

CORBA (*Common Object Request
Broker Architecture*)
в Java

CORBA (*Common Object Request Broker Architecture*)

Понятие CORBA .

- **Архитектура CORBA**
- **IDL (Interface Definition Language)**
- **Брокер объектных запросов (ORB).**
- **Этапы создания проекта на основе CORBA**

Контрольные вопросы .

Вопрос 1

Понятие CORBA

Основные термины

- OMG (сокр. от англ. Object Management Group)—
- 800 организаций
- С 1989 American Airlines один из основателей OMG.
- Некоторое время в консорциуме участвовала корпорация Microsoft.

- **CORBA - Common Object Request Broker Architecture**
- — общая архитектура брокера объектных запросов) — одна из технологий написания распределённых приложений, продвигаемый консорциумом (рабочей группой) **OMG** - объектно-ориентированная
- задача - обеспечение взаимодействия программ,
- суть — независимая от языка работа с удаленными объектами

ПОНЯТИЕ CORBA

- Задача CORBA - интеграция изолированных систем.
- **Цель** - создание распределенных компьютерных систем.
- **Преимущества** :
 1. прозрачность вызова (*invocation transparency*),
 2. прозрачность реализации (*implementation transparency*)
 3. и прозрачность локализации (*location transparency*)

CORBA использует:

1. инкапсуляцию,
2. наследование,
3. полиморфизм,
4. динамическое связывание.

ППО:

1. через SQL,
2. RPC (Remote Procedure Call),
3. TP мониторы (Transaction Processing)
4. Groupware –
5. MOM - Message-Oriented Middleware

Основные понятия

- **1. CORBA-объект (target object) –**
 - object reference
- **2. Идентификатор объекта (object ID) -**
 - (Portable Object Adapter).
- **3. Сервант**

необходимо отделять CORBA-объекты от сервантов.

4. Скелетон

5. Активизация

6. Активизированные объекты

6. Временные объекты

8. Деактивизация

9. Объектный адаптер

10. ORB (Брокер объектных запросов) - ОСНОВНОЙ КОМПОНЕНТ CORBA..

■ коммутационная плата

1. библиотека

2. процесс-демон.



Архитектура CORBA

четыре основных элемента:

- 1) ORB
- 2) Сервисы CORBA
- 3) CORBA facilities
- 4) Application Objects

Технология работы:

- «заглушка» (*Stub*) – в клиентской части.
- «скелет» (*Skeleton*) – у серверного объекта.
 - реализация серверного объекта.
 - уникальная объектная ссылка.
 - передача ORB данных от клиента.

- **Интерфейс** не зависит:

1. от местоположения реализации объекта,
2. языка программирования.

- При создании определения интерфейса оговариваются:

1. операции
2. параметры
3. исключения

интерфейс

- взаимодействие с клиентом объекта,
- доступ к реализациям объектов.

- **Клиент** - это приложение
- **реализация объекта** - это код и данные
- **ORB**- все действия для:
 1. нахождения,
 2. реализации,
 3. подготовке реализации
 4. передаче данных

интерфейс

```
public interface Hello extends java.rmi.Remote
{
    String sayHello() throws java.rmi.RemoteException;
}
```

класс

```
public class HelloImpl extends PortableRemoteObject implements
Hello
{
    public HelloImpl() throws RemoteException {}

    public String sayHello() throws RemoteException
    {
        return "Hello World!\n";
    }
}
```


ПОНЯТИЕ CORBA

Сформируем серверную часть (3 пункта из предыдущего списка)

```
import org.omg.CORBA.ORB;
import org.omg.CORBA.Object;
import javax.rmi.PortableRemoteObject;
public class helloServer{
    public static void main(String[] args)  {
    try {
    ORB orb = ORB.init(args, null);
    Hello hello = new HelloImpl();
    String helloIOR = orb.object_to_string((Object)
    PortableRemoteObject.toStub(hello));
    \\\"ior«(Interoperable Object Reference)
    Util.writeIOR(helloIOR, \"ior\", true);
    System.out.println(\"waiting for invocations ...\");
    orb.run();
    } catch (Exception ex) {
    ex.printStackTrace();}  }}
```

РОА и ВОА

- Объектные адаптеры
- Basic Object Adapter (базовый объектный адаптер), или ВОА,
- Portable Object Adapter (РОА - переносимый объектный адаптер).

ROA - возможность
отделить *доступ* к серванту от
самого серванта.

- у клиента есть *объектная ссылка*,
информация о
местонахождении *объекта* OA.
- OA - решение,
какой *объект* (сервант) может
обработать данный вызов.

