Курс: «Проектирование программ в интеллектуальных системах»

Цель курса:
научить вас строить программное обеспечение интеллектуальных систем с использованием современных методологий и языков программирования.

Основная методология:
объектно-ориентированное программирование.

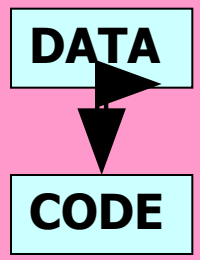
Основные языки: C++ и Java.

Программирование – это не только кодирование, но и поддержка всех этапов жизненного цикла программного продукта , который содержит такие этапы, как

- формирование требований (спецификаций);
- проектирование;
- реализация (кодирование);
- тестирование (отладка);
- внедрение;
- эксплуатация и сопровождение.

Объектное программирование

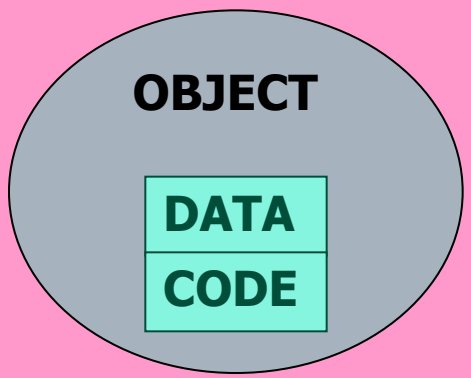
Было:



Процедурное программирование

Объектно-ориентированное

Стало:



Объект: некоторая структура, которая включает в себя данные и код, который взаимодействует с этими данными.

Для создания объектов используют специальные структуры данных, называемые *классами*.

```
class <имя класса>
{
      [описание данных]
      [описание кода]
};
```

Класс – тип данных, объект – экземпляр класса:

<имя класса> <объект1>, <объект2>,…

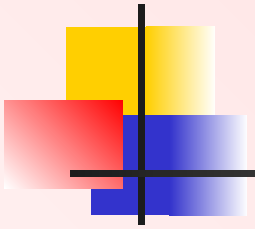
Вычисление в ООП рассматривается как модель поведения

Принципы объектного программирования

Инкапсуляция: пакетирование данных – механизм языка программирования, который позволяет объединять данные и код, взаимодействующий с этими данными. *Инкапсуляция* – это способность скрывать внутренние детали при предоставлении открытого интерфейса к определяемому пользователем типу данных.

Наследование: это механизм, который позволяет определить новые классы в терминах существующих классов. Может быть единичное наследование и множественное, когда новый класс наследует свойства сразу нескольких базовых.

Полиморфизм: способность различных объектов по разному обрабатывать одинаковые сообщения. Полиморфизм – это свойство, позволяющее использовать одно и то же имя для обработки различных типов данных.



Концепции объектного программирования

Моделирование деятельности мира

Наличие типов, определяемых пользователем

Сокрытие деталей реализации

Повторное использование кода через наследование

Разрешение интерпретации вызова функции во время выполнения

Некоторые из концепций неясны, некоторые абстрактны, а некоторые обобщены. Тем не менее они образуют фундамент иной точки зрения на программирование!

История языка C++

C++ - преемник C. Он был изобретен в фирме Bell Labs Бьярном Страуструпом в середине 1980 года.

Решались две основные задачи:

1) *Сделать C++ совместимым со стандартным C;*
2) *Расширить C конструкциями ООП (Объектно-ориентированного программирования)*

Истоки ООП – итоги 25 летнего опыта и практики программирования.
Впервые сформулированы в языке Simula 67 и развивались в таких языках, как
Smalltalk, Lisp, Clu,
а еще позже –
Actor, Eiffell, Objective C и C++

Конечная цель C++ - обеспечить язык для профессионального программиста, который может быть использован для создания объектно-ориентированного программного обеспечения, без ущерба для эффективности и компактности C

C++ лучше C

- C++ - преемник C. Он сохраняет большую часть этого языка: состав операторов, выразительный стиль, мобильность, расширяемость, эффективность (быстродействие программ) и экономичность (не расходуются ресурсы на динамическую проверку типов и сборку мусора) реализации.
- C++ совершенствует C по разным направлениям, особенно с точки зрения строгой типизации данных, и в силу этого является более безопасным языком.
- C++ - гибридный язык. Он позволяет писать программы как на низком, так и на высоком уровнях.
- В C++ возможна перегрузка операторов, поддерживающая реализацию новых пользовательских типов
- В C++ для удобного управления динамически распределяемой памяти введены два специальных оператора – new и delete
- Не смотря на то, что работа с указателями в C++ сохраняется в том же объеме, что и для языка C, классы обеспечивают удовлетворительные средства простой реализации обобщенных массивов.

Язык С - Ключевые слова

Слова, которые имеют специальное значение для компилятора языка. В языке Си используются следующие ключевые слова:

auto	break	case	char
const	continue	default	do
double	else	enum	extern
float	for	goto	if
int	long	register	return
short	signed	sizeof	static
struct	switch	typedef	union
unsigned	void	volatile	while

Вы не можете переопределять ключевые слова у себя в программе. В рамках конкретной системы программирования состав ключевых слов может быть расширен.

Ключевые слова языка C++ (дополнительно к C)

Ключевые слова языка C++ (дополнительно к C)

class	delete	friend	inline
new	operator	private	protected
public	this	template	virtual

Операции

:: - операция разрешения контекста;

.* и ***->**** - операции обращения через указатель к компоненте класса

new и ***delete*** – операции динамического выделения и освобождения памяти

Типы данных – те же, что и в C + классы и ссылки

Передача аргументов по умолчанию

Пространство имен – **using namespace std;**

Организация Windows-программы

- Компьютер
  - Операционная система
    - Программа пользователя

Как они взаимодействуют?

Любое приложение в Windows = Функция инициализации (WinMain) + функция обслуживания приложения (WinProc)

Основное назначение WinMain связано с решением 3 основных задач:

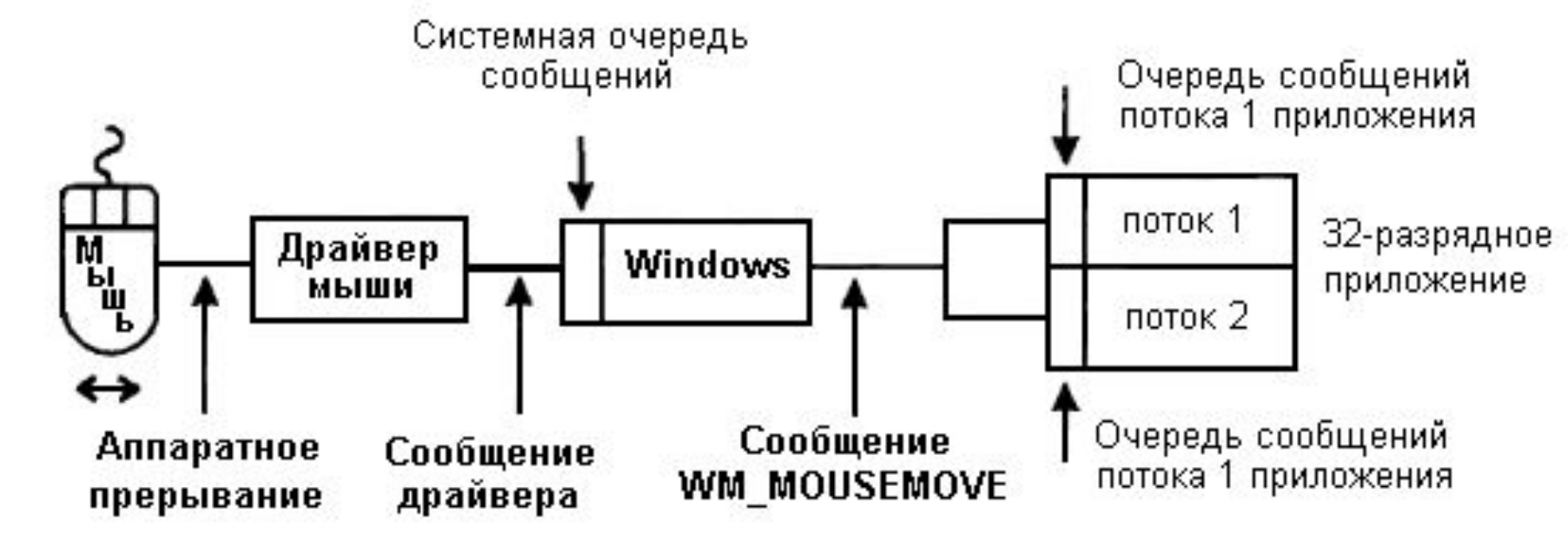
– зарегистрировать в Windows класс главного окна приложения (всех используемых окон)
– создать главное окно и показать его на экране
– организовать цикл обнаружения поступающих в приложение сообщений

Функция обслуживания приложения WinProc – обработка присланных сообщений.

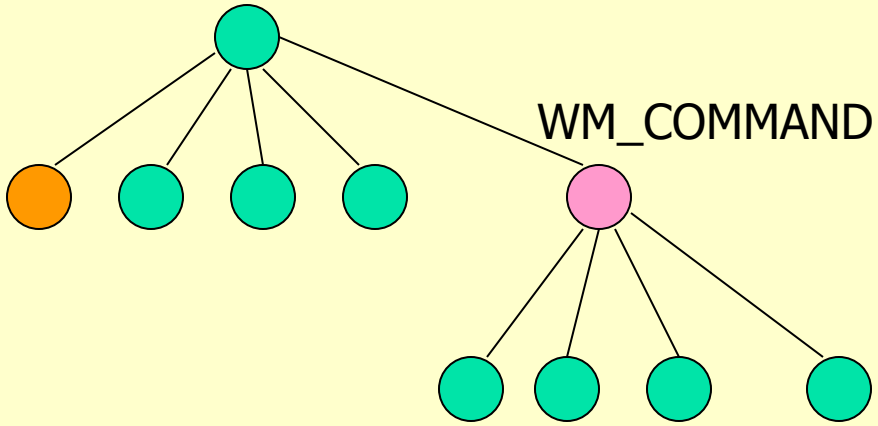
Сообщения являются реакцией системы Windows на различные происходящие в системе события: движение мыши, нажатие клавиши, срабатывание таймера и др. Отличительной особенностью сообщений является их специальный код. Для системных сообщений зарезервировано значение кода от 1 до 0x3FF.

Организация обработки сообщений

Процедура создания и пересылки сообщений от мыши:



Примеры сообщений:
WM_LBUTTONDOWN, WM_TIMER, WM_SIZE, WM_ERASEBKGND, WM_COMMAND.



