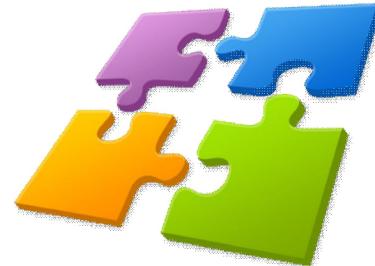


## مدل اشیای توزیع شده

- روش RPC برای اولین بار امکان به اشتراک گذاری عملکرد بین برنامه های کاربردی مختلف که ممکن است روی ماشینهای متفاوتی اجرا شوند را فراهم کرد.
- با این وجود، این روش معایب بسیاری داشت:
  - پایین بودن قابلیت استفاده مجدد کد
  - وابستگی به زبان
  - ثابت بودن نقش کلاینت و سرور
  - ارتباط نقطه به نقطه
  - فرآخوانی همگام
  - عدم مقیاس پذیری



## مدل اشیای توزیع شده ...

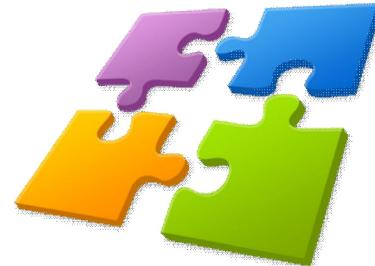
- راه حل پیشنهادی برای حل معایب روشن RPC باید دارای ویژگی های زیر باشد:
  ١. جداسازی کد تجمیع و کد ارتباطات در یک مولفه مستقل
    - مزیت:
    - افزایش قابلیت استفاده مجدد
  - حل مشکل رابطه "نقطه به نقطه" بین کلاینت و سرور (زیرا برنامه های کاربردی متعددی می توانند با استفاده از این مولفه مستقل به یکدیگر متصل شوند.)
  - "همتا به همتا" بودن رابطه بین کلاینت و سرور
- ٢. عدم وابستگی به زبان و عدم وابستگی به پلت فرم

راه حلی که شامل این ویژگی ها می باشد: استفاده از اشیای توزیع شده



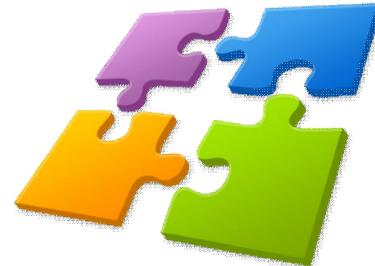
## مدل اشیای توزیع شده ...

- اشیای توزیع شده مفاهیم مرتبط با کلاس ها و برنامه نویسی شی گرا را توسعه می دهند.
- سه رکن اصلی برنامه نویسی شی گرا:
  - **کلاس:**
  - ساختار تعریف شده توسط کاربر که داده و رفتار مربوط به یک نهاد معین را کپسوله می کند.
  - مزایای کپسوله سازی: برنامه نویسی قابل اطمینان تر و قوی تر
- **ارث بری:**
  - منجر به استفاده مجدد از کد در سطح برنامه نویسی می شود.
- **چند ریختی:**
  - یعنی توابعی که کار مشابهی انجام میدهند میتوانند نام یکسانی داشته باشند.
  - البته ویژگی سوم در این بخش اهمیت زیادی ندارد



## مدل اشیای توزیع شده ...

- نشی: نمونه خاصی از یک کلاس معین در زمان اجرا.
- قبل از اشیای توزیع شده فقط اشیایی که متعلق به یک برنامه کاربردی بودند می توانستند با یکدیگر در زمان اجرا تعامل داشته باشند.
- با اشیای توزیع شده اشیای متعلق به برنامه های مختلف نیز می توانند با یکدیگر تعامل داشته باشند و داده و عملکرد را در زمان اجرا مبادله کنند.



# مدلهای مبتنی بر اشیای توزیع شده

وابسته به پلت فرم ویندوز

مستقل از پلت فرم، وابسته به زبان Java

مستقل از پلت فرم و مستقل از زبان

مدل شی مولفه توزیع شده /  
Distributed Component Object Model

احضار متده راه دور /  
Remote Method Invocation

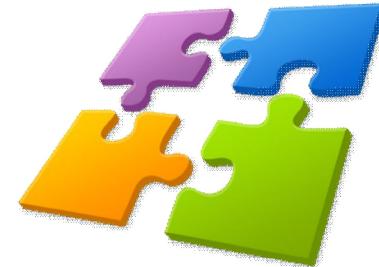
معماری عمومی رابط درخواست شی /  
Common Object Request Broker Architecture

# CORBA مدل



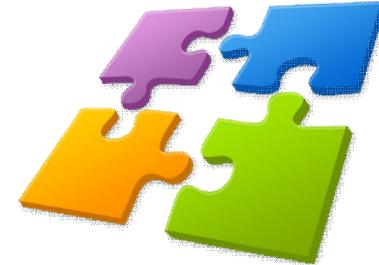
- مولفه های اصلی مدل CORBA
  - رابط درخواست شی / ORB /
  - زبان تعریف رابط / IDL /
- مدل شی CORBA
- نقشه های کلاینت و سرور در CORBA
- سرویس های CORBA

