**Коллоквиум 1, СП-ПОИТ-3-2023**

***Вопросы не пишем, пишем номер и ответ***.

1. В правом верхнем углу листа:

***1-СП-ПОИТ-3/группа, Фамилия И.О.***

1. Расшифруйте аббревиатуру DLL.

Dynamic Link Library

1. Поясните понятия «раннее связывание» и «позднее связывание».

***Раннее связывание*** (early binding): связывание данных с кодом происходит на этапе компиляции программы.

***Позднее связывание*** (late binding): связывание данных с кодом происходит в процессе выполнения программы, а не на этапе компиляции.

Ранее связывание - связывание переменных и функций происходит на этапе компиляции программы, до момента её выполнения. Позднее связывание - подключение программного модуля во время исполнения программы

1. Как называется функция, которая является точкой входа DLL-библиотеки и в каких случаях эта функция вызывается.

DllMain

Вызывается в следующих случаях:

1) Когда процесс загружает DLL с помощью функции LoadLibrary.

2) Когда процесс выгружает DLL с помощью функции FreeLibrary.

3) Когда создается поток в процессе, в который была загружена DLL.

4) Когда завершается поток в процессе, в который была загружена DLL.

1. Какая программа создает DLL?

DLL создается с использованием компилятора, который поддерживает создание динамических библиотек

--или--

Совместно компилятором и компоновщиком

1. Для чего применяется директива *EXTERN “C”.*

extern "C" указывает, что функция определена в другом месте и указывает компилятору, что следующий блок кода должен быть компилирован с использованием соглашений о вызовах, принятых в языке C.

[Когда вы объявляете функцию с extern "C", компилятор обещает не изменять ее имя](https://ru.stackoverflow.com/questions/169022/%D0%98%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5-extern-c-%D0%B2-%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D0%B5-%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B0). [Это позволяет функциям, написанным на C++, вызываться из кода на языке C и других языках, которые не поддерживают декорирование имен](https://ru.stackoverflow.com/questions/169022/%D0%98%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5-extern-c-%D0%B2-%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D0%B5-%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B0)

1. Для чего экспортируются функции DLL?

Экспорт функций из DLL (Dynamic Link Library) позволяет другим программам или модулям использовать эти функции. Экспортированные функции предоставляют интерфейс для взаимодействия с кодом, находящимся внутри DLL. [Это позволяет нескольким программам совместно использовать функции, запрограммированные в одном файле, и даже делать это одновременно](https://windows-school.ru/blog/dll_fajly_chto_takoe/2019-05-10-387)

1. Назовите 2 способа экспорта функций DLL.

[1. Использование файла определения модуля (DEF)](https://learn.microsoft.com/ru-ru/cpp/build/exporting-from-a-dll?view=msvc-170) (Создание файла .def (definition) для указания функций, которые нужно экспортировать.)

LIBRARY MyLibrary

EXPORTS

MyExportedFunction

[2. Использование ключевого слова \_\_declspec (dllexport) в определении функции](https://learn.microsoft.com/ru-ru/cpp/build/exporting-from-a-dll?view=msvc-170)

\_\_declspec(dllexport) void MyFunction1();

1. Как называется функция загрузки DLL? Какие параметры принимает эта функция?

Для этого используется функция LoadLibrary, Она принимает один параметр - строку, содержащую имя DLL, которую нужно загрузить и возвращает ее дескриптор.

1. Как называется функция выгрузки DLL? Какие параметры принимает эта функция?

Для этого используется функция FreeLibrary, которая получает дескриптор библиотеки, возвращенный из функции LoadLibrary

1. Для чего применяется функция GetProcAddress? Какие параметры принимает эта функция? Что возвращает эта функция?

Функция GetProcAddress для получения адреса функции, экспортированной из DLL.

Параметры функции GetProcAddress:

1. дескриптор модуля.
2. Имя функции, адрес которой нужно получить.

Возвращает указатель на функцию, если она была найдена, и NULL в противном случае.

1. Поясните выражение «DLL проецируется в адресное пространство процесса».