Событийное управление

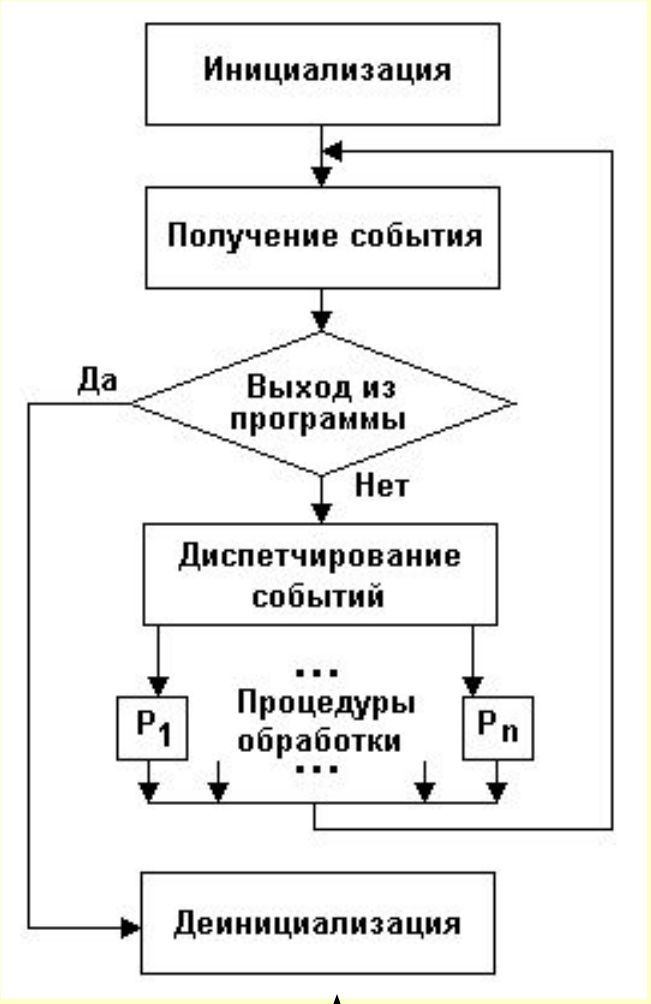
Прием сообщения – *событие*.

Характеристики события в Windows:

- Получатель (кому адресовано)
- Тип, связанный с источником
- Время возникновения
- Положение на экране курсора мыши
- Дополнительные параметры, специфичные для каждого конкретного случая

Цикл обработки сообщений организуется при помощи функций ***GetMessage*** (***PeekMessage***) и ***DispatchMessage***.

При создании приложений в среде VS'98 – цикл организуется автоматически и не требует от программиста каких-либо действий



Программа с событийно-управляемой архитектурой

Распределение оперативной памяти

Оперативная память в рамках операционной системы организована в виде нескольких списков – различия между кодом программы и ее данными нет!

Динамическая память – на самом деле способ ее распределения.

В языке C – захват памяти при помощи специальных функций: *malloc*, *calloc*, *realloc*. Освобождение ранее захваченной памяти – *free*.

В языке C++ - можно использовать еще и специальные операторы:

new и *delete*.

Фрагментация памяти – разбиение пространства памяти на множество мелких кусочков, что может приводить к существенному замедлению работы программы.

Компоненты среды программирования Visual Studio 98

Одним из языков программирования, поддерживаемых средой VS'98 является язык C++.

Разработка программ осуществляется в рамках интегрированной среды разработки IDE (*Integrated development environment*).

Данная среда объединяет такие компоненты как:

Компилятор C/C++ (Compiler).

Редактор исходного текста (Editor).

Компилятор ресурсов (Resource compiler).

Компоновщик (Linker).

Отладчик (Debugger).

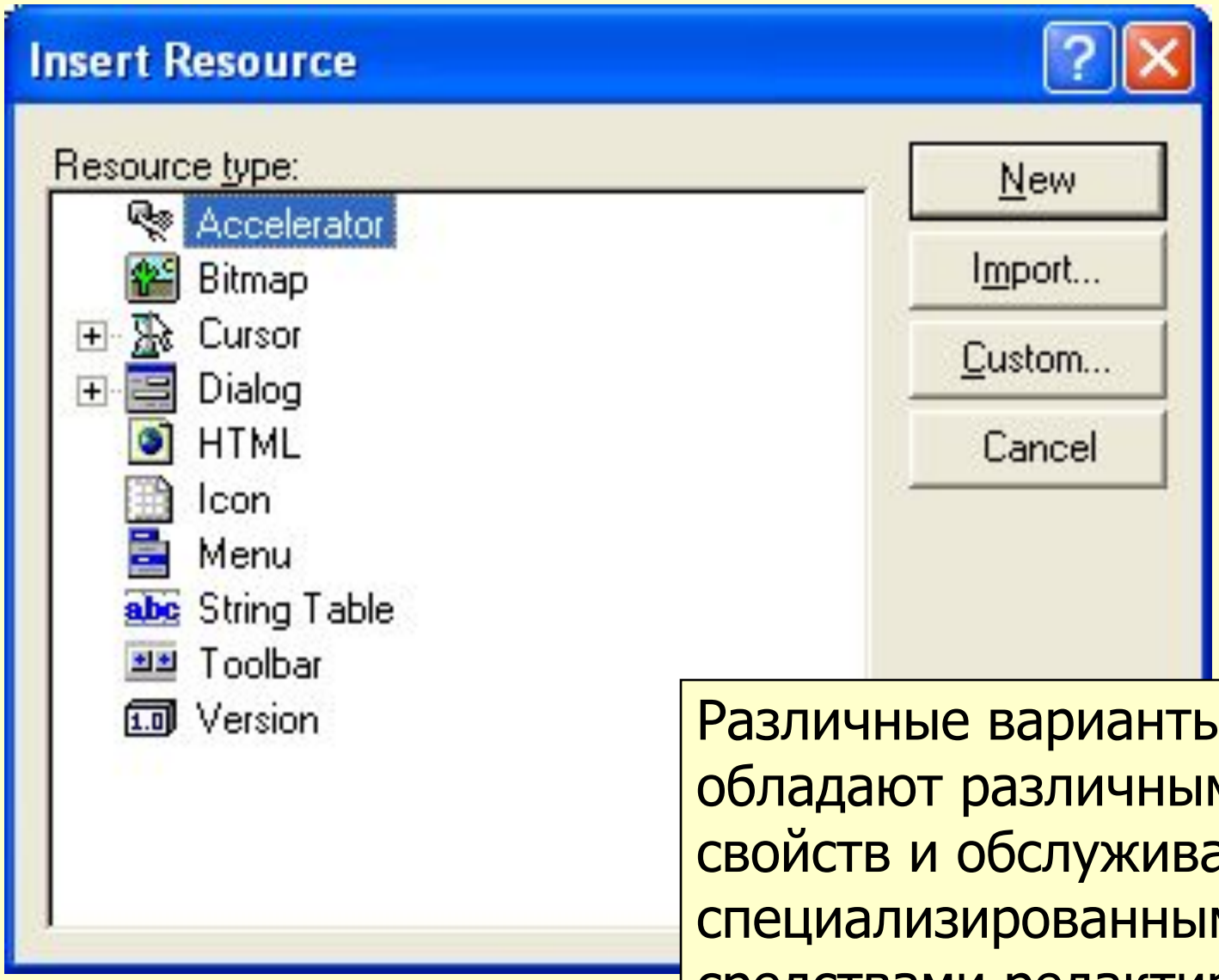
Генератор кода (AppWizard).

Поддержка ведения классов (ClassWizard).

Справочная система (MSDN).

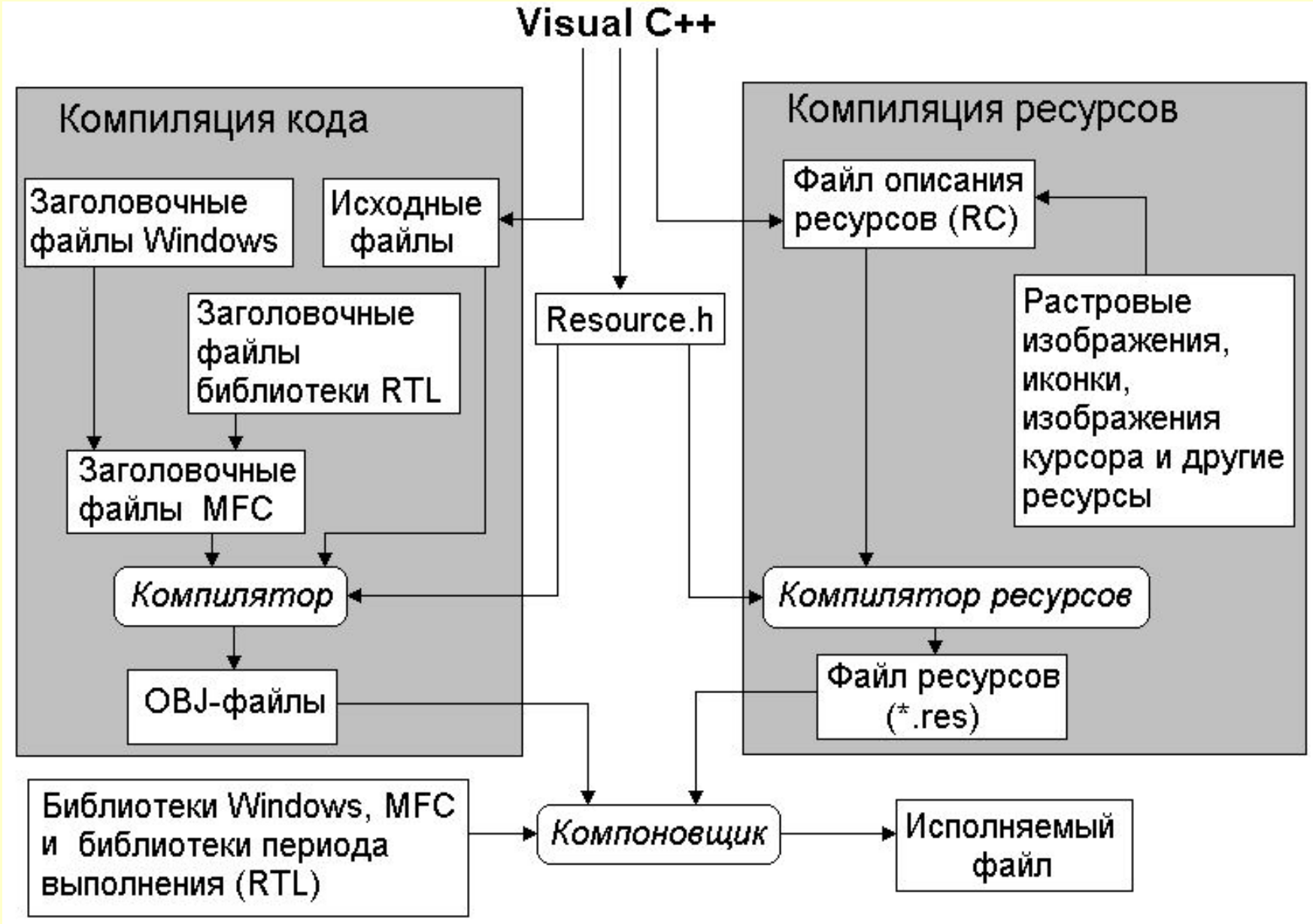
Библиотека классов (MFC).

