ЧЕРНІГІВСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

Коледж транспорту та комп'ютерних технологіЙ

Освітньо-кваліфікаційний рівень молодший спеціаліст

Спеціальність 122 «Комп’ютерні науки та інформаційні технології»

Навчальна дисципліна Розробка клієнт-серверних застосувань

**Екзаменаційний білет № 3**

1. Вид розподіленої системи, в якій є сервер, що виконує запити клієнта, причому сервер і клієнт спілкуються між собою з використанням того або іншого протоколу це :

а) клієнт

б) сервер

**в) клієнт-сервер**

г) сервер додатків

1. У стандартній файл-серверній архітектурі вся відповідальність за отримання та обробку даних, а також за підтримку цілісності бази даних лежить на:

а) додатку

**б) сервері**

в) клієнті

г) сервері та клієнті

1. Послідовна сукупність операцій над даними це:

а) тринер

**б) транзакція**

в) процедура

г) функція

1. Додаток включає модулі для:

**а) організації діалогу з користувачем, бізнес-правила, ядро СУБД**

б) організації діалогу з користувачем, бізнес-правила

в) бізнес-правила, ядро СКБД

г) організації діалогу з користувачем, ядро СУБД

1. Головною передумовою появи клієнт-серверної технології є:

**а) число користувачів більше 10 - 15 осіб**

б) число користувачів менше 10 - 15 осіб

в) число користувачів більше 50 осіб

г) число користувачів менше 50 осіб

1. В додатках клієнт-сервер обсяг переданої інформації може бути радикально скорочений за рахунок:

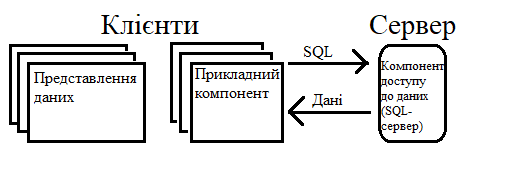
а) використання низькорівневіх запитів до даних

**б) використання запитів до даних**

в) використання високорівневих запитів до даних

г) архітектурі системи

1. За допомогою схеми взаємодії зобразіть основні елементи «моделі віддаленого доступу до даних». Вкажіть які дані передаються між основними компонентами.



1. Дайте визначення поняттю «клієнт-сервер».

Архітектурний підхід, що використовується в розподілених системах, де взаємодія між компонентами відбувається за принципом клієнтів та серверів. Клієнти- це зазвичай користувачі або програми, які виконують запити до сервера для отримання певних послуг або ресурсів. Сервери, з свого боку, це компоненти, які надають послуги, ресурси або оброблюють запити від клієнтів. На серверах лежить відповідальність за всі дані.

1. Опишіть COM – об’єкт який буде виконувати множення двох чисел. Для виконання задачі реалізуйте клас для математичної операції в якому будуть використовуватися всі необхідні методи COM – об’єкта (метод IUnknown, фабрика класів IClassFactory), передбачити у класі глобальний унікальний ідентифікатор (GUID). Для класу передбачити конструктор та деструктор.

#include <Windows.h>

#include <iostream>

// Уникальный идентификатор (GUID) для интерфейса IMyMath

static const GUID IID\_IMyMath = { 0xA1234567, 0xB123, 0xC123, { 0xD1, 0xE1, 0xF1, 0x11, 0x22, 0x33, 0x44, 0x55 } };

// Интерфейс IMyMath, который содержит метод умножения двух чисел

class IMyMath : public IUnknown

{

public:

virtual HRESULT \_\_stdcall Multiply(int a, int b, int\* result) = 0;

};

// Класс MyMath, который реализует интерфейс IMyMath

class MyMath : public IMyMath

{

private:

ULONG m\_refCount; // Счетчик ссылок

public:

// Конструктор

MyMath() : m\_refCount(1)

{

std::cout << "MyMath: Constructor called" << std::endl;

}

// Деструктор

virtual ~MyMath()

{

std::cout << "MyMath: Destructor called" << std::endl;

}

// Реализация методов IUnknown

HRESULT \_\_stdcall QueryInterface(const IID& riid, void\*\* ppvObject)

{

if (riid == IID\_IUnknown || riid == IID\_IMyMath)

{

\*ppvObject = this;

AddRef();

return S\_OK;

}

else

{

\*ppvObject = NULL;

return E\_NOINTERFACE;

}

}

ULONG \_\_stdcall AddRef()

{

return InterlockedIncrement(&m\_refCount);

}

ULONG \_\_stdcall Release()

{

ULONG refCount = InterlockedDecrement(&m\_refCount);

if (refCount == 0)

{

delete this;

}

return refCount;

}

// Реализация метода умножения

HRESULT \_\_stdcall Multiply(int a, int b, int\* result)

{

if (result == NULL)

{

return E\_POINTER;

}

\*result = a \* b;

return S\_OK;

}

};

// Функция создания объекта MyMath

extern "C" HRESULT \_\_stdcall CreateMyMathObject(IMyMath \* \*ppMyMath)

{

if (ppMyMath == NULL)

{

return E\_POINTER;

}

MyMath\* pMyMath = new MyMath();

if (pMyMath == NULL)

{

return E\_OUTOFMEMORY;

}

\*ppMyMath = pMyMath;

(\*ppMyMath)->AddRef();

return S\_OK;

}

int main()

{

MyMath myMath;

int result;

myMath.Multiply(6, 5, &result);

std::cout << result << std::endl;

return 0;

}