

РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

Технология CORBA

РАННИЕ ТЕХНОЛОГИИ ПОСТРОЕНИЯ РАСПРЕДЕЛЕННЫХ СИСТЕМ

- ◎ Удаленный вызов процедур
 - ◎ Сильно завязан на базовом протоколе
 - ◎ Сильно завязан на семантике языка
 - ◎ Проприетарные протоколы
- ◎ Ориентированы на гомогенные вычислительные среды
- ◎ Обычно ориентированы на связь 1 к 1
 - ◎ Не высоко масштабируемы

Эволюция РВС

- ◎ Объектно-ориентированный дизайн РВС
 - ◎ Связан с расцветом ОО-подхода
- ◎ Многозвенные архитектуры
- ◎ Внедрение “унаследованных” приложений
- ◎ Интеграция корпоративных приложений
- ◎ Работа на базе TCP/IP
- ◎ Интернет-вычисления

OBJECT MANAGEMENT GROUP (OMG)

OBJECT MANAGEMENT GROUP (OMG)

5

- ◎ OMG (Object Management Group) — консорциум, занимающийся разработкой и продвижением объектно-ориентированных технологий и стандартов.
- ◎ Это некоммерческое объединение, разрабатывающее стандарты для создания корпоративных платформо-независимых приложений.
- ◎ В него входит более 800 крупнейших производителей ПО
- ◎ Не отвечает за разработку реализаций своих стандартов

АРХИТЕКТУРА УПРАВЛЕНИЯ ОБЪЕКТАМИ

- ◎ Архитектура управления объектами (Object Management Architecture – ОМА) – это две основные модели, на которых основывается технология OMG
- ◎ Core Object Model
 - ◎ Абстрактное определение объекта
 - ◎ Определяет как брокер запросов объектов (ORB или Object Request Broker) взаимодействует с РВС
- ◎ Reference Object Model
 - ◎ Архитектура инфраструктуры РВС

CORE OBJECT MODEL

- ◎ Переносимый дизайн
 - ◎ Взаимодействие с объектом на основе интерфейса
 - ◎ Независимый от платформы и языка доступ
- ◎ Обеспечение взаимодействия
 - ◎ Возможность инициировать операции других объектов независимо от их расположения, платформы, языка или реализации

REFERENCE OBJECT MODEL

- ⊙ Определяет интерфейсы к инфраструктуре и сервисам удаленных объектов
- ⊙ Руководство для разработчиков и поставщиков
- ⊙ Определяет пять основных компонентов:
 - Брокер Объектных Запросов (ORB - Object Request Broker)
 - Стандартные сервисы (*Services*)
 - Общие средства (*Facilities*)
 - Отраслевые интерфейсы (*Domains*)
 - Прикладные объекты (*Application Objects*)



CORBA

Common Object Request Broker Architecture

ЧТО ТАКОЕ CORBA?

Common Object Request Broker Architecture — общая архитектура брокера объектных запросов. Включает в себя:

- ⊙ архитектуру для совместимых распределенных вычислений
- ⊙ протокол взаимодействия на основе Интернета (IIOP)
- ⊙ методы отображения языков (OMG IDL)
- ⊙ комплексные и многократно используемые сервисы