Понятие ресурсов

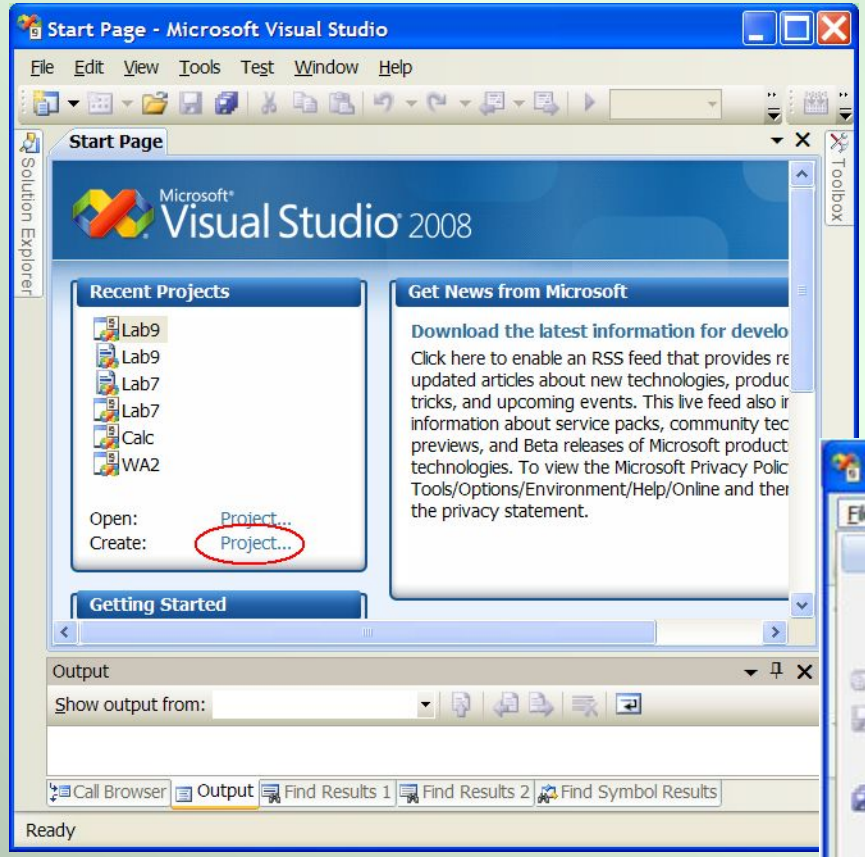


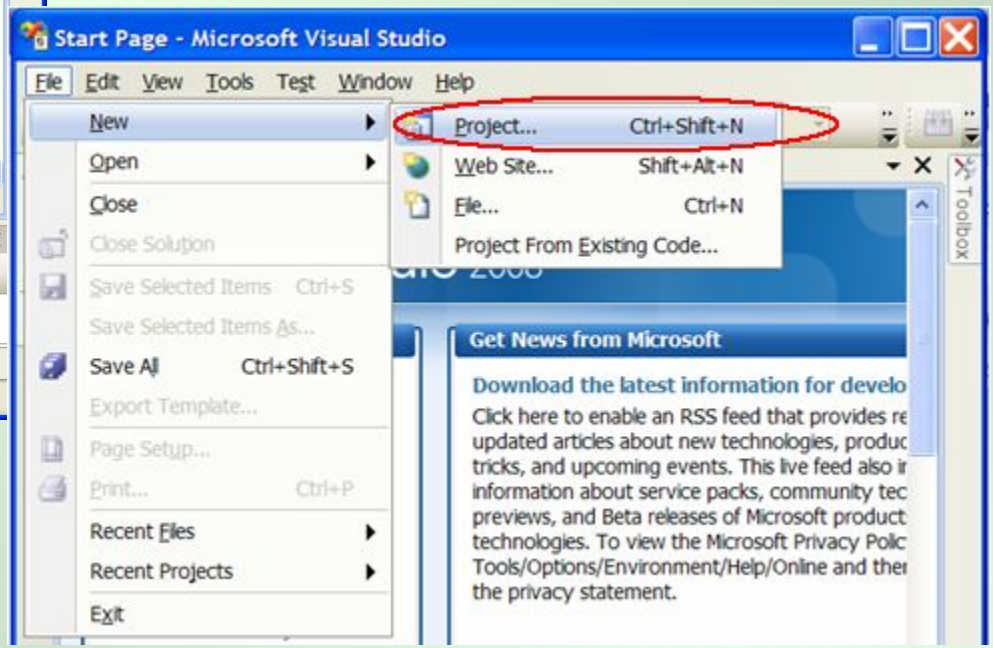
Различные варианты ресурсов обладают различными наборами свойств и обслуживаются специализированными средствами редактирования

Процесс построения программ

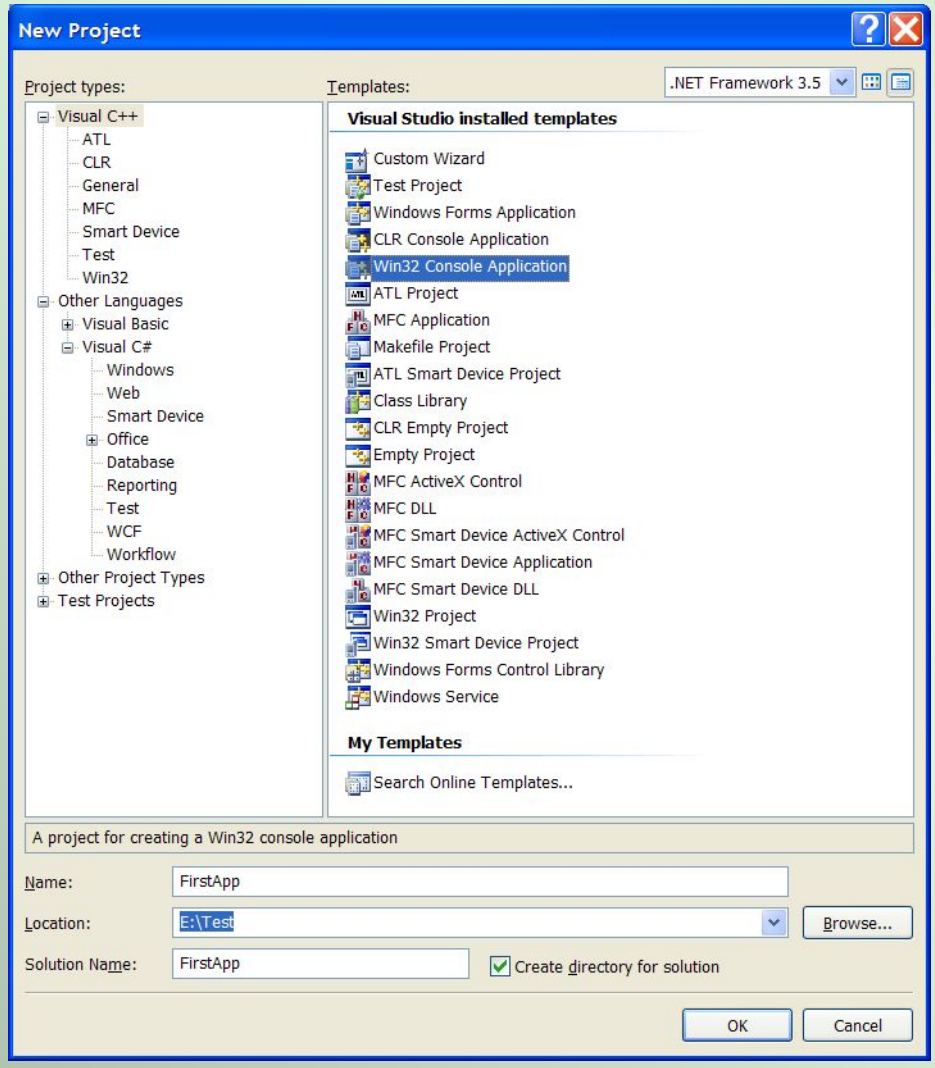


Начало 1


Start Page - Microsoft Visual Studio
File Edit View Tools Test Window Help
Start Page
Solution Explorer
Toolbox
Microsoft Visual Studio 2008
Recent Projects
Lab9
Lab9
Lab7
Lab7
Calc
WA2
Open: Project...
Create: Project...
Get News from Microsoft
Download the latest information for develo
Click here to enable an RSS feed that provides re
updated articles about new technologies, produc
tricks, and upcoming events. This live feed also ir
information about service packs, community tec
previews, and Beta releases of Microsoft product
technologies. To view the Microsoft Privacy Polic
Tools/Options/Environment/Help/Online and ther
the privacy statement.
Getting Started
Output
Show output from:
Call Browser
Output
Find Results 1
Find Results 2
Find Symbol Results
Ready


Start Page - Microsoft Visual Studio
File Edit View Tools Test Window Help
New
Open
Close
Close Solution
Save Selected Items Ctrl+S
Save Selected Items As...
Save All Ctrl+Shift+S
Export Template...
Page Setup...
Print... Ctrl+P
Recent Files
Recent Projects
Exit
Project... Ctrl+Shift+N
Web Site... Shift+Alt+N
File... Ctrl+N
Project From Existing Code...
Toolbox
Get News from Microsoft
Download the latest information for develo
Click here to enable an RSS feed that provides re
updated articles about new technologies, produc
tricks, and upcoming events. This live feed also ir
information about service packs, community tec
previews, and Beta releases of Microsoft product
technologies. To view the Microsoft Privacy Polic
Tools/Options/Environment/Help/Online and ther
the privacy statement.