Работа сервера на POA состоит из (общий случай):

1. ссылка на корневой адаптер POA;
2. определение политик нового POA;
3. создание нового POA
4. создание серванта;
5. активация POA.

CORBA-сервер

- на основе стандартного механизма POA
- экземпляр ORB
- ссылка на службу имен
- объект "обрезается" (**narrow**)
- публикуется в службе имен.
- запускается ORB

1. 1. **ORB orb=ORB.init(args, null);**
2. 2. метод **resolve_initial_references()**
3. Object, привести к типу POA с помощью **narrow()** в Helper-классе объекта POA:

POA

```
rootPOA=POAHelper.narrow(orb.resolve_initial_references("RootPOA"));
```

4. POA interface =
`poaRoot.create_POA(new_POA_name, poaRo
ot.the_POAManager(),
TestInterfacePolicies);`
5.
`interface.activate_object_with_id(new_POA_n
ame.getBytes(),
new TestInterfaceImpl());`
6. `poaRoot.the_POAManager().activate();`
7. `orb.run();`

■ Object

Management Architecture (OMA) -

эталонная архитектура
распределенных систем,.

■ Объекты CORBA – удаленные объекты
могут обращаться,

·
Объект - это нечто, что обеспечивает
один или более сервисов

Серверный объект - набор классов

особенности структуры объектной ссылки CORBA:

- формат IOR - Interoperable Object Reference
- протокол GIOP
- язык IDL.
- транспортные протоколы
- параметры для различных транспортов должны быть различными.
- ОС несколько профилей. (Совокупность параметров для ТП)

Компонентная модель CORBA (CCM)

- CCM была введена начиная с CORBA 3.0
- стандартный каркас приложения для компонент CORBA.
- Контейнер - набор служб - сервисов CORBA ,
- работу выполняет а контейнер приложения.
- значительно снижает сложность реализации.

Сервисы CORBA (CORBAservices)

- базовые сервисы, которые доступны всем объектам, подключенным к коммуникационной шине ORB.



IDL (Interface Definition Language)

IDL (Interface Definition Language)

- описательный язык,
- расширение **.idl**
- нет деталей реализации.
- Код, который реализует операции - сервант (servant).
- конкретные языки программирования.

На IDL существует несколько возможных структурных объединений:

■ Enum

```
module Finance {  
    enum Currency {pound, dollar, yen, franc};  
    interface Account {  
        readonly attribute float balance;  
        readonly attribute Currency balanceCurrency;  
        ...  
    };  
};
```

■ Struct

A struct data type allows you to package a set of named members of various types, for example:

```
module Finance {  
    struct CustomerDetails {  
        string name;  
        short age;  
    };  
    interface Bank {  
        CustomerDetails getCustomerDetails(in string name);  
        ... }; };
```

■ UNION - аналогично

Пример использования типа String:

```
module Finance {  
    interface Bank {  
attribute string sortCode<10>;  
attribute string address;  
        ...  
    };  
};  
typedef sequence<long> Unbounded;  
typedef sequence<long, 31> Bounded;
```

Спецификацию можно почитать на

<http://www.ibm.com/developerworks/webservices/library/co-corbajct3.html>

Этапы создания распределенных объектных приложений на IDL :

1. описание объектов на IDL,
2. их трансляции на какой-либо язык программирования,
3. добавление бизнес-логики и компиляция.

Основные понятия

- модуль
- интерфейс
- операция
- IDL стаб (stub)

- C,
- C++,
- Java,
- Cobol,
- Smalltalk ,
- Lisp
- и Ada;
- для objective C, Visual Basic, Perl и других языков – не все стандартизированы но работают.



Брокер объектных запросов (ORB).

oRВ - это промежуточное ПО, которое устанавливает клиент-серверные отношения между объектами + спектр сервисов распределенного промежуточного программного обеспечения.

