

# ООП

Семестр 2. Лекция 6.

Создание библиотек

Развёртывание приложений

Кафедра ИВТ и ПМ  
ЗабГУ

2018

# План

Создание и использование библиотек

Развёртывание

windeployqt

Дополнительно

Ссылки и литература

# Outline

Создание и использование библиотек

Развёртывание

windeployqt

Дополнительно

Ссылки и литература

- ▶ Динамические

- ▶ Загружает при запуске программы либо во время её работы
- ▶ Это отдельный файл: dll - windows; so - UNIX; dylib - Mac OS

- ▶ Статические

- ▶ Добавляется в исполняемый файл (программу) на этапе компиляции
- ▶ lib - windows; a - UNIX;

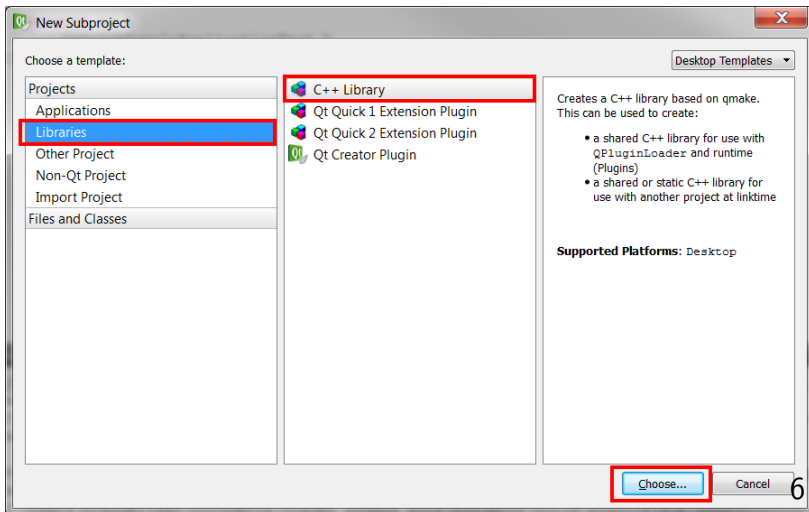
# Подключение динамических библиотек

- ▶ Во время запуска программы  
В дополнение к скомпилированной библиотеке используется заголовочный файл библиотеки.
- ▶ Во время выполнения программы. Дополнительно объявляются используемые идентификаторы библиотеки из библиотеки.  
Фреймворки часто включают в себя классы для удобного динамического подключения библиотек.

# Создание библиотек в Qt Creator

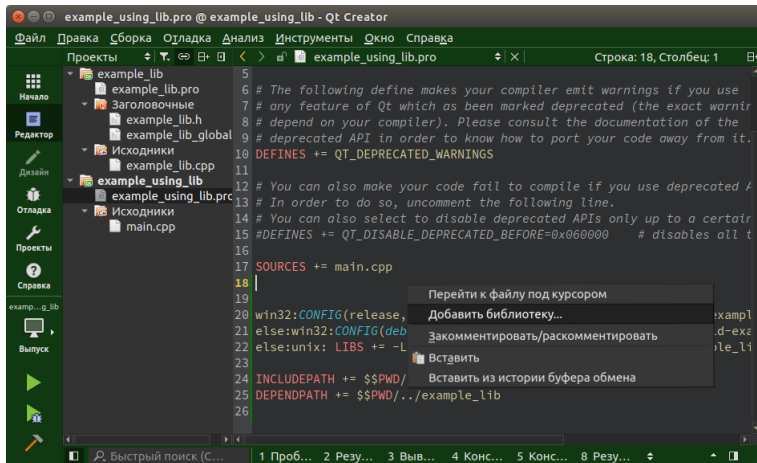
В Qt Creator существуют шаблоны для создания динамических и статических библиотек.

Автоматически будут созданы заголовочный и src файл.