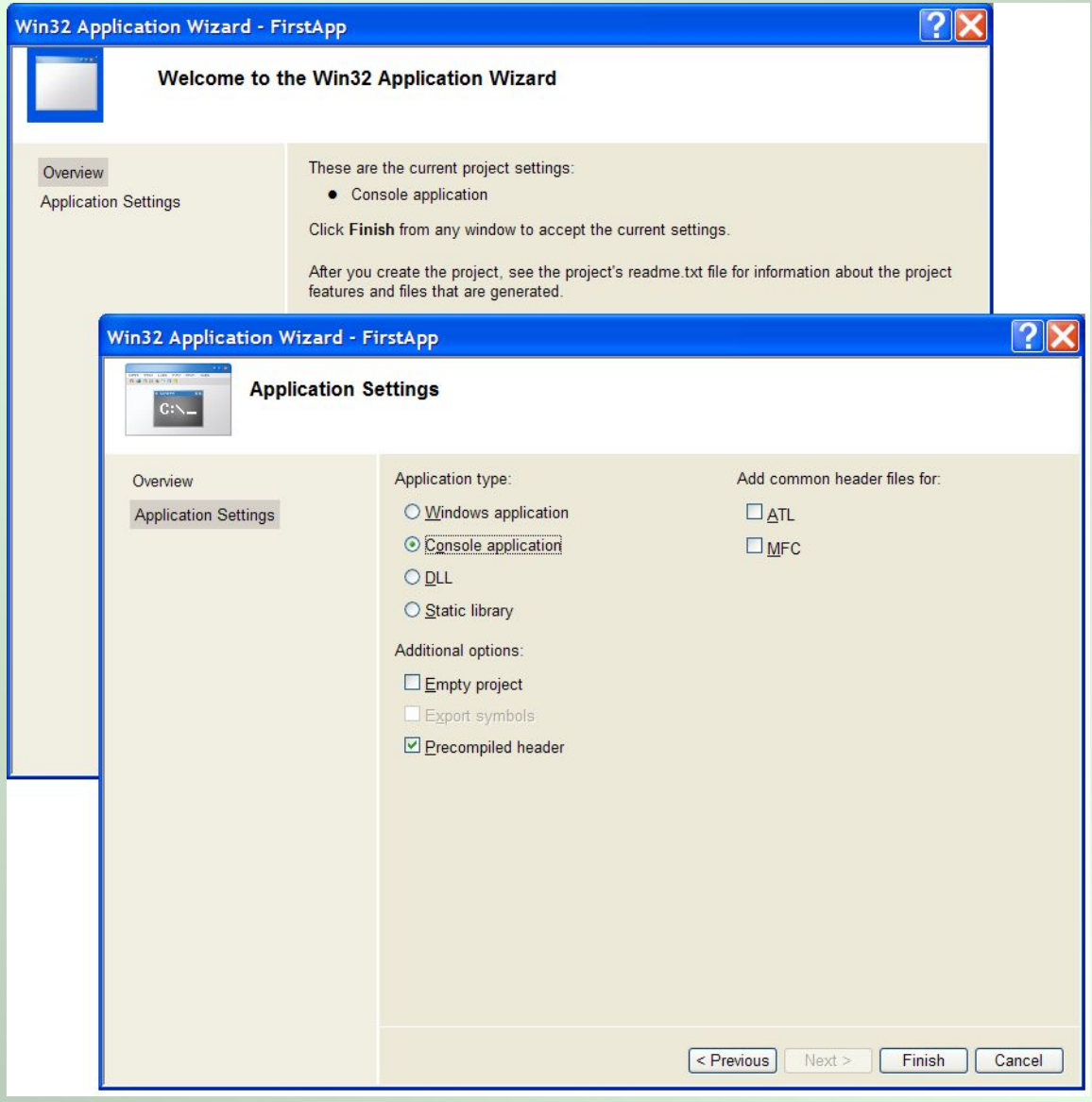
Начало 2



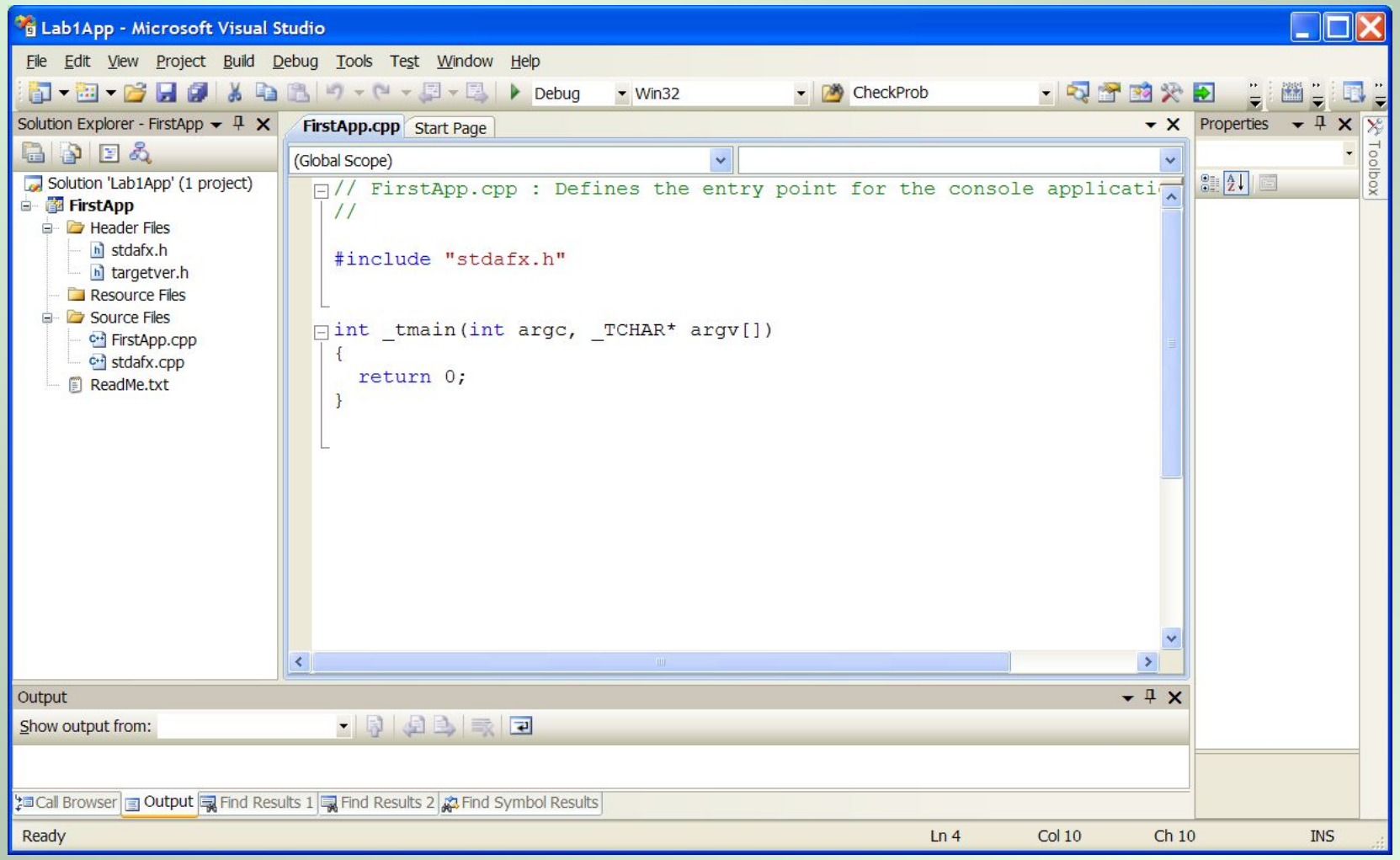
Solution (sln) – решение – группа проектов, объединенных под одной крышей. Например, все проекты, выполненные в рамках одной лабораторной работы.

Project (vcproj) – отдельный проект, исходное состояние которого генерируется при помощи мастера Application Wizard. Проект содержит информацию о версии языка, используемой платформе, конфигурации и других особенностей, заказываемых пользователем при создании проекта

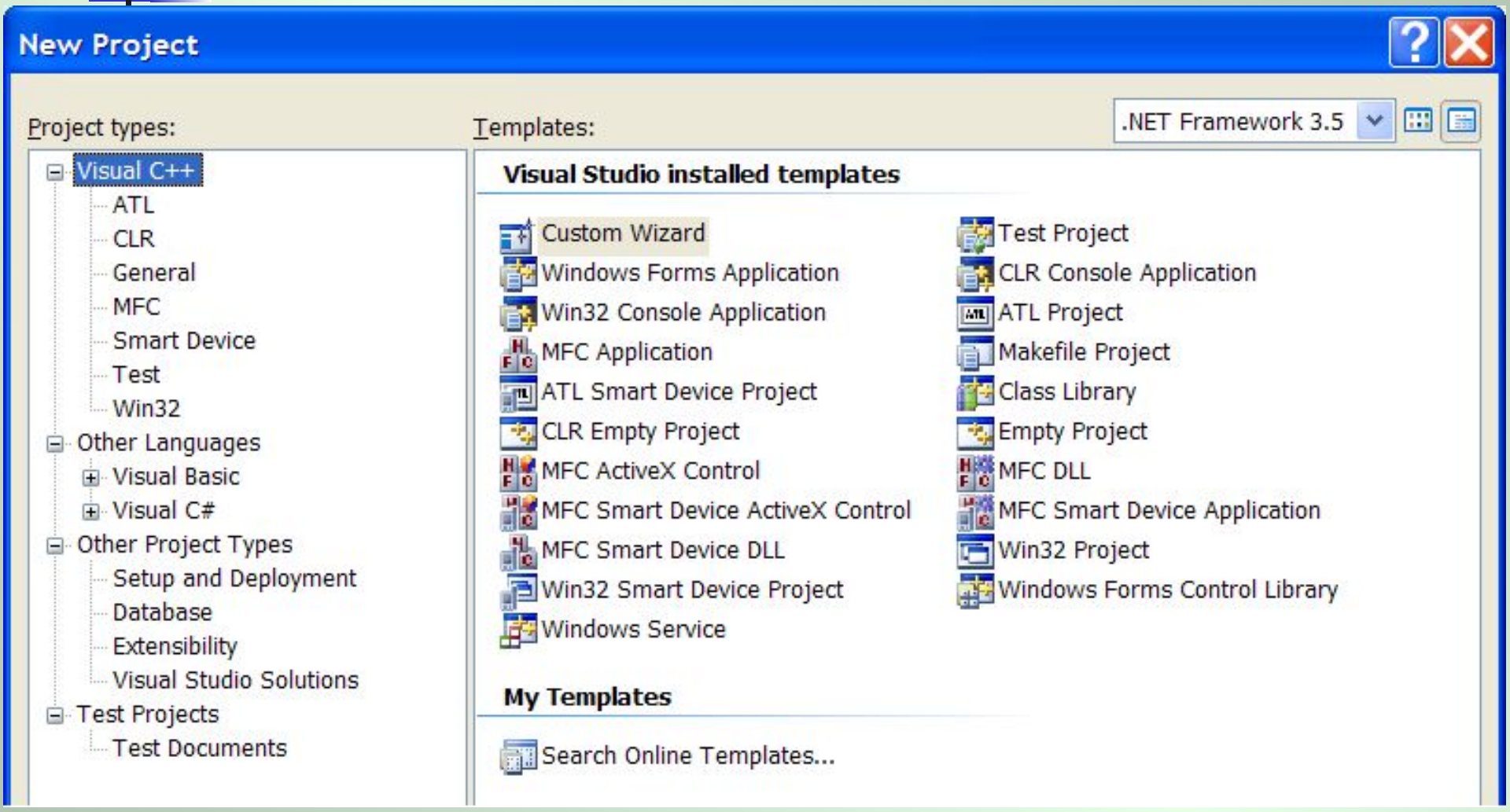
Начало 3


Win32 Application Wizard - FirstApp
Welcome to the Win32 Application Wizard
Overview
Application Settings
These are the current project settings:
Console application
Click Finish from any window to accept the current settings.
After you create the project, see the project's readme.txt file for information about the project features and files that are generated.
Win32 Application Wizard - FirstApp
Application Settings
Overview
Application Settings
Application type:
Windows application
Console application
DLL
Static library
Additional options:
Empty project
Export symbols
Precompiled header
Add common header files for:
ATL
MFC
< Previous
Next >
Finish
Cancel