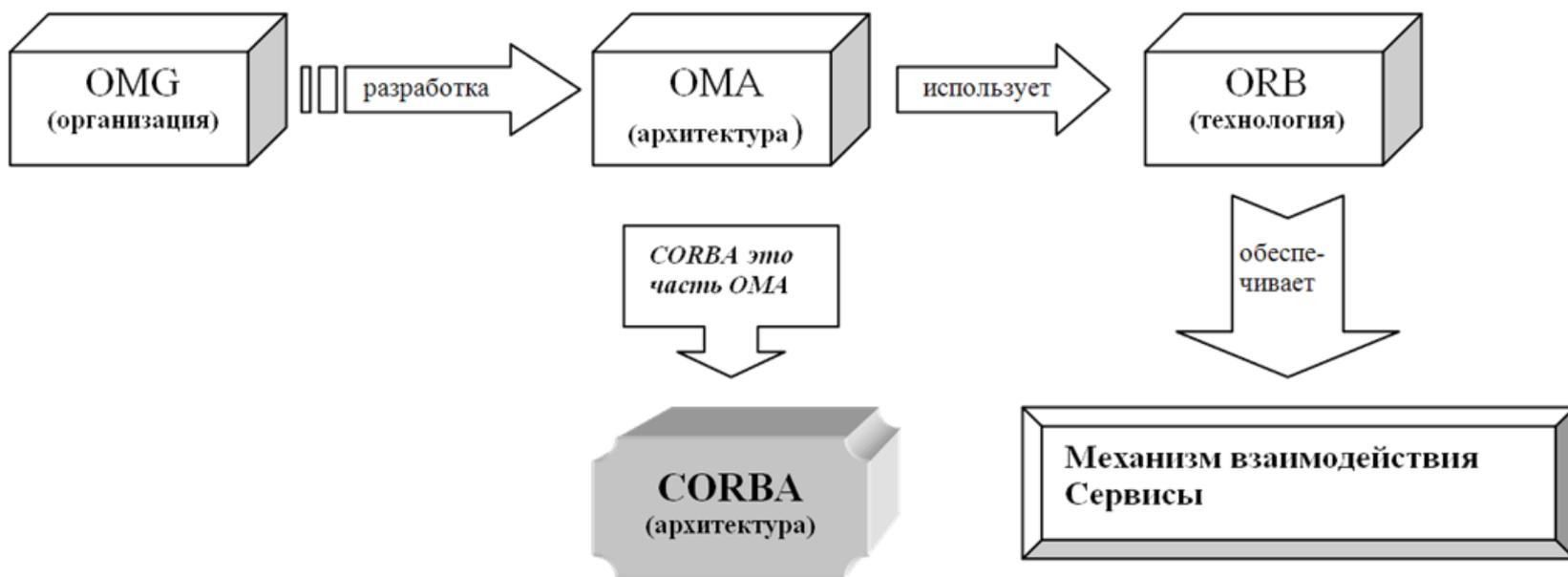
ИСТОРИЯ CORBA

- ◎ Object Management Group основана в 1989
- ◎ CORBA 1.x (91-93)
 - ◎ Экспериментальная, сосредоточена на архитектуре, IDL
- ◎ CORBA 2.0 (1996)
 - ◎ Функциональная совместимость и интеграция с COM
- ◎ CORBA 2.1 (1997)
 - ◎ Безопасность, отображение языков

ИСТОРИЯ CORBA (2)

- ◎ CORBA 2.2 (1998)
 - ◎ Совместимость с технологией DCOM, IDL/Java
- ◎ CORBA 2.3 (1999)
 - ◎ Объекты по значению, RMI/IIOP, улучшение привязки к языкам
- ◎ CORBA 2.4 (2001)
 - ◎ Улучшение QoS, асинхронный обмен сообщениями, поддержка работы в реальном времени

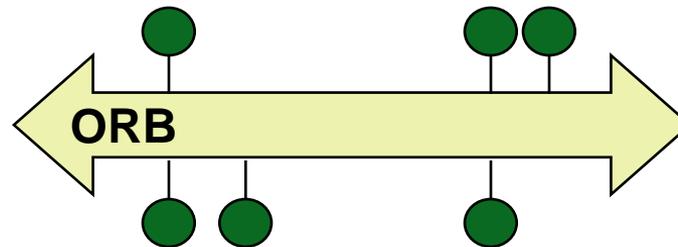
OMG, OMA, ORB и CORBA



БРОКЕР ОБЪЕКТНЫХ ЗАПРОСОВ (ORB)

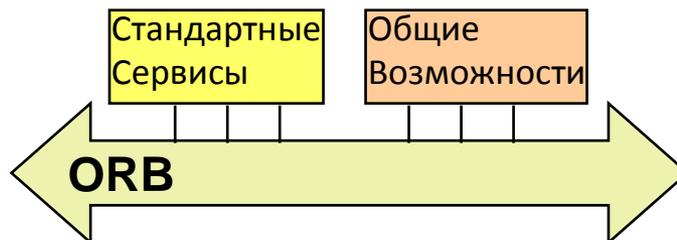
- ◎ Обеспечивает прозрачное взаимодействие между удаленными объектами посредством запросов

“... как телефонная станция, обеспечивая основной механизм для отправки и приема вызовов...”