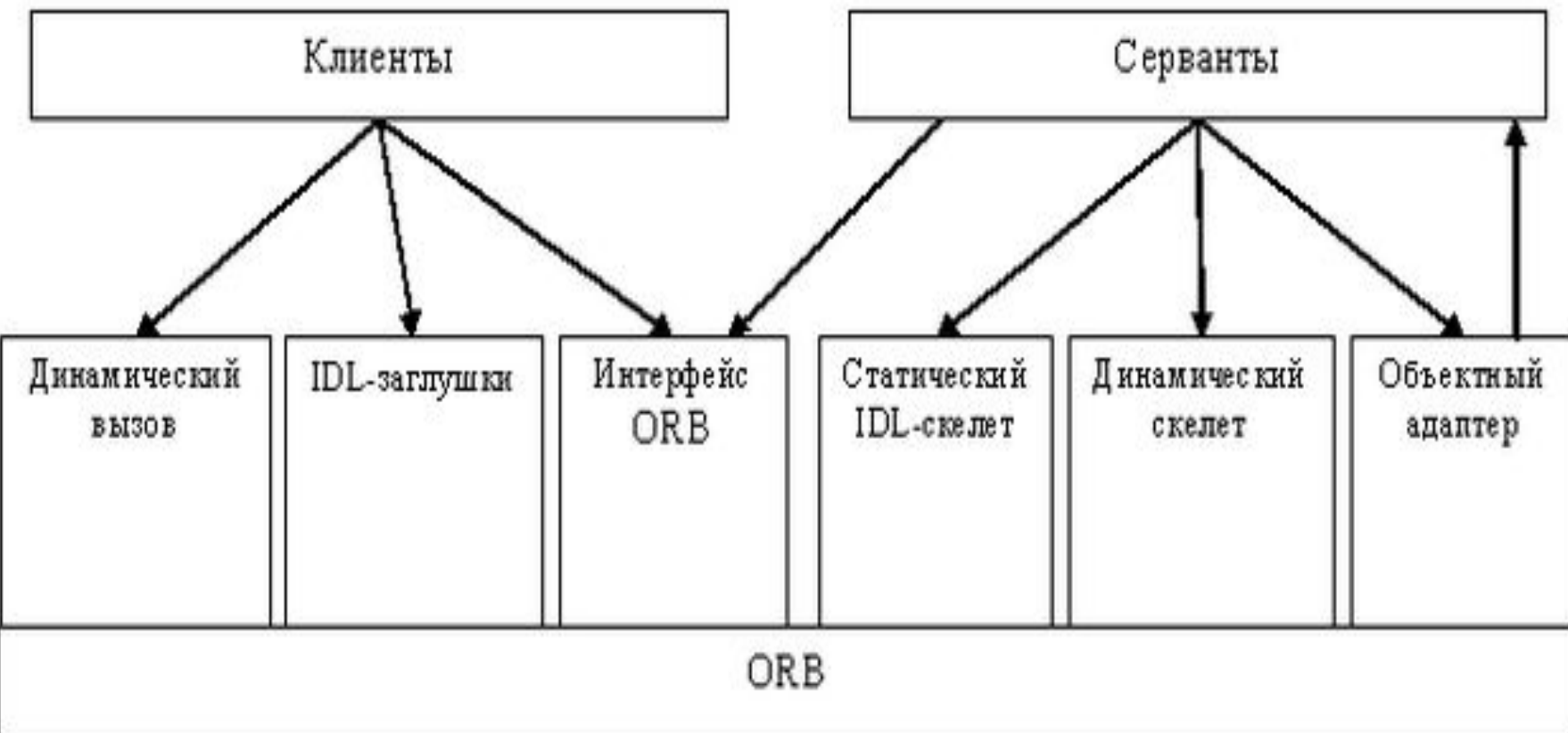
Позволяет:

- вызывать сервисы
- интероперабельность
- посылать - принимать заявки,
- поиск реализации объекта-сервера

- ВЫЗОВЫ МЕТОДОВ КОМПОНЕНТОВ;
- реализация как синхронного, так и асинхронного взаимодействия КОМПОНЕНТОВ.

- Брокер объектных запросов играет в ОМА основную роль.
- Взаимодействие Клиент – посредством
 1. *заглушки*
 2. Динамический интерфейс
 3. АРІ БОЗ.
- Брокер и сервант:
 1. статический скелет,
 2. динамический интерфейс
 3. объектный адаптер.
- Рис. иллюстрирует взаимодействие с объектным брокером.

ПОНЯТИЕ CORBA



- Пример создания объекта ORB на стороне клиента:

```
ORB orb = ORB.init(args, null);  
org.omg.CORBA.Object objRef =  
orb.resolve_initial_references("NameService");
```

Пример формирования стаба при маршалинге:

```
Hello hello = (Hello) orb.string_to_object(helloIOR);
```


Интероперабельность брокеров –

- CORBA определяет среду для различных реализаций ORB, поддерживающих общие сервисы и интерфейсы.
- обеспечивает мобильность клиентов

- Совместная работа брокеров поддерживается *Универсальным Межброкерным Протоколом* (General Inter-ORB Protocol, сокращенно GIOP).
- *Межброкерный Протокол Internet* (Internet Inter-ORB Protocol, сокращенно IIOP).