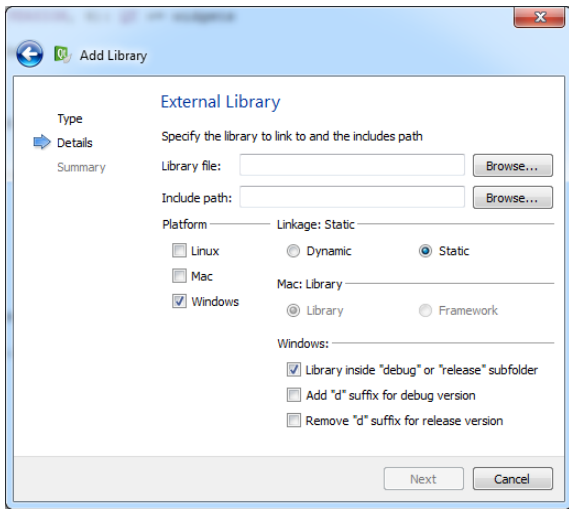
# Использование сторонних библиотек в Qt Creator

Добавление готовой библиотеки - правый клик в pro файле -> добавить библиотеку... -> внешняя.



# Использование сторонних библиотек в Qt Creator

Для внешней библиотеки требуется указать путь к заголовочным файлам и непосредственно к файлу библиотеки.





# Использование сторонних библиотек в Qt Creator

Qt Creator после выполнения этих действий автоматически добавит дополнительные строки в pro файл:

```
win32:CONFIG(release, debug|release):
    LIBS += -L$$PWD/../../build-example_lib_Release/release/ -lexample_lib
else:win32:CONFIG(debug, debug|release):
    LIBS += -L$$PWD/../../build-example_lib_Release/debug/ -lexample_lib
else:unix:
    LIBS += -L$$PWD/../../build-example_lib_Release/ -lexample_lib

INCLUDEPATH += $$PWD/../../example_lib
DEPENDPATH += $$PWD/../../example_lib
```

Так как в проект добавлены пути поиска заголовочных файлов, то подключать их можно так, как если бы они были расположены в каталоге проекта.

# Использование сторонних библиотек в Qt Creator

См. пример `example_lib`: [github.com/VetrovSV/OOP](https://github.com/VetrovSV/OOP)

Другие примеры: [How to link to a dll](#)

# Динамическое подключение библиотеки

Для динамического подключения библиотеки в Qt используется класс `QLibrary`.

см. пример: [QLibrary: Пример использования](#)

Об использовании классов из динамически подключенных библиотек:

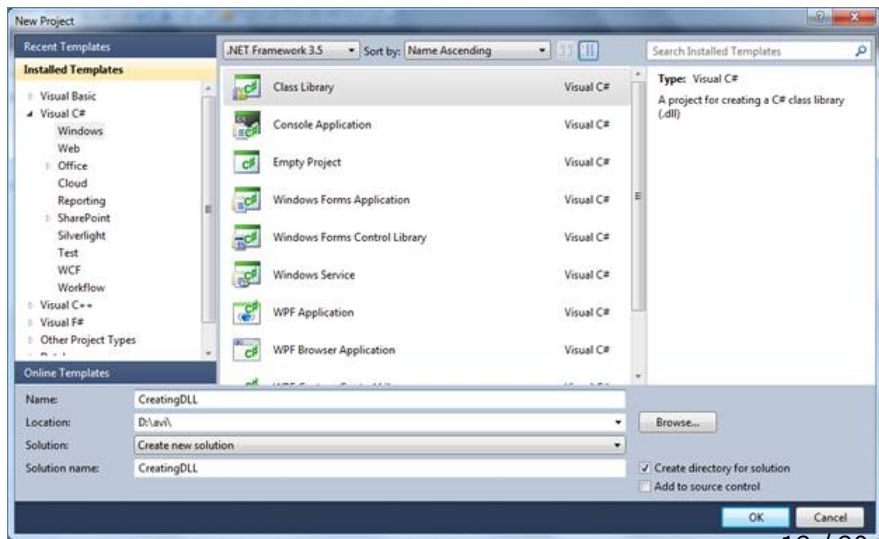
<https://stackoverflow.com/questions/26234327/qlibrary-import-a-class>

Вне зависимости от способа подключения динамической библиотеки конечное приложение должно распространяться вместе с этой библиотекой (dll или so файлом).

При использовании же статической библиотеки, её код будет включен в исполняемый файл приложения.

# Библиотеки в .NET

**Создание dll.** Для создания динамической библиотеки используется шаблон проекта **"Class Library"**



# Библиотеки в .NET

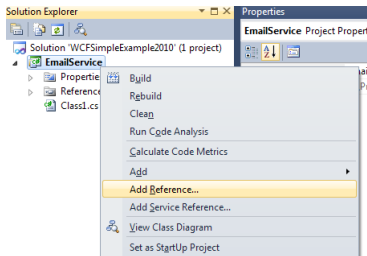
## Подключение dll

Контекстное меню проекта -> добавить -> ссылку...