DLL загружается в адресное пространство процесса, ч[то позволяет процессу вызывать функции DLL](https://ru.stackoverflow.com/questions/752458/%d0%9a%d1%83%d0%b4%d0%b0-%d0%b7%d0%b0%d0%b3%d1%80%d1%83%d0%b6%d0%b0%d0%b5%d1%82%d1%81%d1%8f-dll-%d0%bf%d1%80%d0%b8-%d0%b5%d1%91-%d0%b8%d1%81%d0%bf%d0%be%d0%bb%d1%8c%d0%b7%d0%be%d0%b2%d0%b0%d0%bd%d0%b8%d0%b8-%d0%bf%d1%80%d0%be%d1%86%d0%b5%d1%81%d1%81%d0%be%d0%bc)

Когда программа обращается к функциям из DLL, он использует таблицу адресов функций, известную как таблица импорта.

1. Поясните понятие «библиотека импорта».

[Библиотека импорта – это библиотека, которая автоматизирует процесс подключения и использования динамической библиотеки](https://learn.microsoft.com/ru-ru/cpp/build/reference/using-an-import-library-and-export-file?view=msvc-170). В ней содержится не сам код библиотеки, а только ссылки на все функции, экспортируемые из файла DLL, в котором все и хранится.

1. Какая программа создает библиотеку импорта.

Создается на этапе **компановки** (линковки)

1. Расшифруйте аббревиатуру COM. Поясните смысл термина.

Component Object Model – это технология, разработанная Microsoft. Используется для создания компонентов ПО, которые могут взаимодействовать между собой на основе стандартизированных интерфейсов в различных приложениях, написанных на любом ЯП.

1. Поясните понятие «клиент-серверная архитектура приложения».

Клиент-серверная архитектура - это модель организации вычислительных систем, в которой задачи распределены между клиентами и серверами. В такой архитектуре клиент, обычно являющийся пользователем или программой, запрашивает услуги или ресурсы у сервера, который отвечает на запросы, предоставляя необходимые данные или функциональность.

Основное отличие процесса-клиента от процесса-сервера в том, что инициатором обмена данными всегда является процесс-клиент.

1. Для чего применяется OLE32.DLL?

OLE32.DLL — это динамическая библиотека, используемая в операционных системах Microsoft Windows для реализации технологии Object Linking and Embedding.

**При создании объекта** посредником между COM-клиентом и COM-сервером выступает библиотека **OLE32.DLL** (библиотека импорта **OLE32.LIB**).

1. Назовите 2 типа контейнера для COM-сервера.

DLL-файл и EXE-файл.

1. Перечислите три типа COM-серверов.

CLSCTX\_INPROC\_SERVER - dll-сервер внутрипроцессовый

CLSCTX\_LOCAL\_SERVER - exe-сервер, который работает за пределами процесса, но на той же машине

СLSCTX\_REMOTE\_SERVER - exe-сервер, который работает на удаленной машине

Внутрипроцессный сервер (in-process server) Представляет собой библиотеку DLL, проецируемую на адресное пространство клиентского процесса, то есть использующую его адресное пространство.

Внепроцессный сервер (out-process server) Представляет собой исполняемый файл EXE, выполняющийся в собственном адресном пространстве, однако на том же компьютере, что и клиент.

Удалённый сервер (remote server) Выполняется на удалённом компьютере и доступен благодаря DCOM.

1. Чем отличаются однокомпонентные сервера от многокомпонентных.

com-сервер называется "однокомпонентым" если состоит из одного пользовательского компонента помимо стандартного компонента, который реализует интерфейс IClassFactory

"многокомпонентный" сервер состоит из нескольких пользовательских компонентов

1. Расшифруйте аббревиатуру GUID, поясните смысл термина.

GUID (Globally Unique IDentifier -Глобально уникальный идентификатор)

GUID - тип данных размером 128 бит, который используется для идентификации com-компонета или com-интерфейса. Гарантируется, что GUID, сгенерированный на конкретном компьютере в определённое время, не имеет себе аналогов во всём мире.

1. Для чего используется макрос STDMETHODCALLTYPE?

[Макрос STDMETHODCALLTYPE](https://stackoverflow.com/questions/59735880/stdcall-winapi-vs-stdmethodcalltype-vs-apientry)[определяет соглашение о вызове функции, которое используется для вызова функций в COM](https://learn.microsoft.com/en-us/cpp/cpp/stdcall?view=msvc-170). Расширяется в \_\_stdcall.

