**Модель COM (Common Object Model)**

COM – модель спецификации для разработки слабосвязанных компонентных приложений.

Задача COM – разработка стандарта программного кода, который представляет собой компоненту.

Пример COM-компонента – когда можно чёт перетащить из Word в Excel, и наоборот.

Браузер – тоже COM-компонент.

В основе COM-спецификации лежит модель ПО

Каждый COM-компонент имеет свой айди, типа GUID(Globally usinque identifier), длиной 128 бит).

Пример GUID:

#pragma once

#include <objbase.h>

GUID CLSID CA = {0x3f5d8796, …}

COM-объекты размещаются в контейнерах DLL и EXE (DLL – это объект, который работает в процессе, работающем в процессе другого приложения).

DLL загружается в единственном экземпляре (свойство DLL)

COM-объект + COM-компонент = сервер.

COM-сервер: однокомпонентный (один тип объекта), двухкомпонентный, …, многокомпонентный.

Идентификатор соответствует типу COM-объекта.

Вся поддержка COM-технологии осуществляется специальной библиотекой #include <objbase.h>

Типы COM-серверов:

1. CLSCTX\_INPROC\_SERVER (DLL внутрипроцессорный сервер);
2. CLSCTX\_LOCAL\_SERVER (EXE-сервер за границами процесса, но на том же компьютере);
3. CLSCTX\_REMOTE\_SERVER (EXE-сервер на удаленном компьютере).

COM-клиент – программа, которая обращается к COM-серверу, вызывает функцию из библиотеки ole32 и создает COM-объект.

COM-клиент -> OLE32.DLL/OLE32.LIB -> COM-сервер -> возврат управления в OLE32 -> возврат управления в COM-клиент.

Все функции, за небольшим исключением, в OLE32.dll возвращают код, который описан структурой HRESULT (4 байта = 32 бита, первые 2 бита – код ошибки).

Для того, чтобы работать с экземплярами, нам нужны интерфейсы. Для работы с объектами – должны знать SID и набор интерфейсов (вызов функций этого объекта).

Один COM-объект может иметь один и более интерфейс.

Каждый интерфейс идентифицируется собственным ID(GUID).

Стандартный интерфейс IUnknown (объявлен в <Unknwnn.h>, известен его ID и структура).

В IUnknown 3 метода:

1. QueryInterface;
2. AddRef;
3. Release.

Для того, чтобы COM-клиент мог работать с COM-сервером, он должен знать CLSSID, тип сервера, идентификаторы интерфейса (методы).

CoCreateInstance создает COM-объект

Параметры CoCreateInstance:

1. GUID;
2. NULL;
3. Тип сервера;
4. Какой хочу интерфейс;
5. Указатель на запрашиваемый интерфейс.