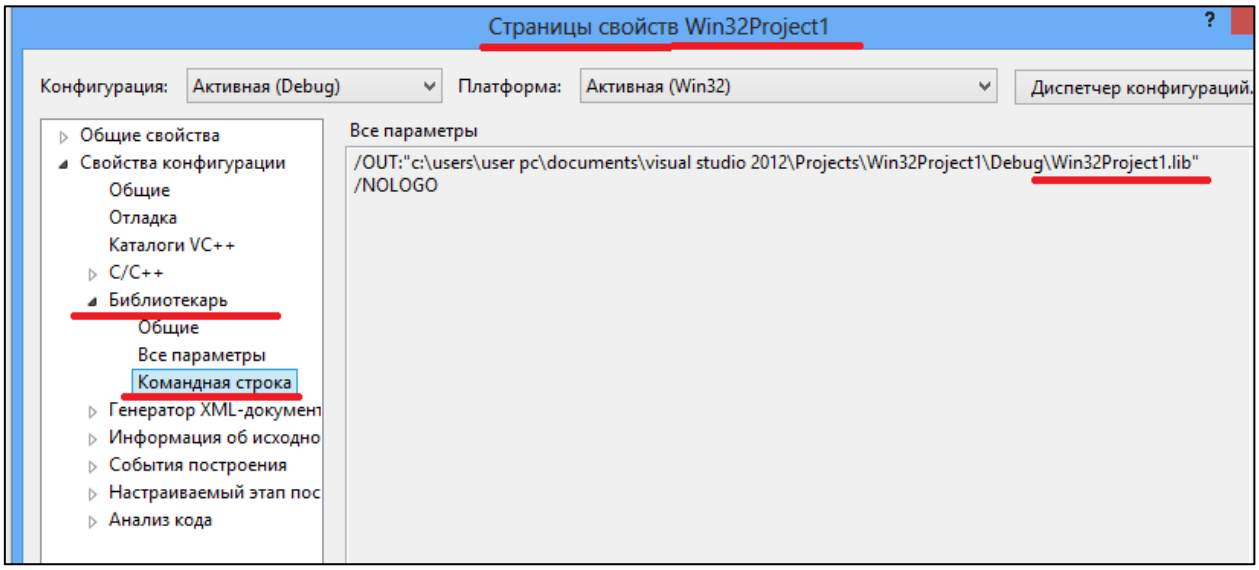
Страница свойств проекта:



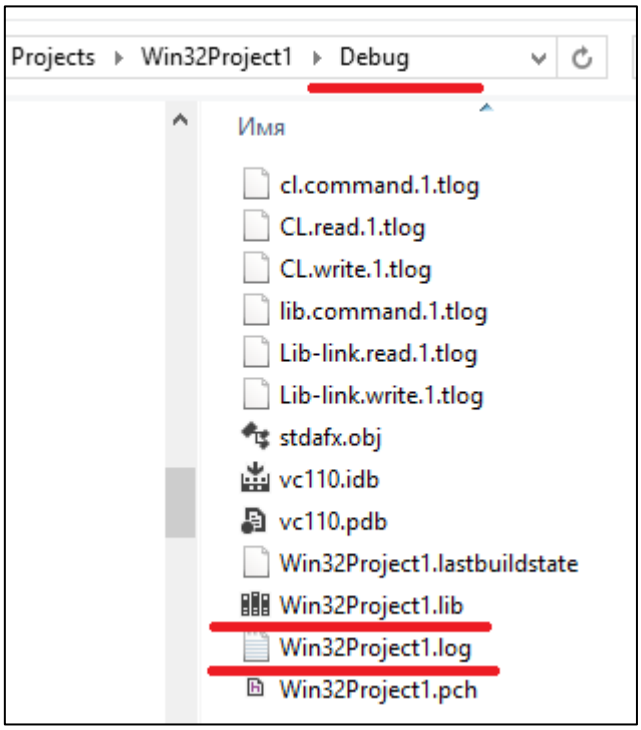
В пункте раздела «Библиотекарь» –> «Командная строка» отображается текущее значение параметра /OUT.

Расширение выходных файлов определено как .lib.

Директорий – папка проекта Debug.