ОБЪЕКТНЫЕ СЕРВИСЫ

- ◎ Набор общих сервисов и интерфейсов
- Стандартные Сервисы
 - Поддержка функции для внедрения и использования объектов
 - например управление временем жизни и т.п.
- Общие Возможности
 - Сервисы, которыми многие приложения могут делиться, но не являются фундаментальными
 - Менеджер печати и т.п.



ОТРАСЛЕВЫЕ И ПРИКЛАДНЫЕ ОБЪЕКТЫ

- ◎ Отраслевые объекты
 - ◎ Вертикальные (от самых общих, до самых частных) стандарты, специфичные для определенной прикладной области (производство, телекоммуникации, финансы и т.п.)
- ◎ Прикладные объекты
 - ◎ Другие объекты, относящиеся к CORBA, не стандартизованные OMG

СХЕМА ВЗАИМОДЕЙСТВИЯ ОБЪЕКТОВ ОМА

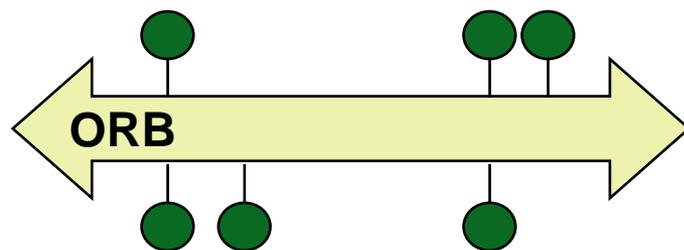


БРОКЕР ОБЪЕКТНЫХ ЗАПРОСОВ

Object Request Broker (ORB)

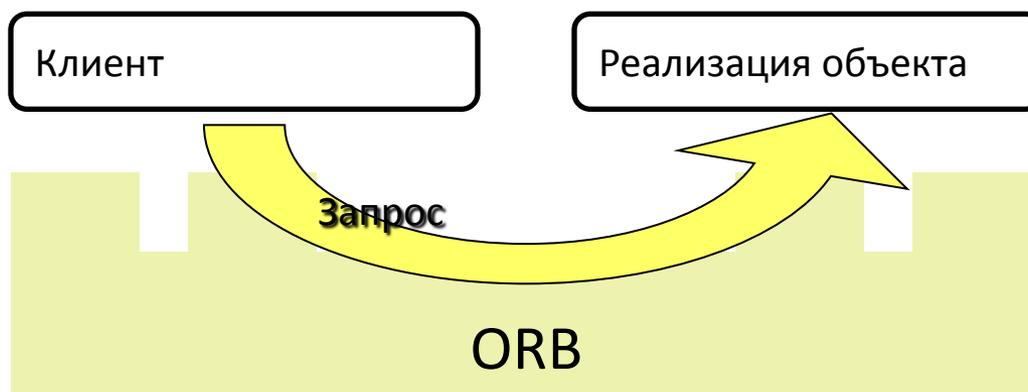
ЧТО ТАКОЕ ORB?

- ◎ Шина для работы распределенных объектов
- ◎ Скрывает механизмы передачи данных
 - ◎ Расположение
 - ◎ Вызов методов
 - ◎ Маршализация
- ◎ Язык описания интерфейсов (OMG IDL или Interface Definition Language) предоставляет независимую от языка семантику работы с объектами



ОБЪЕКТ REQUEST BROKER (ORB)

- ◎ Абстрагирует механизмы удаленных запросов и ответов
- ◎ Транспорт для удаленного вызова методов



