

1. Что такое Component Object Model (далее - COM)?

Модель для проектирования и создания компонентных объектов.

Модель определяет множество технических приемов, которые могут быть использованы разработчиком при создании независимых от языка программных модулей, в которых соблюдается определенный двоичный стандарт.

Это независимая от платформы, распределенная, объектно-ориентированная система для создания бинарных программных компонентов, которые могут взаимодействовать между собой

2. Что такое COM-компонент?

программный модуль,

это некоторая сущность, имеющая состояние и методы доступа, позволяющие изменять это состояние

COM-объекты можно создавать прямым вызовом специальных функций, но напрямую уничтожить его невозможно. Вместо прямого уничтожения используется механизм самоуничтожения, основанный на подсчете ссылок

3. Что такое COM-интерфейс?

CoClass – это класс, поддерживающий набор методов и свойств (один или более), с помощью которых можно взаимодействовать с объектами этого класса. Такой набор методов и свойств называется COM-интерфейсом

COM интерфейс – это контракт, состоящий из списка связанных прототипов функций, чье назначение определено, а реализация – нет!

классам, имеющим только виртуальные методы, описания без реализации

4. Что такое COM-контейнер? Какие бывают?

Для размещения компонентов в Windows могут быть применены два вида контейнеров: DLL-файл и EXE-файл

5. Что такое COM-сервер и COM-клиент? Какие бывают сервера?

Приложения, использующие COM-компоненты (вызывающие функции интерфейсов, реализованных COM-компонентами), называют COM-клиентами

а контейнеры с расположенными в них компонентами – COM-серверами

В зависимости от типа контейнера и места его расположения (локальное или удаленное) различают несколько типов серверов: INPROC (DLL, локальный), LOCAL (EXE, локальный), REMOTE (EXE, удаленный)

COM-серверы в зависимости от количества реализуемых ими компонентов подразделяются на «однокомпонентные» и «многокомпонентные»

6. Что такое GUID? Для чего он используется?

CLSID является глобально уникальным идентификатором (GUID). GUID имеет размер 128 бит и уникален в пространстве и времени.

С помощью CLSID можно точно указать, какой именно объект требуется. Тип данных GUID применяется и для идентификации COM-интерфейсов. В этом случае он называется IID

7. Какие бывают интерфейсы?

Интерфейсы бывают двух типов: стандартные и произвольные

За стандартными интерфейсами закреплены predetermined GUID-идентификаторы. Важнейшим среди стандартных интерфейсов является интерфейс IUnknown

Все остальные интерфейсы являются производными (наследуют все методы) от IUnknown. Каждый компонент должен поддерживать (часто говорят «реализовывать») как минимум стандартный интерфейс IUnknown

8. Поясните функции интерфейса IUnknown.

AddRef – увеличивает счётчик ссылок на интерфейс на 1

QueryInterface – получение указателя на интерфейс по IID

Release - уменьшает счётчик ссылок на интерфейс на 1

9. Объясните «счётчик ссылок на интерфейс». Когда увеличивается/уменьшается?

Данный «счётчик ссылок на интерфейс» необходим для отслеживания момента, когда экземпляр COM-компонента больше не требуется и может быть удалён

10. Поясните функции интерфейса IClassFactory.

CreateInstance – метод предназначенный для создания экземпляра COM-компонента

LockServer – увеличение счётчика блокировки COM-сервера

11. Объясните «счётчик ссылок на компонент». Когда увеличивается/уменьшается?

Увеличение этого счётчика происходит в конструкторе COM-компонента (он вызывается методом CreateInstance), а уменьшается в деструкторе

Фабрика классов также помогает удобно следить за жизненным циклом COM-компонент

Это является важной частью работы COM-сервера, так как при попытке освободить его ресурсы, вывод о том можно это сделать или нет, основывается на том факте используются ли хоть какие-то его COM-компоненты или нет

12. Какие соглашения о вызове и возврате должны соблюдаться для COM-функций?

Все методы COM-интерфейса должны поддерживать соглашение о вызовах stdcall, а также возвращать HRESULT за исключением AddRef и Release – они возвращают текущее значение счётчика ссылок на интерфейс

13. Опишите структуру HRESULT.

```
// Values are 32 bit values laid out as follows:
//
//  3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1
//  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
//  +---+---+-----+-----+-----+-----+
//  |Sev|C|R|   Facility   |           Code           |
//  +---+---+-----+-----+-----+-----+
//
```

30-31 биты HRESULT отображают успешность выполнения функции COM-компонента

29 бит HRESULT отображает кем был определен данный статус код: пользователем или системой

28 бит HRESULT является зарезервированным

16-27 биты HRESULT отображают к какой технологии относится статус код

0-15 биты HRESULT отображают точный результат в рамках заданной технологии и серьезности

14. Что должен знать COM-клиент чтобы взаимодействовать с COM-сервером?

При этом при работе с COM-компонентом клиент должен «знать» только GUID-идентификатор этого компонента (CLSID), GUID идентификаторы (IID), тип сервера и структуры (сигнатуры соответствующих методов) произвольных интерфейсов компонента, которые он предполагает применять

15. Что такое regsvr32? Принцип работы?

Реестр Windows – это централизованная иерархическая база данных, хранящая информацию о параметрах конфигурации операционной системы и установленных приложений

по сути данная утилита просто вызывает некоторые экспортируемые функции из DLL

Что это и зачем?

Для реализации свойства «Независимости от местоположения» в Windows каждый COM-компонент должен быть зарегистрирован в Windows-реестре. Для регистрации компонента и применяется специальная утилита regsvr32

```
regsvr32.exe ./ComSampleServer.dll
```

16. Где найти информацию о компоненте в реестре?

```
regedit
```

17. Перечислите и объясните 5 COM-функций экспортируемых из DLL.

CoInitializeEx - инициализации библиотеки OLE32

CoCreateInstance - Для создания экземпляра компонента необходимо вызвать функцию

QueryInterface - Для получения указателя на другие интерфейсы

CoUninitialize - Работа клиента должна завершаться освобождением библиотеки OLE32

DllGetClassObject PRIVATE – Эта функция используется для предоставления экземпляра класса, реализующего COM-интерфейс.

DllCanUnloadNow PRIVATE – Эта функция проверяет, может ли DLL быть выгружена из памяти.

DllRegisterServer PRIVATE – Эта функция используется для регистрации COM-компонента в реестре Windows.

DllUnregisterServer PRIVATE – Эта функция используется для отмены регистрации COM-компонента в реестре.

DllMain PRIVATE – Это основная точка входа для DLL. Она вызывается при загрузке и выгрузке DLL

18. Какая системная библиотека отвечает за COM-инфраструктуру?

OLE32