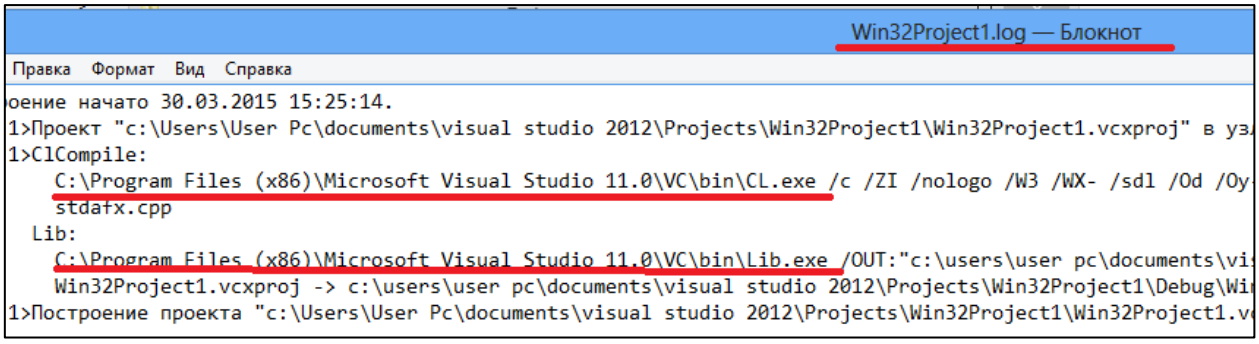


После построения проекта в папке Debug размещен файл статической библиотеки (.lib).



В журнале проекта зафиксировано выполнение сборки проекта.

Видим, что файл статической библиотеки создан утилитой LIB.



Утилиту LIB можно использовать в следующих режимах:

- построение или изменение библиотеки;
- извлечение элемента-объекта библиотеки в файл;
- создание файла экспорта и библиотеки импорта.

Эти режимы взаимоисключающие, LIB можно использовать только в одном режиме.

LIB принимает те или иные входные файлы в зависимости от режима использования.

4. Статическая библиотека. Параметры утилиты LIB.

Параметры LIB:
https://docs.microsoft.com/ru-ru/cpp/build/reference/overview-of-lib?view=vs-2019

/DEF

Создание библиотеки импорта и файла экспорта
Дополнительные сведения см. в разделе Построение библиотеки импорта и файла экспорта.

/ERRORREPORT

Передача Майкрософт сведений о внутренних ошибках с помощью lib.exe.
Дополнительные сведения см. в разделе Запуск программы LIB.

/EXPORT

Экспорт функции из программы.
Дополнительные сведения см. в разделе Построение библиотеки импорта и файла экспорта.

/EXTRACT

Создание объектного файла (OBJ-файла), содержащего копию элемента существующей библиотеки.
Дополнительные сведения см. в разделе Извлечение члена библиотеки.

/INCLUDE

Добавление символа в таблицу символов.

Дополнительные сведения см. в разделе Построение библиотеки импорта и файла экспорта.

/LIBPATH

Переопределяет путь к библиотеке среды.

Дополнительные сведения см. в разделе Управление библиотекой.

/LIST

Отображает информацию о выходной библиотеке в стандартном виде.

Дополнительные сведения см. в разделе Управление библиотекой.

/LTCG

Инициирует построение библиотеки с помощью создания кода времени компоновки.

Дополнительные сведения см. в разделе Запуск программы LIB.

/MACHINE

Задание целевой платформы для программы.

Дополнительные сведения см. в разделе Запуск программы LIB.

/NAME

При построении библиотеки импорта указывает имя библиотеки DLL, для которой была создана библиотека импорта.

Дополнительные сведения см. в разделе Управление библиотекой.

/NODEFAULTLIB

Удаляет одну или несколько библиотек по умолчанию из списка искомых библиотек при разрешении внешних ссылок.

Дополнительные сведения см. в разделе Управление библиотекой.

/NOLOGO

Отключает вывод программой LIB уведомления об авторских правах и номере версии, а также отображение команд командного файла.

Дополнительные сведения см. в разделе Запуск программы LIB.

/OUT

Переопределяет имя выходного файла используемое по умолчанию.

Дополнительные сведения см. в разделе Управление библиотекой.

/REMOVE

Пропуск объекта из выходной библиотеки.

Дополнительные сведения см. в разделе Управление библиотекой.

/SUBSYSTEM

Сообщает операционной системе способ запуска программы, созданной путем привязки к выходной библиотеке.

Дополнительные сведения см. в разделе Управление библиотекой.

/VERBOSE

Отображает подробные сведения о ходе сеанса, включая имена добавляемых OBJ-файлов.

Дополнительные сведения см. в разделе Запуск программы LIB.