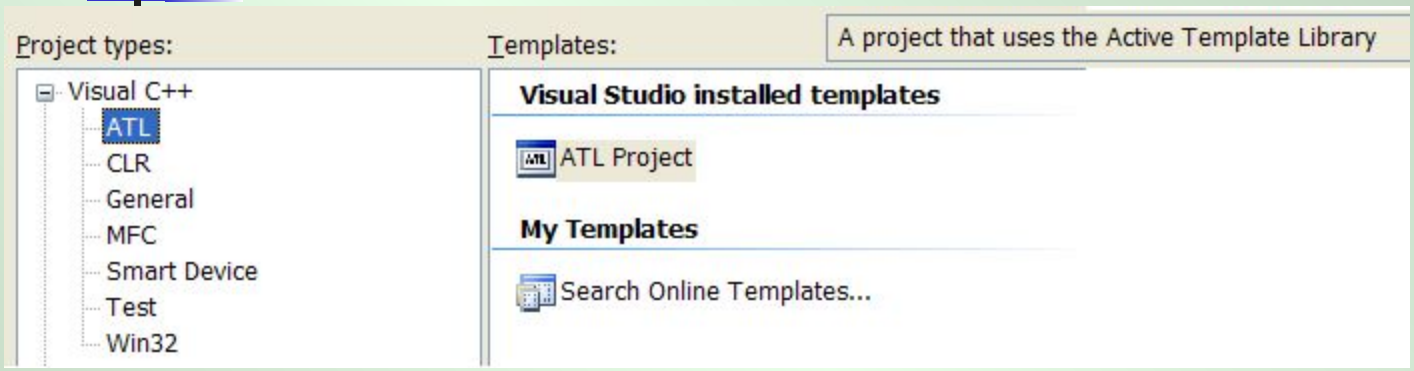
Начало 4

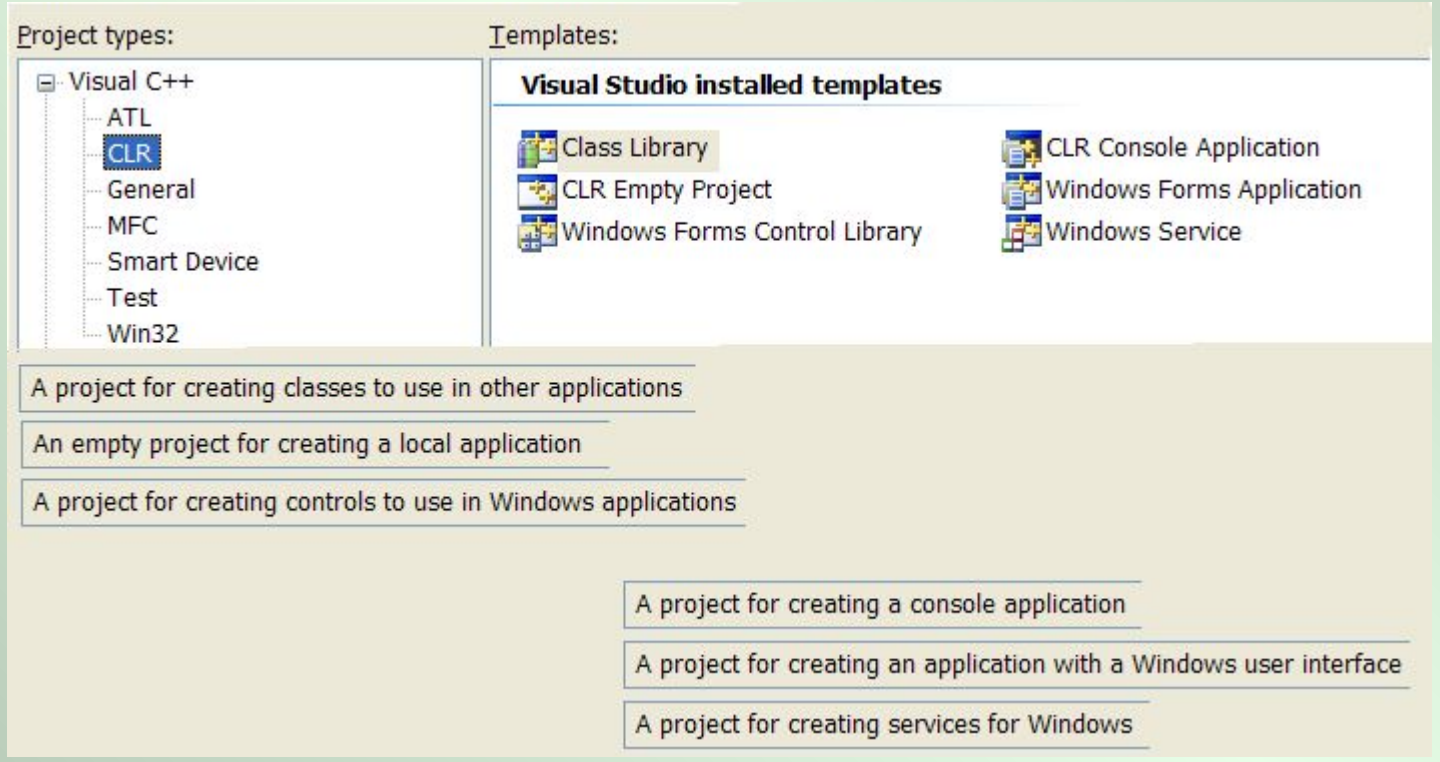

Lab1App - Microsoft Visual Studio
File Edit View Project Build Debug Tools Test Window Help
Debug
Win32
CheckProb
Solution Explorer - FirstApp
Solution 'Lab1App' (1 project)
FirstApp
Header Files
stdafx.h
targetver.h
Resource Files
Source Files
FirstApp.cpp
stdafx.cpp
ReadMe.txt
FirstApp.cpp
Start Page
(Global Scope)
// FirstApp.cpp : Defines the entry point for the console applicati
//
#include "stdafx.h"
int _tmain(int argc, _TCHAR* argv[])
{
return 0;
}
Properties
Toolbox
Output
Show output from:
Call Browser
Output
Find Results 1
Find Results 2
Find Symbol Results
Ready
Ln 4
Col 10
Ch 10
INS