или

Пункт *Ссылки* в проекте -> добавить ссылку...

Затем указать на конкретный dll файл.



Далее можно подключить библиотеку как любой другой модуль с помощью оператора using.

# Управляемый и неуправляемый код

Библиотеки, созданные с использованием фреймворка .NET отличаются от аналогичных, например созданных на чисто C++ или с использованием Qt.

Исходный код использующий .NET (например на C) не компилируется в машинный код, а переводится в байт-код, называемый **управляемым кодом**.

Управляемый код не может быть выполнен сам по себе, а только с помощью виртуальной машины .NET.

- ▶ Создание и использование библиотеки DLL в Visual Studio.  
[msdn.microsoft.com/ru-ru/library/ms235636.aspx](https://msdn.microsoft.com/ru-ru/library/ms235636.aspx)
- ▶ Примеры использования DLL в разных языках  
[ru.wikibooks.org/wiki/DLL](https://ru.wikibooks.org/wiki/DLL)
- ▶ what does `__declspec(dllimport)` really mean?  
[stackoverflow.com/questions/8863193/what-does-declspecdllimport-really-mean](https://stackoverflow.com/questions/8863193/what-does-declspecdllimport-really-mean)
- ▶ Использование dll скомпилированных на C++ в C:  
[stackoverflow.com/questions/16332701/how-to-call-c-dll-in-c-sharp](https://stackoverflow.com/questions/16332701/how-to-call-c-dll-in-c-sharp)



# Outline

Создание и использование библиотек

Развёртывание

windeployqt

Дополнительно

Ссылки и литература

Разработка приложения должна заканчиваться поставкой готового продукта или очередной его версии.

Программа должна быть самодостаточной и поставляются со всеми необходимыми библиотеками или, если библиотеки сторонние, предоставить пользователю инструкцию по их установке.

В идеальном случае программа не должна сваливать работу по поиску и установке отдельных компонентов на пользователя.

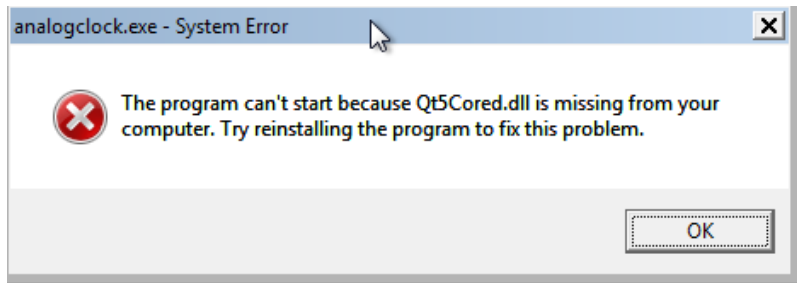
Обязательным этапом перед отправкой программы заказчику должно быть тестирование.

Причем тестирование должно происходить уже не в среде программирования и не в ОС разработчика.

ОС разработчика может содержать множество установленных библиотек которые могут понадобится приложению, а на ОС конечных пользователей таких может не оказаться.

Для окончательного тестирования программы можно использовать, например, виртуальную машину.

Чаще всего после компиляции программы для релиза (release) и запуска вместо работающей программы можно увидеть нечто подобное.



# Outline

Создание и использование библиотек

Развёртывание

**windeployqt**

Дополнительно

Ссылки и литература

Для подготовки Qt приложений к развёртыванию (распространению) используется программа **windeployqt** входящая в состав Qt.

- ▶ Копирует все необходимые библиотеки в каталог со скомпилированным exe файлом.
- ▶ Это консольная утилита<sup>1</sup>.
- ▶ Расположена по адресу QTDIR/bin/windeployqt.
- ▶ В качестве аргумента принимает путь к каталогу с exe файлом.

---

<sup>1</sup>рекомендуется использовать PowerShell или git bash для работы ▶

## Подготовка

Откомпилированная в режиме отладки программа содержит дополнительную информацию для отладки и использует специальный отладочный набор библиотек. Как правило такой исполняемый файл работает медленнее, больше по объёму и использует библиотеки большого объёма.

Поэтому перед использованием `windeployqt` нужно скомпилировать программу в режиме выпуска (release).