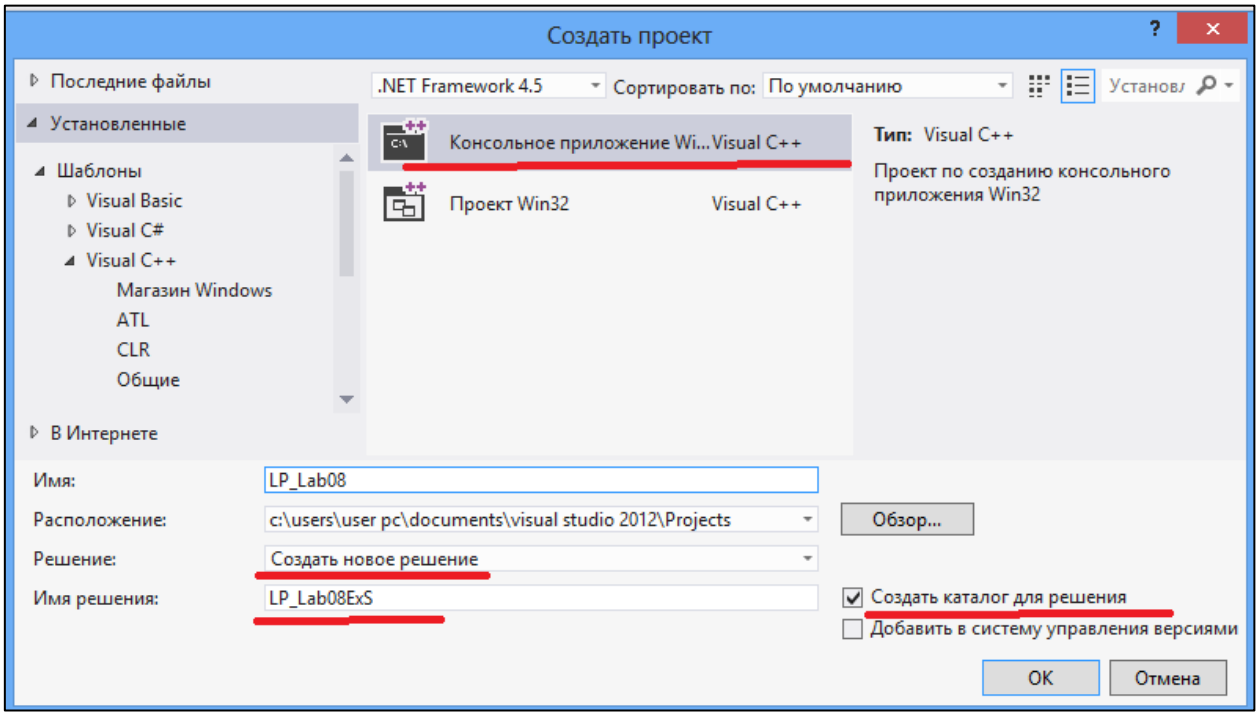
/WX

Обработка предупреждений, как ошибок.
Дополнительные сведения см. в разделе Запуск программы LIB.

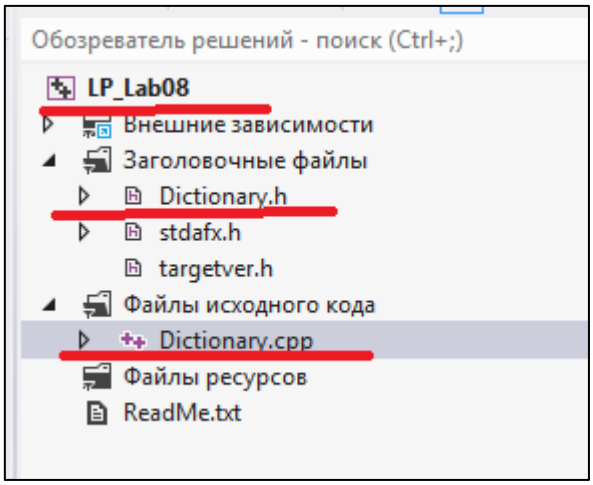
5. Статическая библиотека.

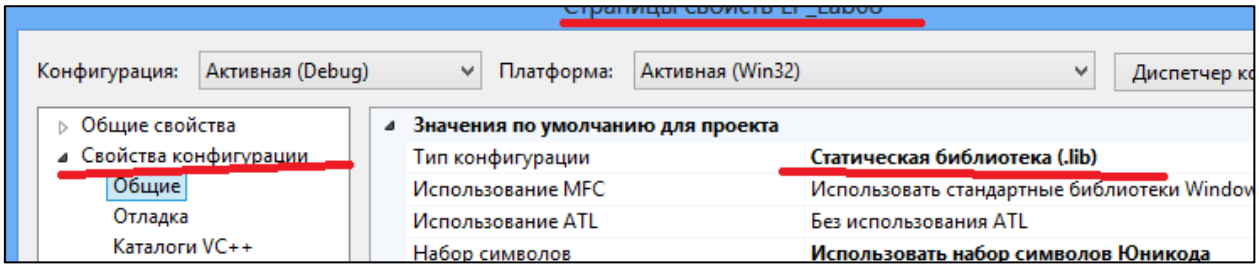
Пример применения в C++.