Варианты проектов

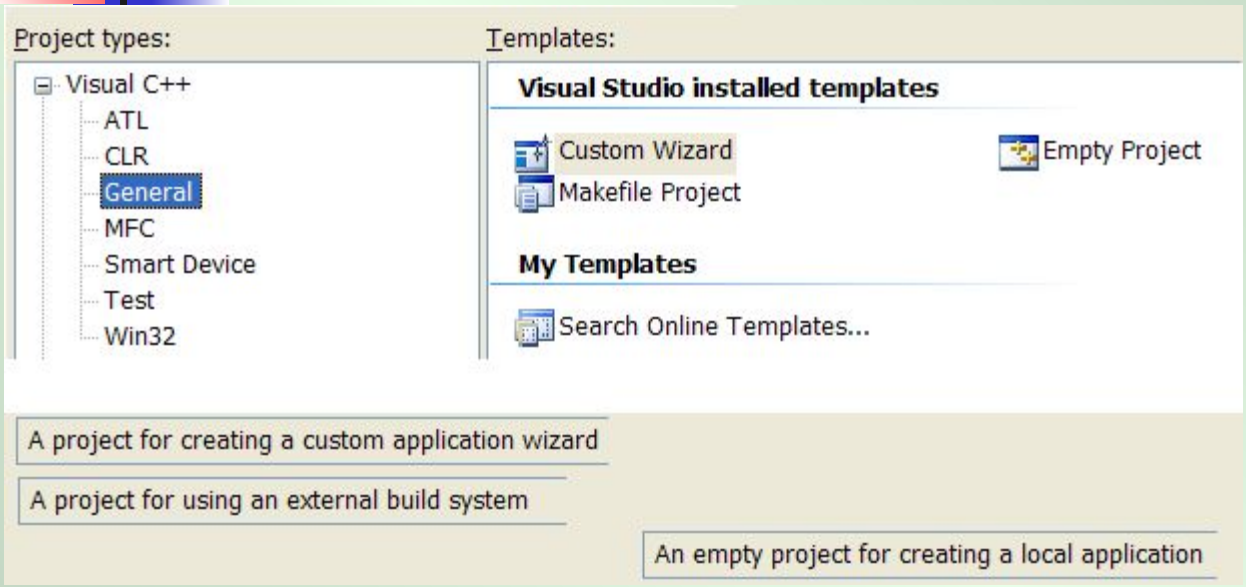


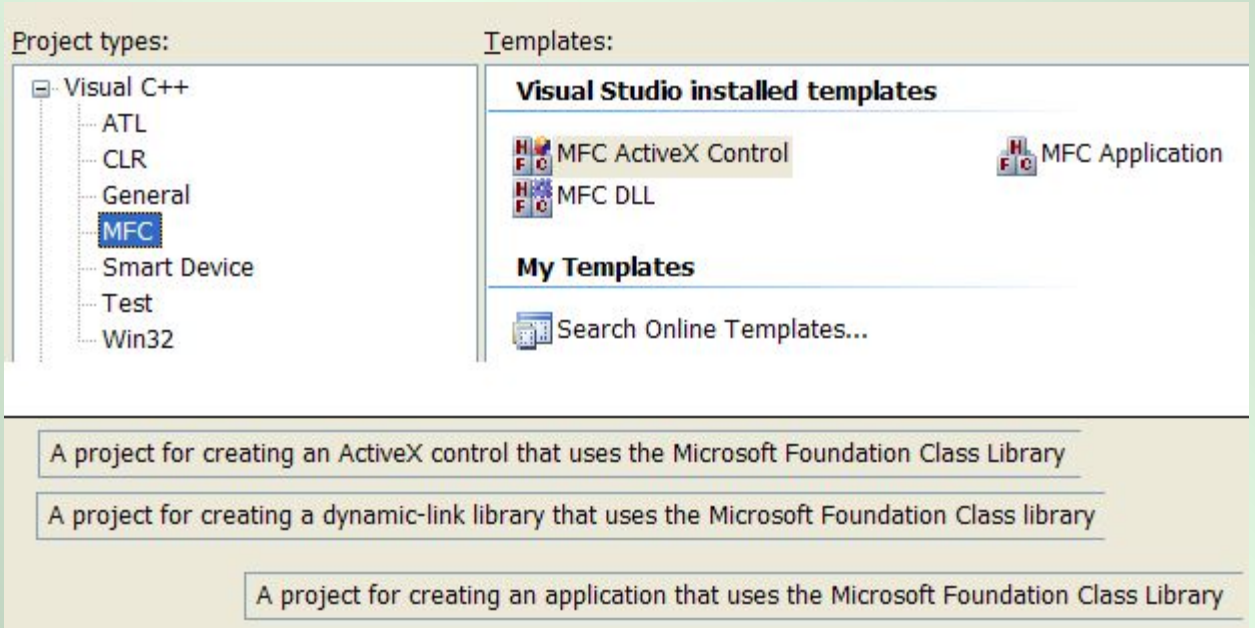
Проекты ATL и CLR



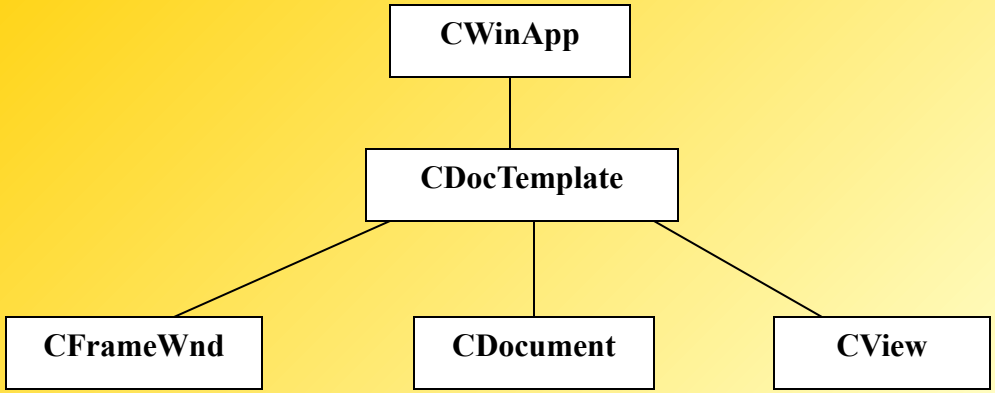


Проекты General и MFC





Архитектура Документ - Вид



CWinApp – класс инициализации приложения. Основная задача – разместить все, что нужно, в памяти и организовать цикл обработки сообщений.

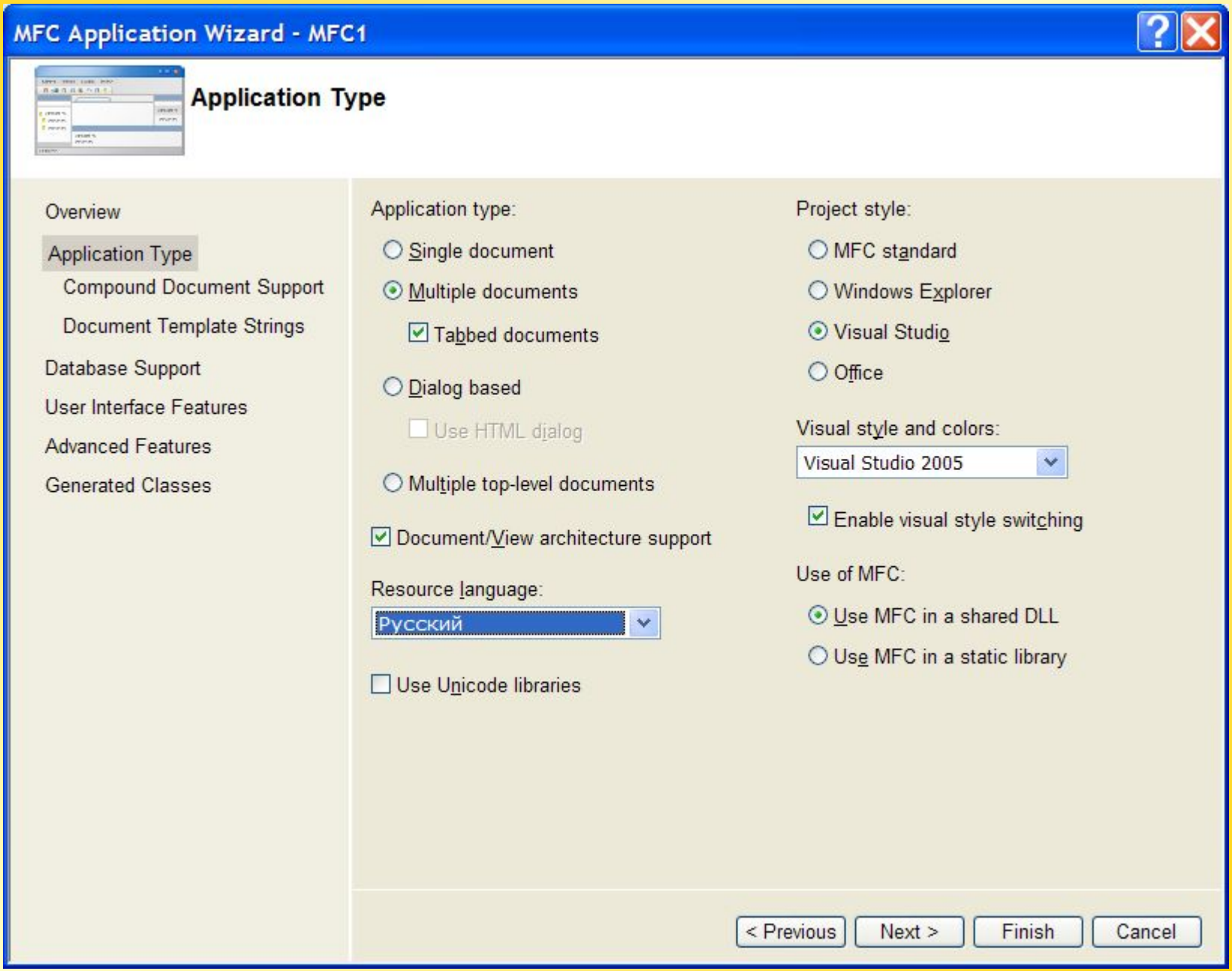
CDocTemplate – стандартный шаблон, объединяющий основные компоненты приложения, каждый из которых предназначен для обработки своей группы сообщений.

CFrameWnd – класс, определяющий основное окно приложения.

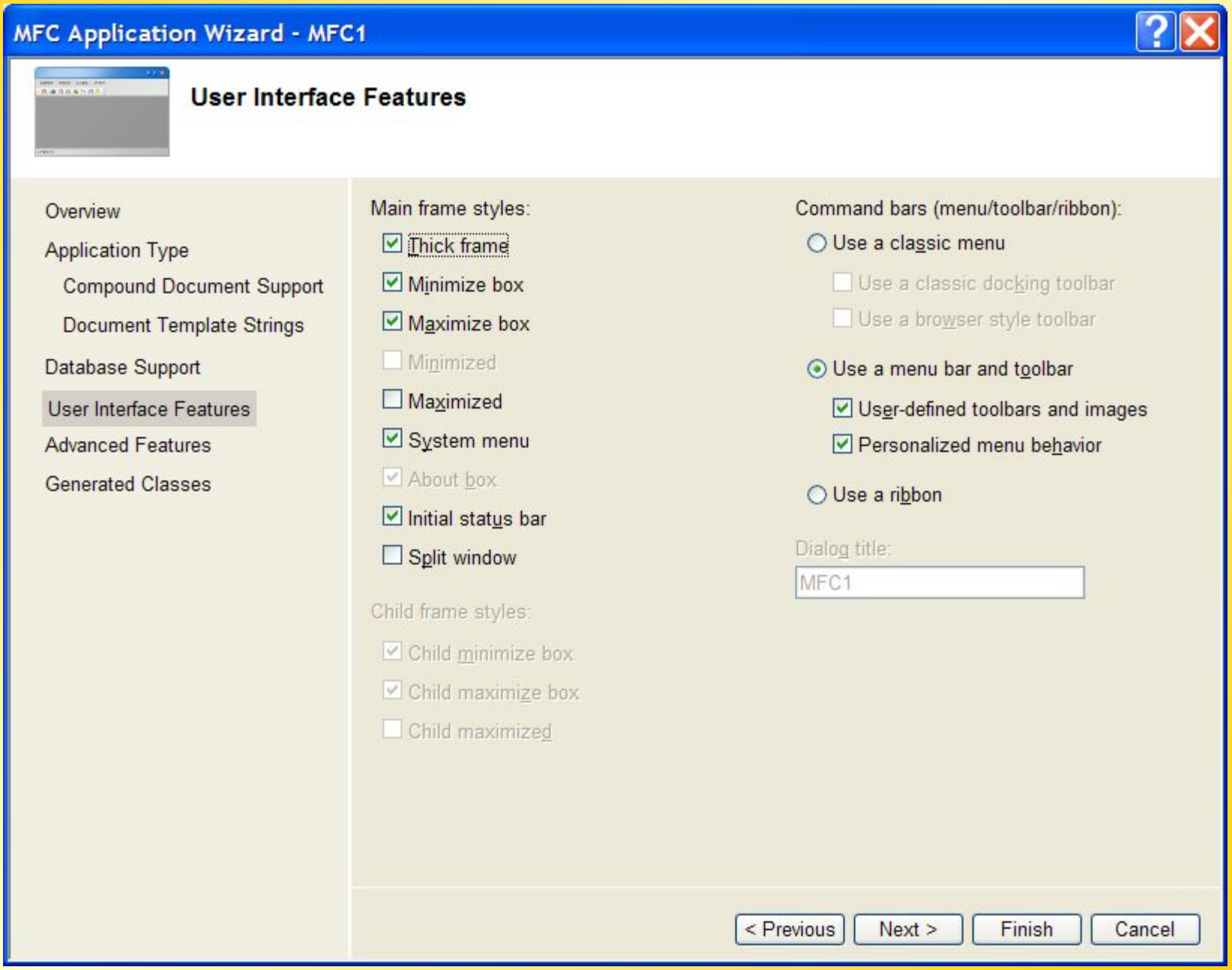
CDocument – класс, представляющий в себе всю совокупность данных, для переработки которых предназначено приложение.