ИСПОЛЬЗОВАНИЕ ORB

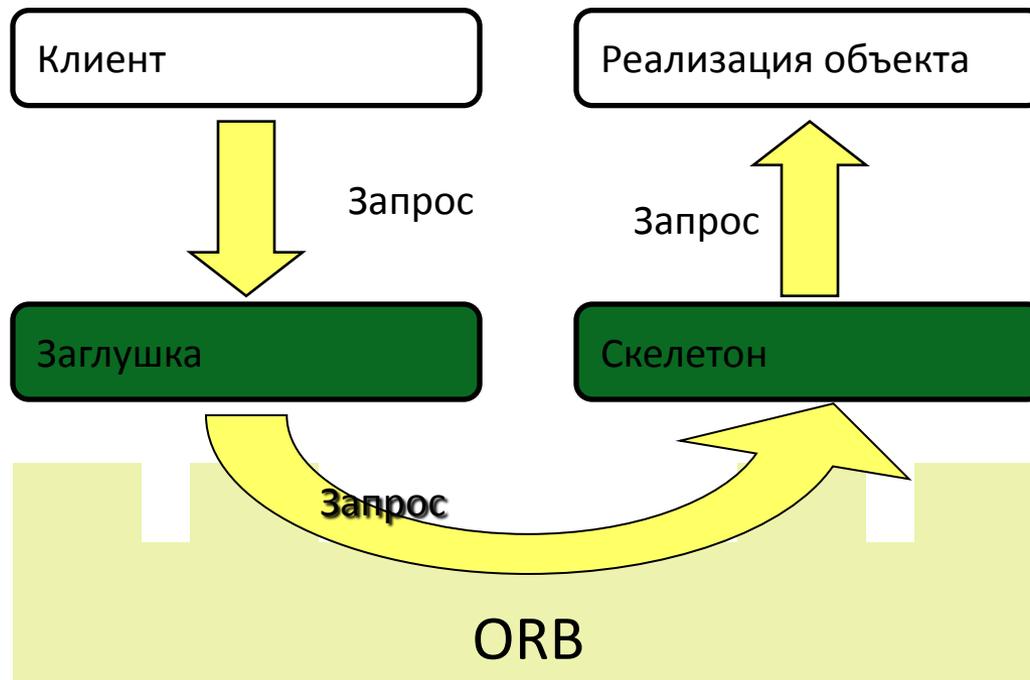
- ◎ ORB реализован как синглтон
- ◎ Инициализация ORB
 - ◎ Вызов метода `init()`

```
org.omg.CORBA.ORB orb = org.omg.CORBA.ORB.init();
```

- ◎ После инициализации производится регистрация объектов ORB посредством `Object Adapter`

