Спецификация COM -для разработки слабосвязанных компонентов (весь офис и половина WINDOWS сделаны с помощью этой спецификации).

Цель Microsoft при разработке: разработать стандарт, описывающий как создать универсальный компоненты, которые бы стыковались между собой.

Пример COM-компонента: проводник, раньше любой браузер

В основе COM лежит модель ПО

Прижилась в полном виде только в windows

Разработка программных объектов – это объекты времени выполнения (до этого времени компиляции работали), т.е. это позднее связывание -> создается на этапе выполнения

Все объекты имеют свой идентификатор (128 битный код, он не повторяется)

Объекты COM могут быть размещены в контейнере (может использоваться dll-inproc объект, exe)

Тип и экземпляр объекта

Dll – загружается в единственном экземпляре в оперативную память. Если сделаем еще раз загрузку библиотеке, то происходит проецирование

У каждого объекта свой экземпляр объекта

COM-server – однокомпонентный или многокомпонентном

Вся поддержка com осуществляется с помощью ole32.dll

Чтобы работать с com – include <objbase>

Можно работать локально (localhost), можно не локально

COM-server, COM-client – обращается к серверу (вызывает функцию из библиотеки ole32) и создаёт объект, с которым начинает работать. После создания – может вызывать методы com-объекта.

COM объект работает через посредника, через ole32, которая включена в Windows API.

Все функции из ole32 делают возврат кода, который описан в структуре HResult.

Первых 2 бита – серьезность ошибки (00 – хорошо, 11 - плохо), резервные биты, если сами генерируем дальше ставим 1, если не мы – то 0, facility – подсистема

Необходимы интерфейсы com-объекта (соответствует типу и набор интерфейсу)

COM-объект может иметь 1 или несколько интерфейсов, каждый com-интерфейс характеризуется идентификатором.

Несколько стандартных интерфейсов: IUnknow (идентификатор заранее известен)

Три метода об соглашениях об вызовах: параметры в обратном порядке, отчищает в вызываемом коде.

QueryInterface, Release (уменьшает счётчик на единицу, если нам не нужен интерфейс, если 0 – никто не использует), AddRef (повышает счётчик обращения к интерфейсу)

Возвращают FResult.

Каждый параметр QueryInterface передаем идентификатор, а получаем адрес -> для получения указателя.