CView – класс, предназначенный для проведения демонстрации пользовательских данных приложения

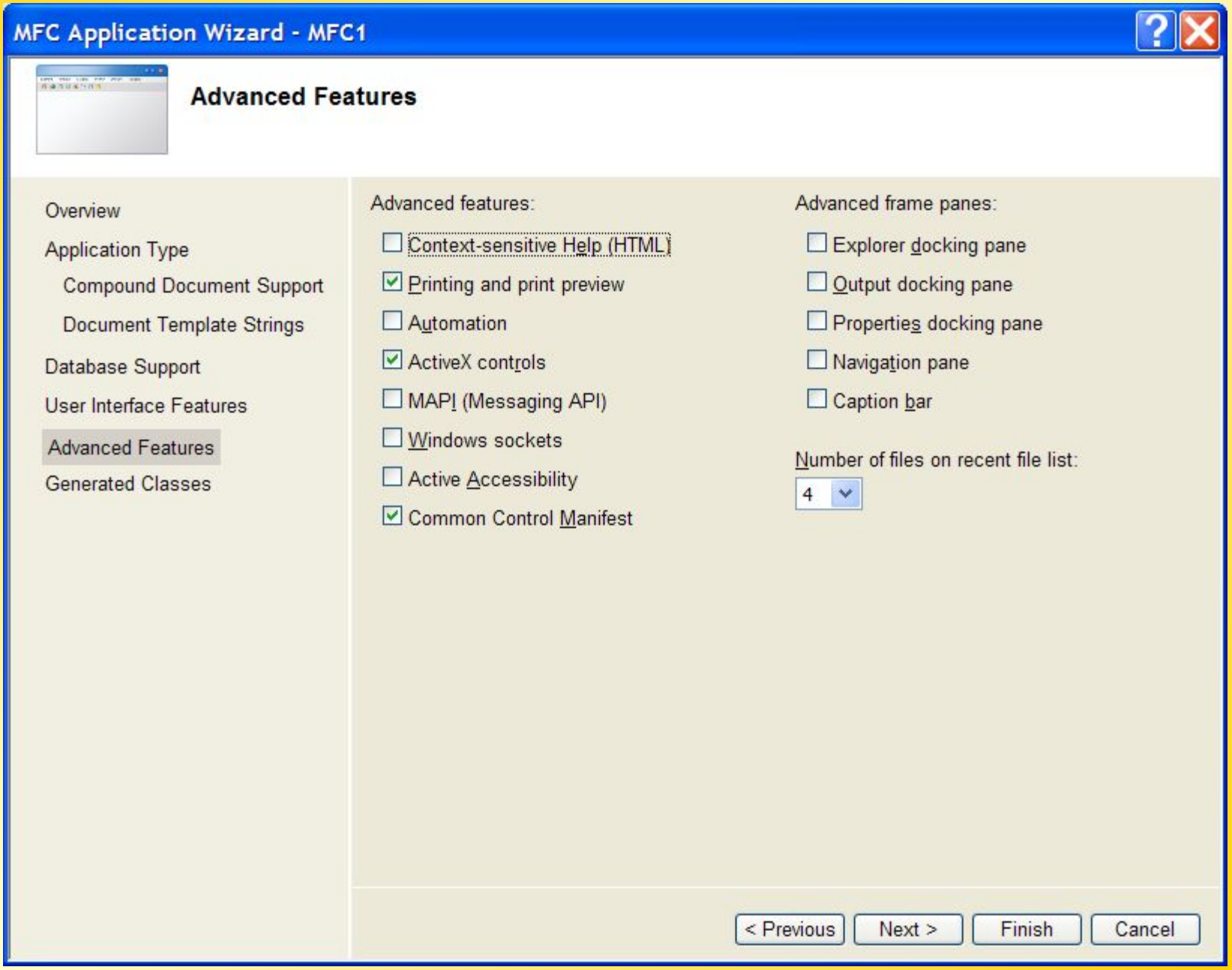
Создание приложения MFC Application



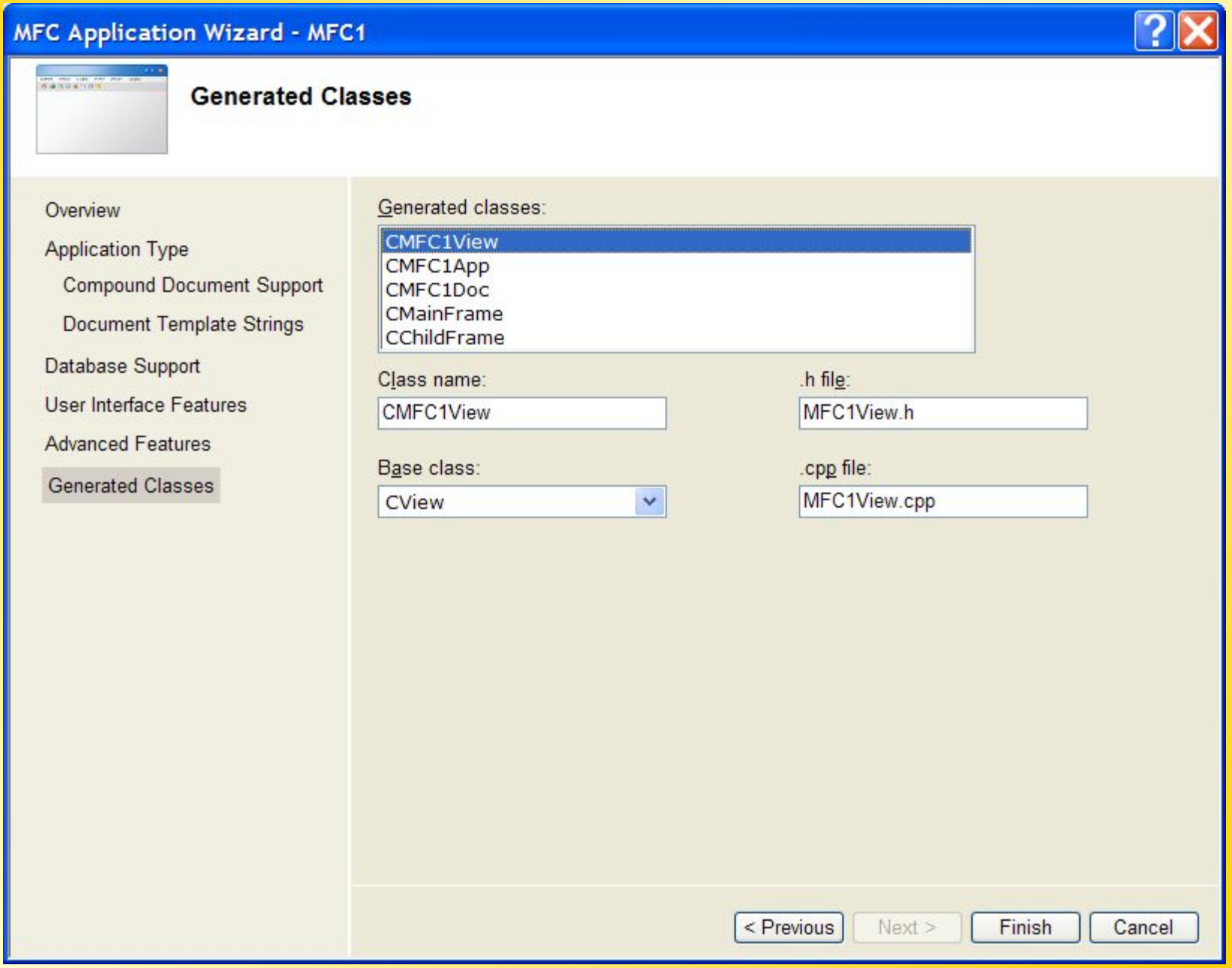
Создание приложения MFC Application



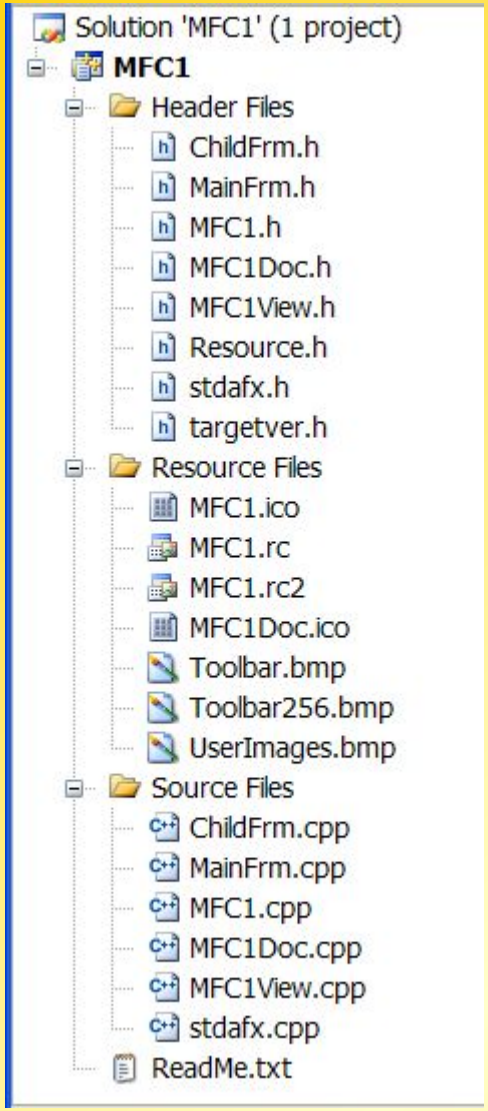
Создание приложения MFC Application



Создание приложения MFC Application



Создание приложения MFC Application



БИБЛИОТЕКА MICROSOFT FOUNDATION CLASS: обзор проекта MFC1

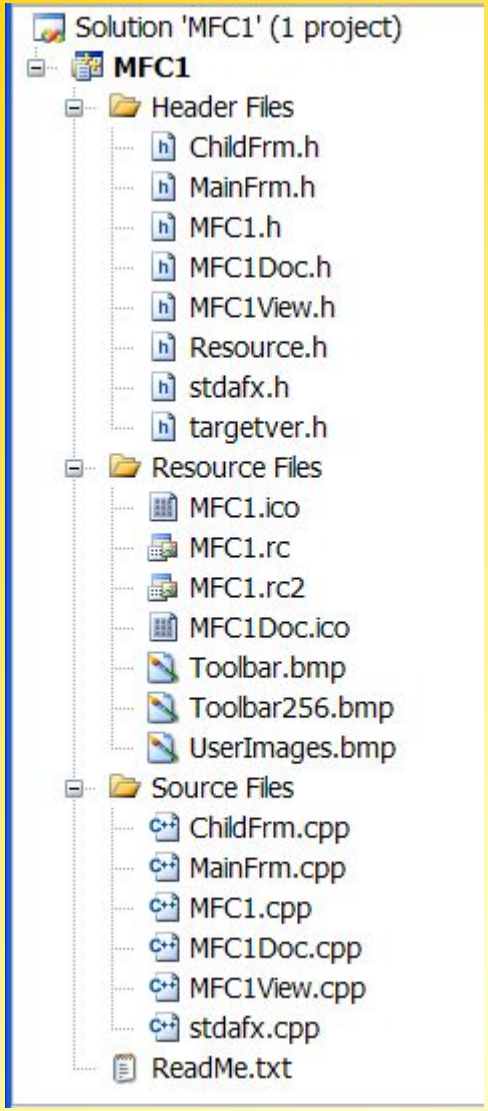
==========

Данное приложение MFC1 создано мастером приложений. Это приложение показывает основы использования фундаментальных классов Microsoft, а также является отправной точкой для создания конкретного приложения. В этом файле содержится краткое описание содержимого файлов, составляющих конкретное приложение MFC1.