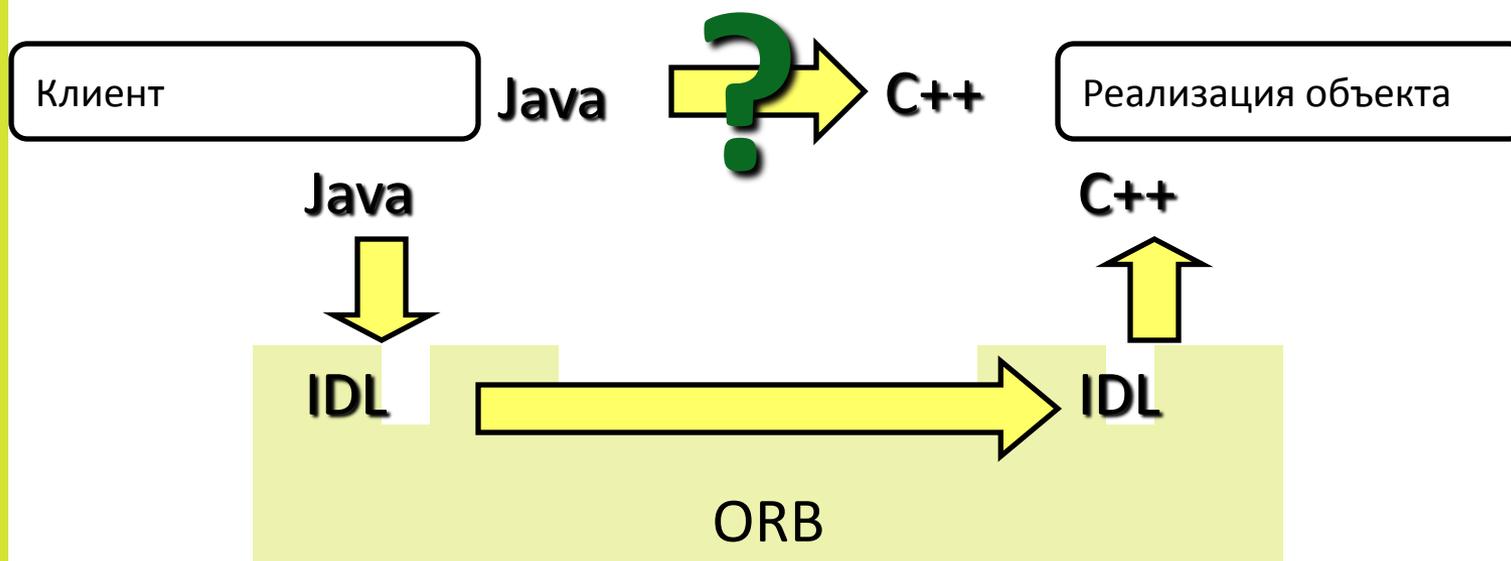
ВЫЗОВ УДАЛЕННОГО ОБЪЕКТА

22



АБСТРАКЦИЯ ORB

- ◎ ORB обеспечивает возможность взаимодействия систем, реализованных на базе различных платформ