Создадим проект с именем LP_Lab08 решения LP_Lab08Ex8, который является статической библиотекой:

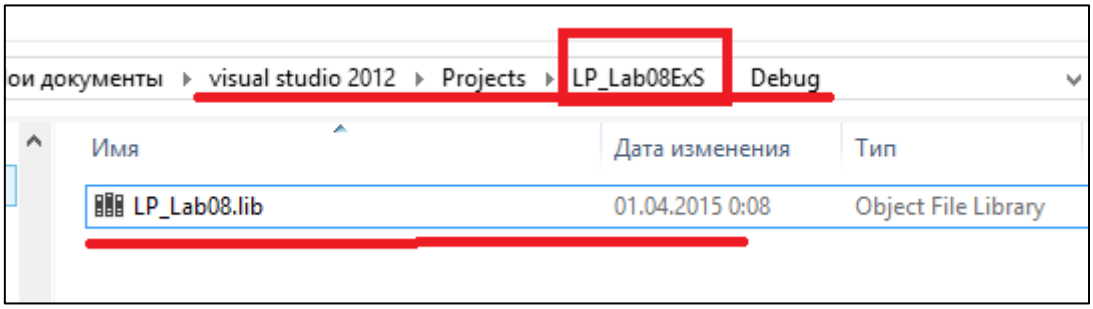


Добавим в проект заголовочный файл и файл исходного кода библиотеки.



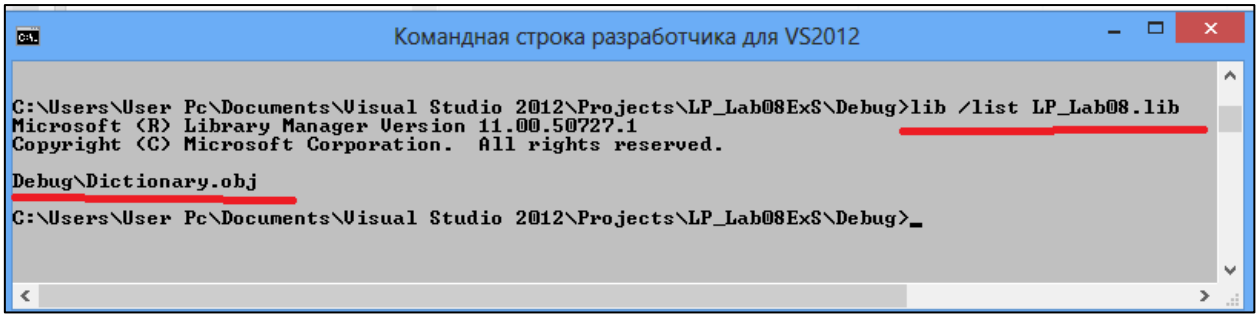


Файл статической библиотеки размещен в папке Debug:

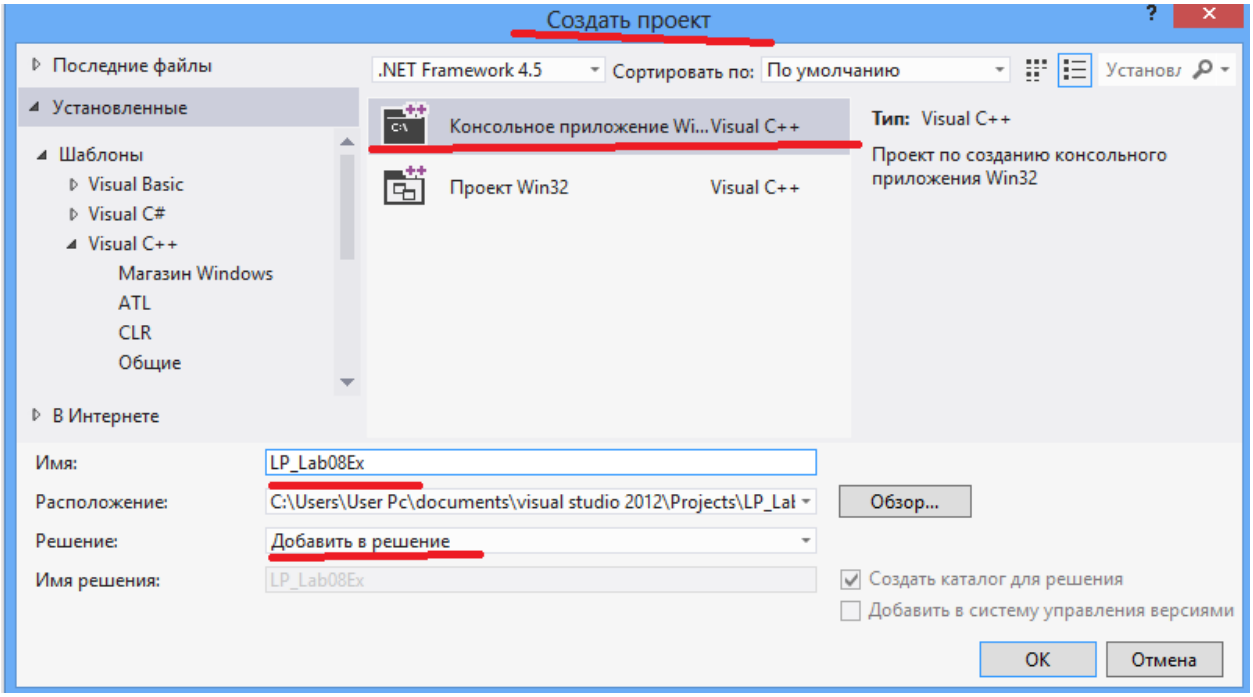


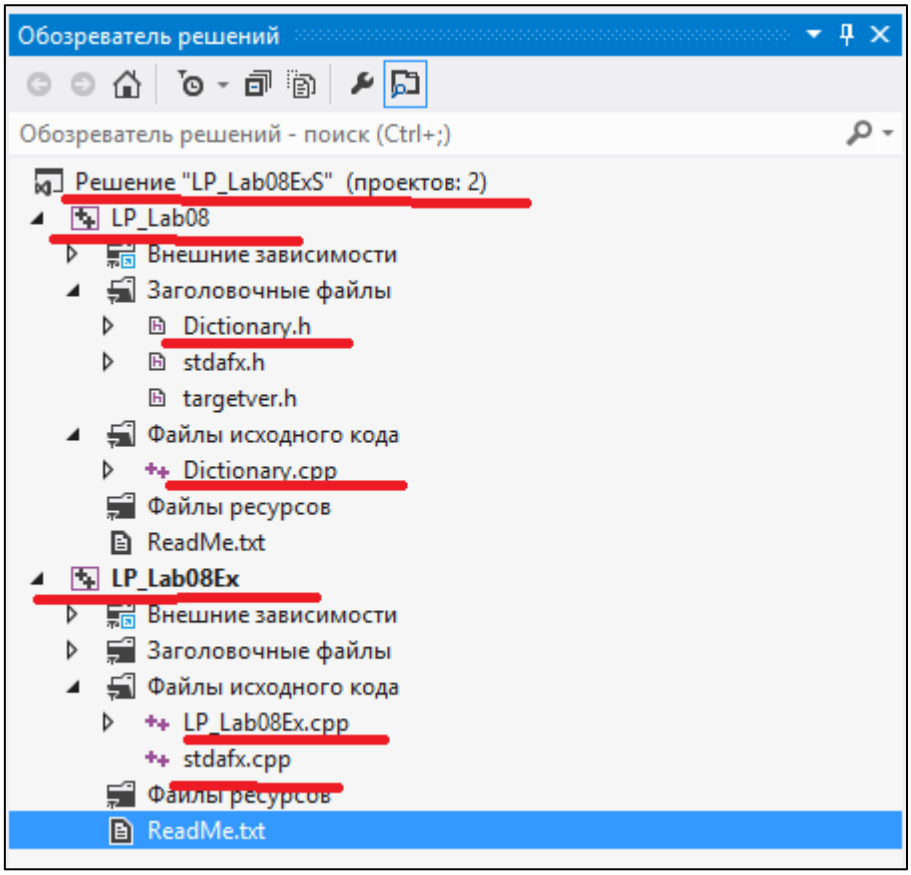
В командной строке разработчика с помощью утилиты **LIB** с ключом /LIST можно получить перечень obj-модулей, содержащихся в LIB-файле.

В статической библиотеке один объектный модуль с именем Dictionary.obj:

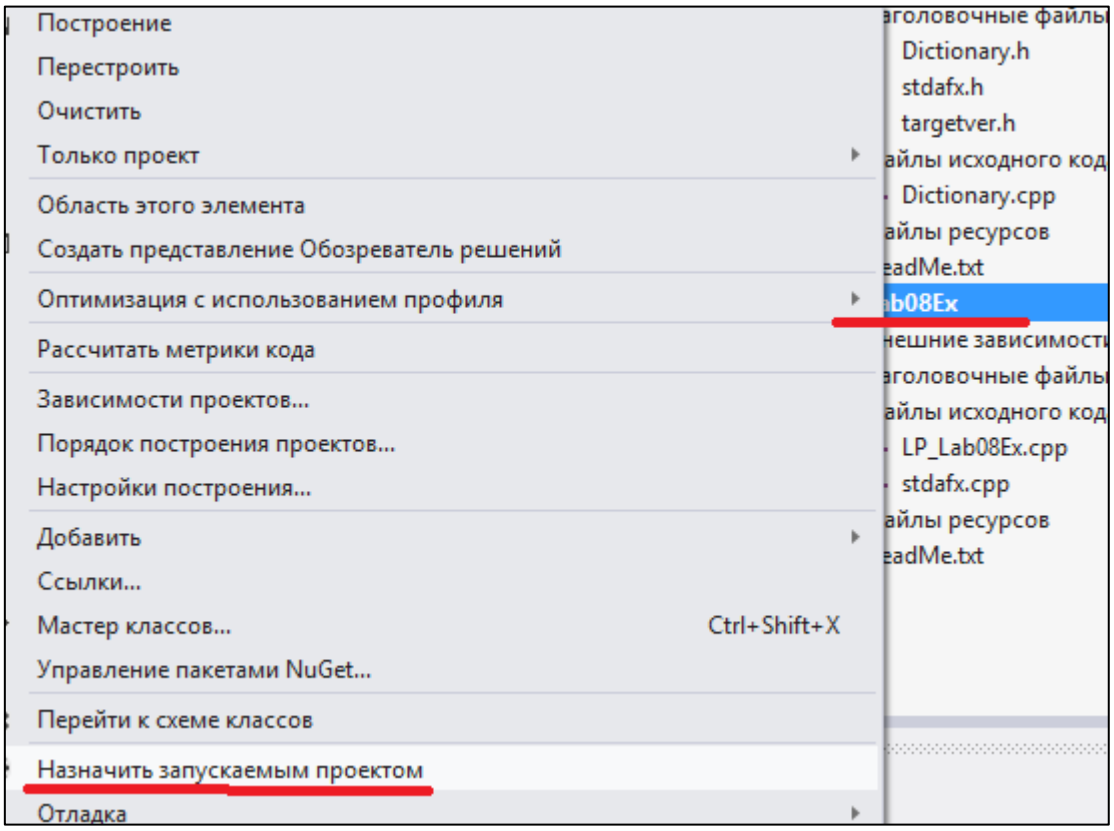


Добавляем в решение новый проект LP_Lab08Ex – консольное приложение, которое будет использовать созданную нами библиотеку.