Чтобы `windeployqt` поместил помимо библиотек Qt в каталог ещё и необходимые библиотеки компилятора, например mingw (g++) путь к нему нужно добавить в переменную `path`.

# Использование

1. Скомпилировать в режиме выпуска
2. Переместить exe файл в отдельную<sup>2</sup> папку
3. запустить windeployqt:

```
windeployqt --compiler-runtime <path-to-app-binary>
```

---

<sup>2</sup>рекомендуется использовать именно отдельную, чтобы не засорять каталог сборки



# Outline

Создание и использование библиотек

Развёртывание

windeployqt

Дополнительно

Ссылки и литература

# Упаковка исполняемых файлов

**Упаковка исполняемых файлов** - сжатие исполняемого файла и прикреплении к нему кода, необходимого для распаковки и выполнения содержимого файла.

см. [UPX](#)

# Dependency Walker

Dependency Walker - программа для получения списка импортируемых и экспортируемых функций PE-файла<sup>3</sup>

The screenshot shows the Dependency Walker application window for the file STOOGES.EXE. The left pane displays a dependency tree where STOOGES.EXE depends on LARRY.DLL, which in turn depends on KERNEL32.DLL, NTDLL.DLL, and CURLY.DLL. CURLY.DLL depends on SHEMP.DLL, which depends on MOE.DLL and KERNEL32.DLL. MOE.DLL also depends on NTDLL.DLL. The right pane shows two tables: one for imported functions (PI^) and one for exported functions (E^). The bottom pane lists loaded modules with their file time stamps, link time stamps, file sizes, attributes, link checksums, real checksums, CPU architectures, and subsystems.

PI^	Ordinal	Hint	Function	Entry Point
	N/A	N/A	IsKnucklehead	Not Bound
	N/A	N/A	int SaySoitenly(char *,...)	Not Bound

E^	Ordinal	Hint	Function	Entry Point
	4 (0x0004)	1 (0x0001)	int SaySoitenly(char *,...)	SHEMP.?SaySoitenly@
	5 (0x0005)	2 (0x0002)	DoinkLarrysEye	0x00001010
	3 (0x0003)	0 (0x0000)	void SayPoifect(_int64)	0x00001020
	1 (0x0001)	N/A	N/A	0x00001020
	2 (0x0002)	3 (0x0003)	DoinkMoesEye	SHEMP.DoinkMoesEye

	Module ^	File Time Stamp	Link Time Stamp	File Size	Attr.	Link Checksum	Real Checksum	CPU	Subs
	CURLY.DLL	11/14/2006 5:17p	11/14/2006 5:13p	2,560	A	0x0000F739	0x0000F759	x86	GUI
	KERNEL32.DLL	08/30/2006 1:22a	08/30/2006 1:20a	871,424	A	0x000E388E	0x000E388E	x86	Cons
	LARRY.DLL	11/14/2006 5:13p	11/14/2006 5:13p	2,560	A	0x000053DB	0x000053DB	x86	GUI
	MOE.DLL	11/14/2006 5:15p	11/14/2006 5:15p	2,560	A	0x0000B191	0x0000B191	x86	GUI
	NTDLL.DLL	08/30/2006 1:23a	08/30/2006 1:21a	1,147,664	A	0x00125FA5	0x00125FA5	x86	Cons
	SHEMP.DLL	11/14/2006 5:13p	11/14/2006 5:13p	2,560	A	0x00001CE7	0x00001CE7	x86	GUI

00:00:00.093: LoadLibraryA("Moe.dll") called from "STOOGES.EXE" at address 0x00401024 by thread 1.

# Outline

Создание и использование библиотек

Развёртывание

windeployqt

Дополнительно

Ссылки и литература

# Ссылки и литература

- ▶ [Qt for Windows - Deployment](#)
- ▶ [habr.com/post/262211](https://habr.com/post/262211) - Развёртывание приложений Qt в Windows

см. также:

- ▶ [Docker](#)
- ▶ [youtube.com/watch?v=IxsK4ouFVBc](https://youtube.com/watch?v=IxsK4ouFVBc) - Создание инсталляционного пакета для Windows приложения (VS 2010)
- ▶ [marketplace.visualstudio.com](https://marketplace.visualstudio.com) - Microsoft Visual Studio 2017 Installer Projects

# Материалы курса

Слайды, вопросы к экзамену, задания, примеры

[github.com/VetrovSV/OOP](https://github.com/VetrovSV/OOP)