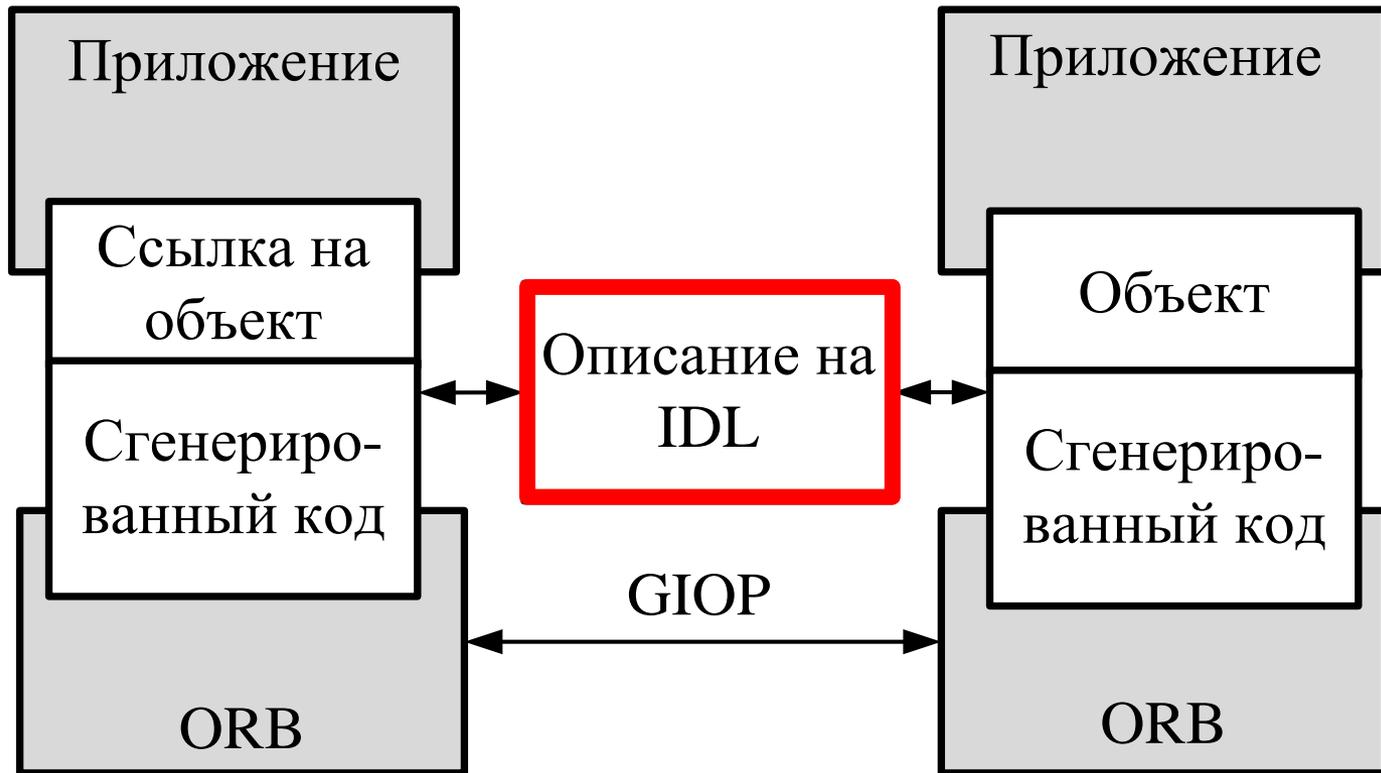


ПРОТОКОЛ GIOP

- ⊙ GIOP (General Inter-ORB Protocol) – это протокол, обеспечивающий взаимодействие между ORB в распределенной вычислительной среде.
- ⊙ Наиболее распространенная реализация этого протокола – IIOP (Internet Inter-ORB Protocol), который отображает сообщения GIOP на уровень TCP/IP.

ОСНОВНЫЕ ЭЛЕМЕНТЫ ВЗАИМОДЕЙСТВИЯ CORBA

25



ЯЗЫК ОПРЕДЕЛЕНИЯ ИНТЕРФЕЙСА

Interface Definition Language (IDL)

ЯЗЫК ОПРЕДЕЛЕНИЯ ИНТЕРФЕЙСА (IDL)

- ◎ Независимый от языка интерфейс
 - ◎ Реализовано отображение IDL на множество высокоуровневых языков программирования
- ◎ Основная парадигма
 - ◎ Обеспечить интерфейс для объектов, независим от их реализации

ОТОБРАЖЕНИЯ IDL

◎ Существуют методы отображения IDL на множество высокоуровневых языков

- C
- C++
- Smalltalk
- Lisp
- Ada '95
- COBOL
- Python
- Java

ПРИМЕР ОПРЕДЕЛЕНИЯ IDL

```

// Модуль системы ценовых предложений
module QuoteSystem
{
    // Структура данных цены
    struct Quote
    {
        string value;
    }

    // Интерфейс сервера ценовых предложений
    interface QuoteServer
    {
        // Атрибут определяющий фондовую биржу
        string exchange;
        // Исключение «Неизвестный идентификатор»
        exception UnknownSymbolException { string message; };
        // Поиск предложения по идентификатору
        Quote getQuote (in string symbol)
            raises (UnknownSymbolException);
    };
};

```

← **Пакет** QuoteSystem

← **член данных**

↑ **Возвращаемый тип**

↑ **Аргумент**
(направление и тип)

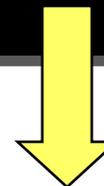
↑ **заявляет, что метод формирует исключение**

Интерфейс объекта QuoteServer

МОДУЛИ И ИНТЕРФЕЙСЫ

- ◎ Модуль
 - ◎ Отображается в пакет Java
 - ◎ Может содержать множество интерфейсов
- ◎ Интерфейс
 - ◎ Отображается в набор связанных классов и интерфейсов

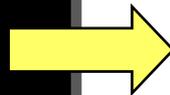
```
module QuoteSystem
{
    interface QuoteServer
    {
        ...
    };
};
```



`QuoteSystem.QuoteServer`

- ◎ Структуры
 - ◎ Отображаются в Java-классы
 - ◎ Разрабатываются для хранения информации

```
struct Quote
{
    string value;
}
```