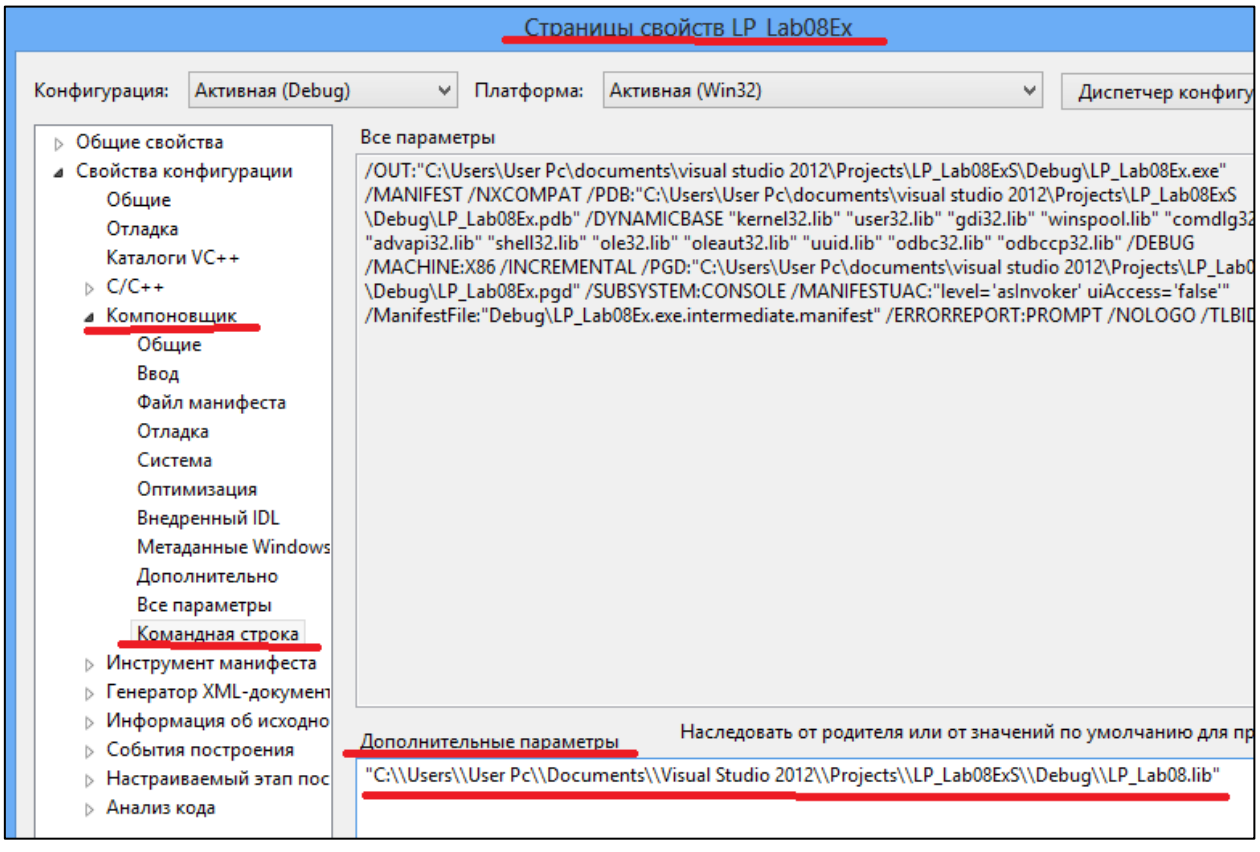




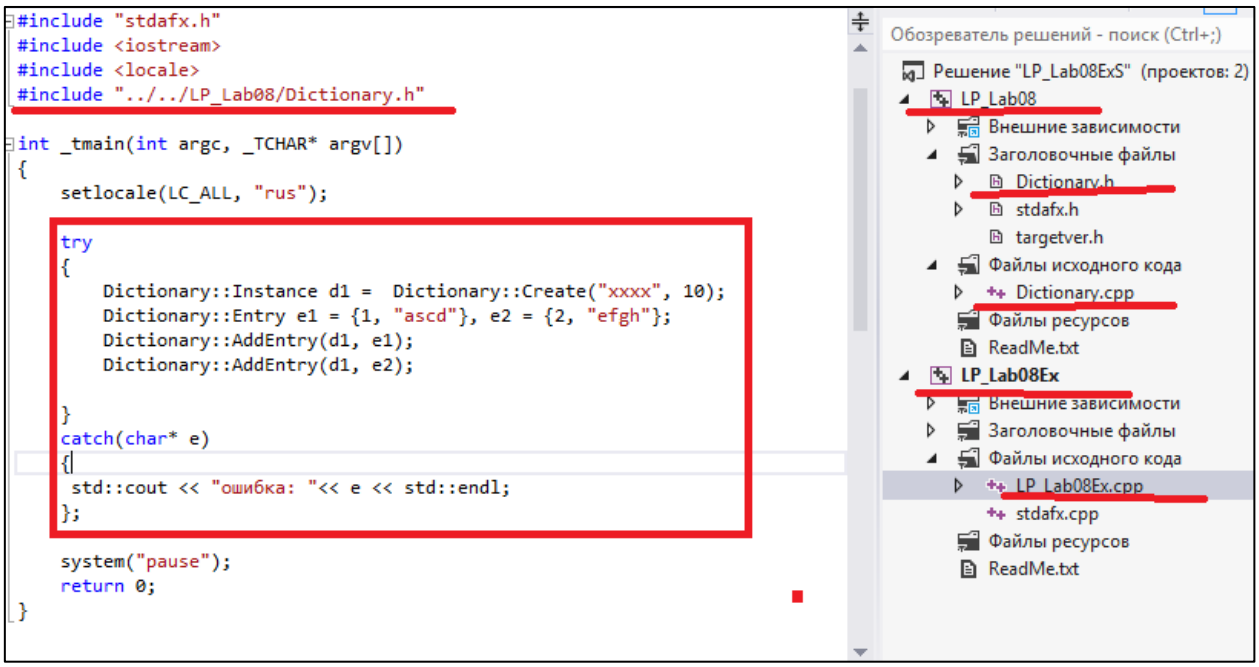
Назначаем созданный проект автозапускаемым.



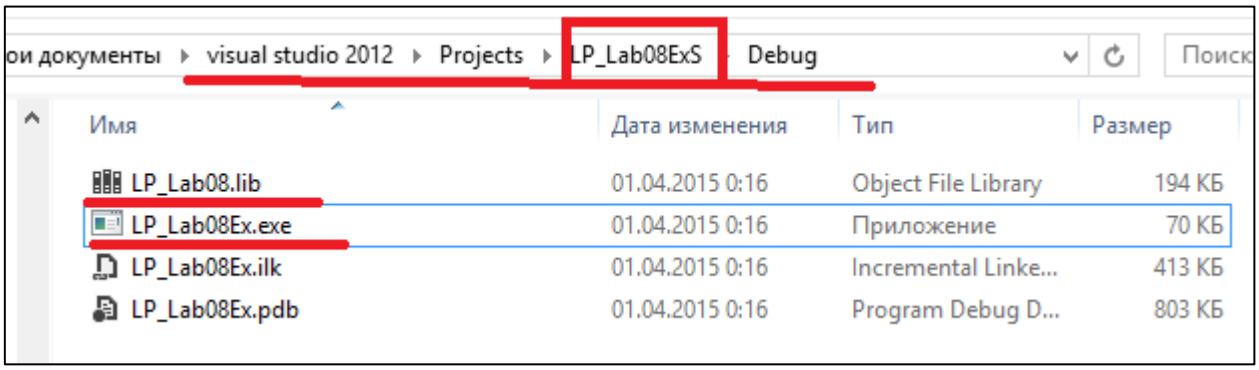
На странице свойств проекта в разделе «Компоновщик» -> Командная строка в окне «Дополнительные параметры» вводим параметр, указывающий имя, местоположение статической библиотеки.