MFC1.vcproj

Это основной файл проекта для проектов VC++, создаваемых с помощью мастера приложений. В нем содержатся сведения о версии Visual C++, создавшей файл, а также сведения о платформах, конфигурациях и свойствах проекта, выбранных с помощью мастера приложений.

Создание приложения MFC Application



MFC1.h

Это основной файл заголовка для приложения. В него включены другие определенные для проекта заголовки (в том числе Resource.h) и объявлен класс приложения CMFC1App.

MFC1.cpp

Это основной исходный файл приложения, содержащий класс приложения CMFC1App.

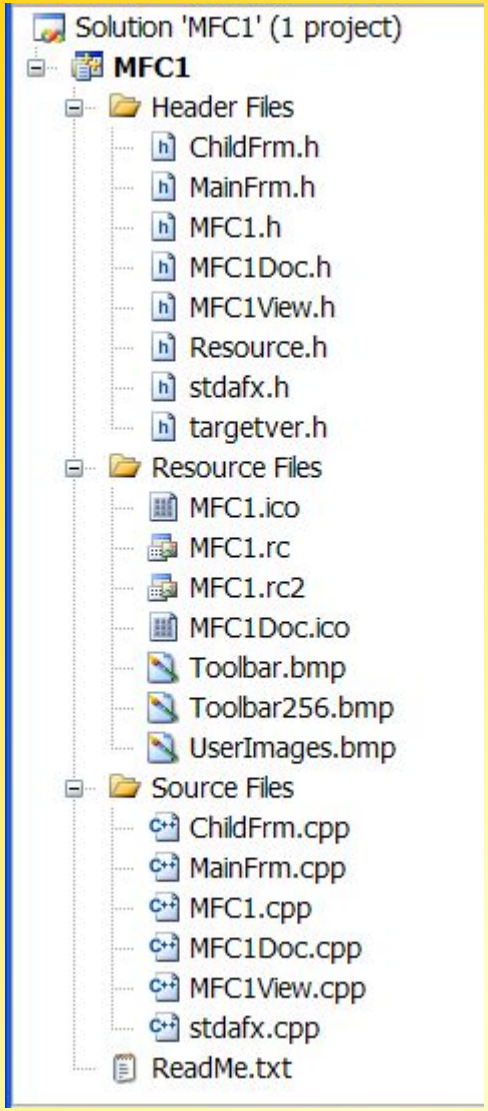
MFC1.rc

Это список всех ресурсов Microsoft Windows, используемых программой. В него включены значки, рисунки и курсоры, хранящиеся в подкаталоге RES. Этот файл можно редактировать непосредственно в Microsoft Visual C++. Ресурсы конкретного проекта находятся в 1049.

res\MFC1.ico

Это файл значка, используемого в качестве значка приложения. Этот значок включен посредством основного файла ресурсов MFC1.rc.

Создание приложения MFC Application



res\MFC1.rc2

Этот файл содержит ресурсы, не редактируемые в Microsoft Visual C++. Все ресурсы, не редактируемые редактором ресурсов, следует поместить в этот файл.

//

Для окна основной рамки:

В проект включается стандартный интерфейс MFC.

MainFrm.h, MainFrm.cpp

Эти файлы содержат класс рамок CMainFrame, получен-ный из CMDIFrameWnd и управляющий всеми свойствами рамок MDI.

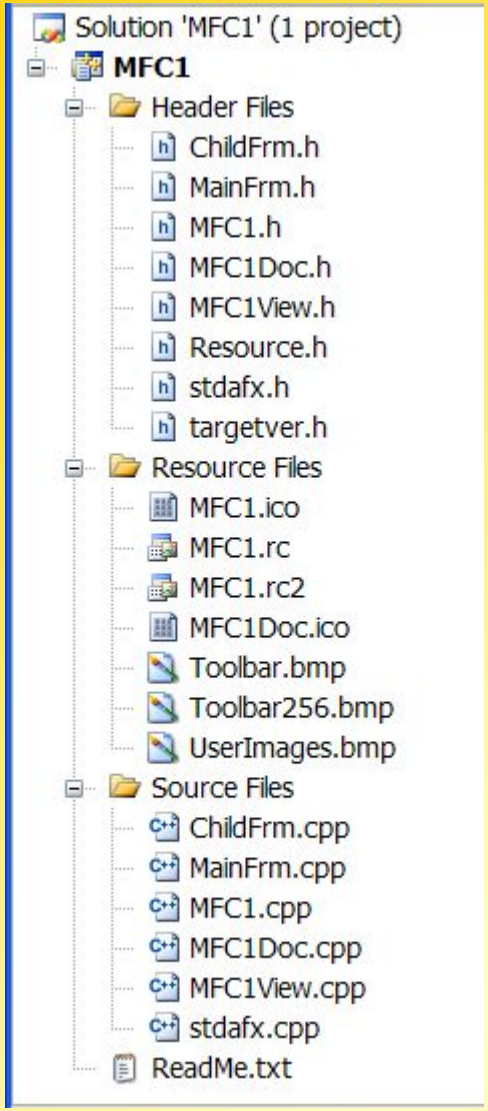
//

Для окна дочерней рамки:

ChildFrm.h, ChildFrm.cpp

Эти файлы определяют и реализуют класс CChildFrame, поддерживающий дочерние окна в MDI-приложении.

Создание приложения MFC Application



//

Мастер приложений создает один тип документов и одно представление:

MFC1Doc.h, MFC1Doc.cpp — документ

Эти файлы содержат класс CMFC1Doc. Измените эти файлы, чтобы добавить особые данные документа и реализовать сохранение и загрузку файлов (с помощью CMFC1Doc::Serialize).

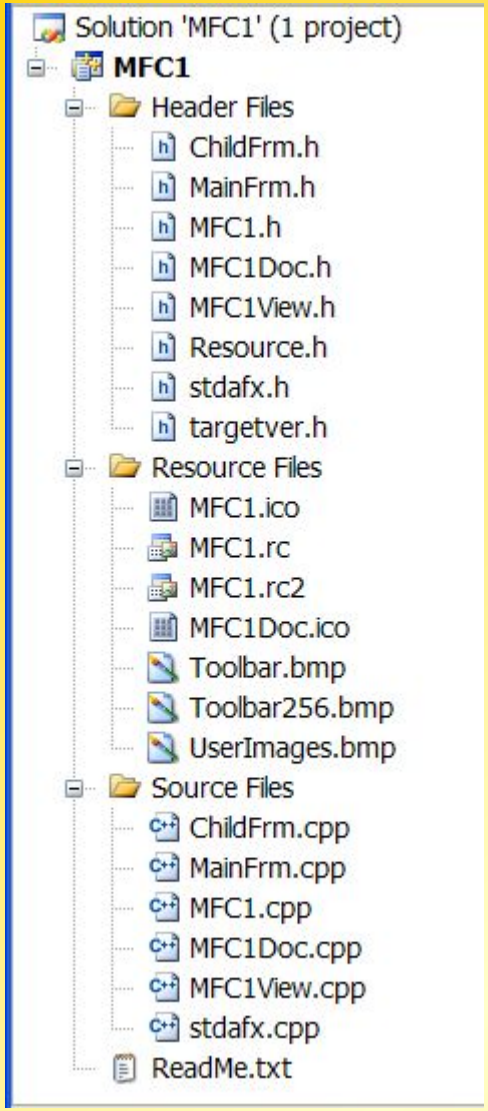
MFC1View.h, MFC1View.cpp — представление документа Эти файлы содержат класс CMFC1View.

Объекты CMFC1View используются для представления объектов CMFC1Doc.

res\MFC1Doc.ico

Это файл значка, который используется в качестве значка для дочерних окон MDI для класса CMFC1Doc. Этот значок включен с помощью главного файла ресурсов MFC1.rc.

Создание приложения MFC Application



//

Другие возможности:

Элементы ActiveX

Приложение включает поддержку использования элементов ActiveX.

Поддержка функций печати и предварительного просмотра

Мастер приложений создает код для обработки команд печати, настройки печати и предварительного просмотра с помощью вызова функций-членов класса CView из библиотеки MFC.

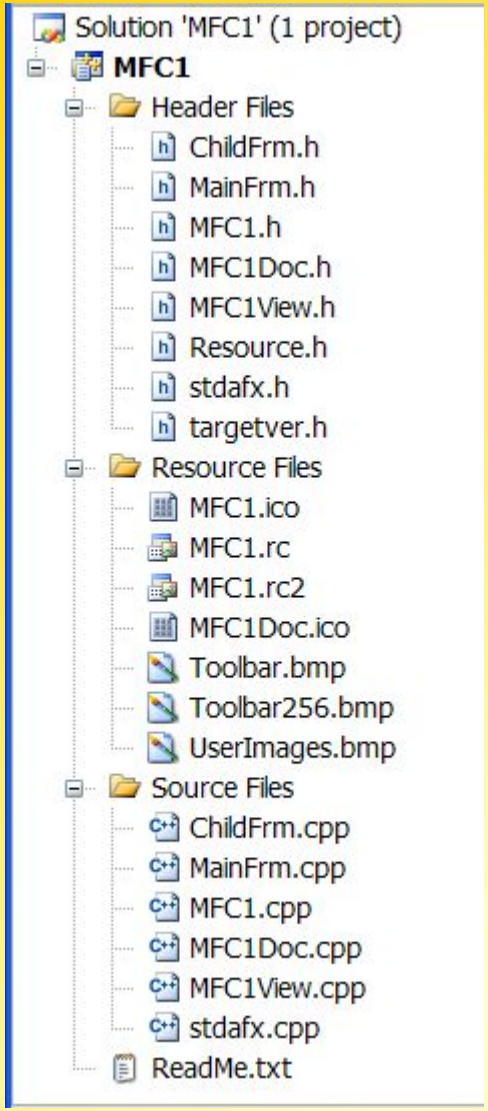
//

Прочие стандартные файлы:

StdAfx.h, StdAfx.cpp

Эти файлы используются для построения файла предкомпилированного заголовка (PCH) с именем MFC1.pch и файла предкомпилированных типов с именем StdAfx.obj.

Создание приложения MFC Application



Resource.h

Это стандартный файл заголовка, определяющий новые идентификаторы ресурсов. Microsoft Visual C++ прочитывает и обновляет этот файл.

MFC1.manifest

Файлы манифестов приложений используются Windows XP для описания зависимости приложений от определенных версий параллельных сборок. Загрузчик использует эти сведения для загрузки надлежащей сборки из кэша сборок или закрытого типа сборки из приложения. Манифест приложения может быть включен для распространения в качестве внешнего файла .manifest, устанавливаемого в ту же папку, что и исполняемый файл приложения, или он может быть включен в исполняемый файл в виде ресурса.