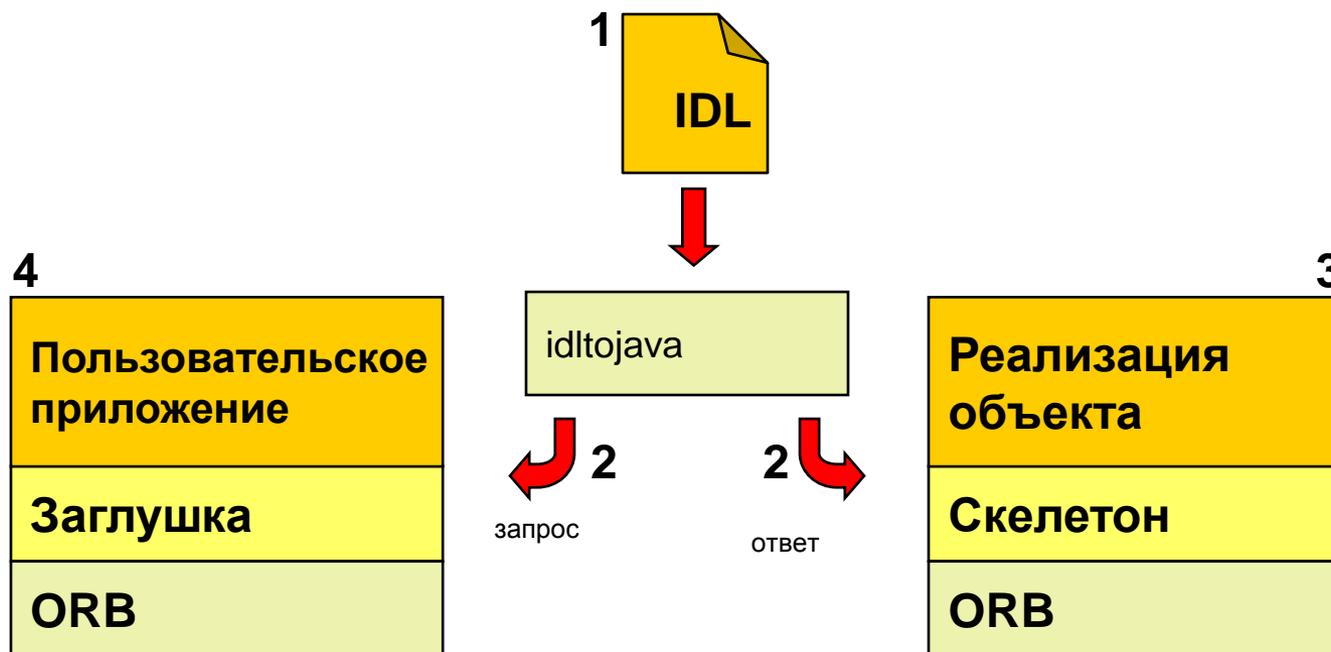


```
public final class Quote
{
    public String value;
}
```

РАЗРАБОТКА ОБЪЕКТОВ CORBA

1. Определение интерфейса на IDL
2. Обработка IDL для создания кода заглушки и скелетона
3. Создание кода реализации объекта (сервер)
4. Создание кода использования данного объекта (клиент)

ПРОЦЕСС РАЗРАБОТКИ



Процесс:

1. Создание файла IDL
2. Компиляция посредством `idltojava` (заглушка и скелетон создается автоматически)
3. Реализация объекта
4. Создание пользовательского приложения

РЕЗЮМЕ

CORBA vs. RMI

CORBA	RMI
Платформо- независимая	Только для Java
Не зависит от протокола	Проприетарный протокол (JRMP)
Независим от языка (IDL)	Java

ПРЕИМУЩЕСТВА CORBA

- ① использование IDL для описания интерфейсов позволяет разрабатывать программные компоненты независимо от языка программирования и базовой операционной системы;
- ① поддержка богатой инфраструктуры распределенных объектов;
- ① прозрачность вызова удаленных объектов.

ПРОБЛЕМЫ CORBA

- ⊙ плохая совместимость различных реализаций технологии CORBA от различных поставщиков;
- ⊙ проблемы взаимодействия узлов CORBA через Интернет;
- ⊙ отсутствие связи с World Wide Web;
- ⊙ несогласованность, отсутствие компонентной модели;
- ⊙ чрезмерная сложность разработки (более 200 строк определения интерфейсов можно было бы уложить в 30, т.к. 170 – не несут функциональной поддержки)

ЧТО ВМЕСТО CORBA?

- ◎ Microsoft: COM, DCOM
 - ◎ Проприетарный, завязан на Microsoft Windows
- ◎ Java: Enterprise JavaBean
- ◎ SOAP и Веб-сервисы