Подключаем в главную функцию заголовочный файл библиотеки:



Получаем исполнимый файл LP_Lab08Ex:



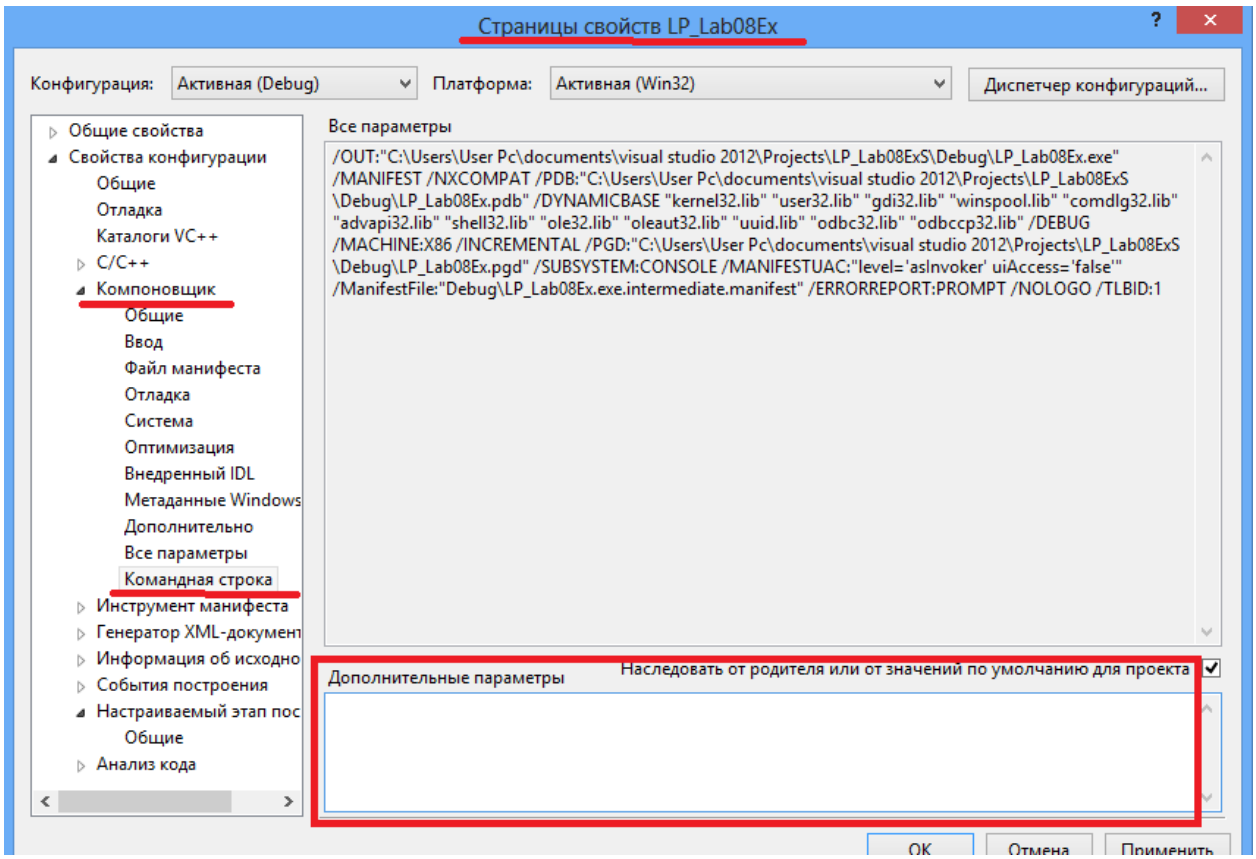
6. Статическая библиотека: директива #pragma comment

Автоматическое связывание (от англ. *Auto-linking*) — механизм автоматического определения необходимых библиотек при сборке программ на языках C/C++.

Активируется при помощи строки вида:

#pragma comment(lib, <название>)

В этом случае для подключения статической библиотеки дополнительные параметры командной строки компоновщика не используются:



Статическая библиотека подключается при помощи строки

#pragma comment(lib, <путь\\имя_библиотеки.lib>),

включенной в главную функцию:

```
3
4  #include "stdafx.h"
5  #include <iostream>
6  #include <locale>
7  #include "../../LP_Lab08/Dictionary.h"
8
9  #pragma comment(lib, "C:\\Users\\User Pc\\Documents\\Visual Studio 2012\\Projects\\LP_Lab08ExS\\Debug\\LP_Lab08.lib" )
10
11 int _tmain(int argc, _TCHAR* argv[])
12 {
13     setlocale(LC_ALL, "rus");
14
15     try
16     {
17         Dictionary::Instance d1 =  Dictionary::Create("xxxx", 10);
18         Dictionary::Entry e1 = {1, "ascd"}, e2 = {2, "efgh"};
19         Dictionary::AddEntry(d1, e1);
20         Dictionary::AddEntry(d1, e2);
21
22     }
23     catch(char* e)
24     {
25      std::cout << "ошибка: "<< e << std::endl;
26     };
27
28     system("pause");
29     return 0;
30 }
31
```