1. Для чего используется структура HRESULT?

HRESULT является 32-битным значением для представления результатов операции, и часто используется в качестве возвращаемого значения в технологии COM (для методов интерфейсов COM).

**Все функции OLE32.DLL возвращают значение типа HRESULT.**

HRESULT тип данных, который хранит информацию о результате вызова функции компонента

размер 32 бит

*Старший бит (31-й)* указывает статус выполнения функции (0 - успешное (S\_OK); 1 – ошибка)

*Следующие 15 бит* хранят информацию об области, в которой произошла ошибка. (Каждая область имеет свой диапазон кодов ошибок)

*Последние 16 бит* - конкретный код ошибки в рамках указанной области.

1. Поясните термин «стандартный CОМ-интерфейс».

Стандартный COM-интерфейс – это интерфейс, который описан в спецификации COM. За стандартными интерфейсами закреплены предопределенные GUID-идентификаторы.

1. ***Перечислите*** стандартные COM-интерфейсы и их методы.

*Описание не надо*[***IUnknown***](https://learn.microsoft.com/ru-ru/windows/desktop/api/Unknwn/nn-unknwn-iunknown)***:***

* QueryInterface - запросить и получить указателя на другой интерфейс объекта по его ID
* AddRef – увеличить на 1 счетчик ссылок на интерфейс;
* Release – уменьшить счетчик ссылок на интерфейс;

***IClassFactory*** - Стандартный интерфейс, который поддерживает фабрики класса для создания компонентов. Он наследуется от IUnknown так что реализует его методы, а также еще 2 метода

* CreateInstance - создает экземпляр компонента
* LockServer – управляет счетчиком активных блокировок сервера. обеспечивающий блокировку программы сервера в памяти.

1. Для чего применяется утилита Regsvr32?

Для регистрации и отмены (regsvr32 /u) регистрации COM/DLL в реестре.

1. Перечислите экспортируемы функции COM-сервера типа INPROC.

DllGetClassObject

DllRegisterServer

DllUnregisterServer

DllCanUnloadNow

DllInstall

1. Перечислите счетчики, которые должен поддерживать COM-сервер.

Счетчик ссылок на интерфейсы

Счетчика ссылок на экземпляр компонентов

Счетчик блокировок

1. Назначение метода QueryInterface.

Запросить и получить указатель на другой интерфейс объекта по его IID.

1. Назначение метода AddRef.

Увеличивает счетчик ссылок на интерфейс компонента на 1.

1. Назначение метода Release.

Уменьшает счетчик ссылок на интерфейс компонента на 1. Кроме того, метод Release уничтожает экземпляр компонента, если обнаруживает, что после уменьшения счетчика ссылок на 1, его значение стало равным 0.

1. Назначение метода CreateInstance.

CreateInstance - создает экземпляр COM-объекта(компонента). Возвращает указатель на созданный объект

1. Назначение метода ServerLock.

LockServer - управляет блокировкой сервера COM.

Если сервер заблокирован, он не будет выгружаться из памяти, даже если нет активных ссылок на COM-объекты.

LockServer(TRUE) счетчик активных блокировок увеличивается на 1. Если счетчик блокировки COM-сервера установлен в значение больше нуля, это предотвращает выгрузку сервера из памяти, даже если все клиенты освободили свои ссылки на объекты.

LockServer(FALSE): клиент вызывает функцию LockServer с параметром FALSE. Это уменьшает счетчик блокировки на 1. Когда счетчик блокировки достигнет нуля (то есть, сервер разблокирован), COM-сервер может быть выгружен из памяти, если не существует активных клиентов.

1. Напишите три условия при котором DllCanUnloadNow возвращает Succeeded-значение.

DllcanUnloadNow – проверяет, можно ли выгрузить библиотеку

1. Отсутствие ссылок на объекты, т. е. нет активных объектов

2. Нет блокировок сервера (Счетчик блокировок сервера = 0)

3.Все ресурсы, выделенные библиотекой DLL, были корректно освобождены и все её операции завершены успешно

1. Что функция DllGetClassObject возвращает последним параметром?

Возврат указателя на объект фабрики классов.

1. Каким способом вызываются функции DllRegisterServer, DllUnregisterServer и DllInstall?

Вызываются утилитой **regsvr32**