# CORBA مدل / ORB رابط درخواست شی



- مهم ترین مولفه در CORBA
- کد تجمعیع و کد فراخوانی های شبکه را در یک مولفه نرم افزاری مجزا، کپسوله می کند.
- کلیه نیازهای تجمعیع و تفکیک آرگومان ها و مقادیر بازگشته اشیای توزیع شده را تامین می کند:
  ۱. باعث میشود چندین برنامه کاربردی بتوانند از یک کد، به طور مجدد استفاده کنند.
  ۲. با کنار گذاشتن روش یکپارچه سازی نقطه به نقطه، باعث از بین رفتن پیوند قوی می شود.
  ۳. چندین برنامه کاربردی روی یک ماشین میتوانند از طریق این رابط، با یکدیگر و با برنامه های کاربردی مقیم در ماشین های دیگر، ارتباط برقرار کنند.

# مدل CORBA رابط درخواست شی / ... ORB

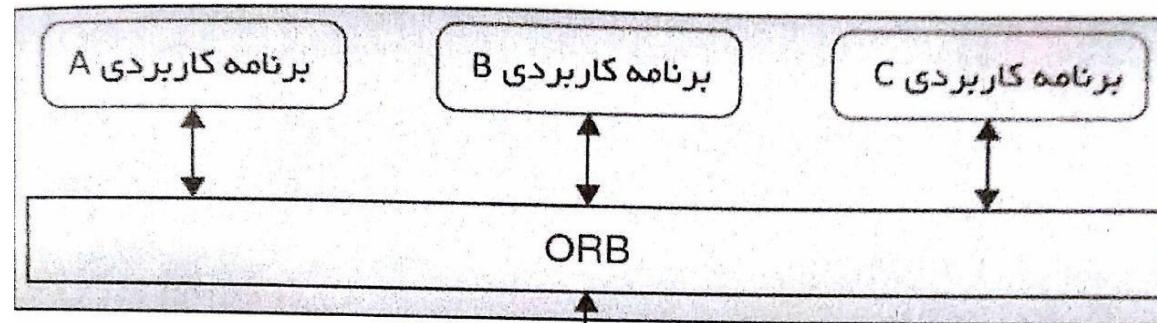


- کلیه نیازهای تجمعیع و تفکیک آرگومان ها و مقادیر بازگشته اشیای توزیع شده را تامین می کند: ...
- ۴. ارتباط بین مولفه ها مستقل از پلت فرم است زیرا پارامترها به محض انتقال به فرمت مستقل از پلت فرم تبدیل می شوند و به محض پذیرش به فرمت خاص پلت فرم مقصد، تبدیل می شوند.
- ۵. تفاوت های سخت افزاری، مشکل ساز نخواهد بود زیرا رابط درخواست شی این تبدیلات را به طور خودکار انجام میدهد.
- ۶. فرایند تجمعیع و تفکیک کاملاً توسط رابط درخواست شی انجام میشود و برنامه نویس نگران جزئیات نحوه این تجمعیع و تفکیک نیست.

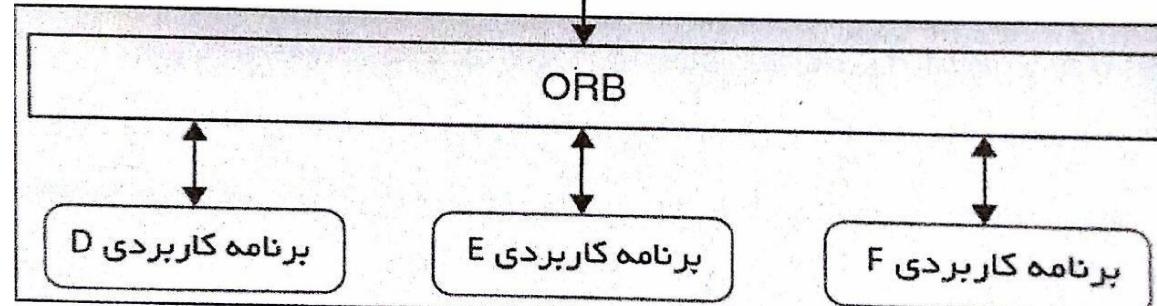
# CORBA مدل رابط درخواست شی / ... ORB



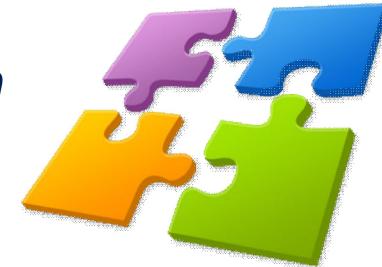
ماشین ۱



ماشین ۲

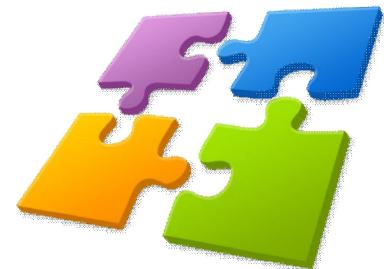


# CORBA مدل رابط درخواست شی / ... ORB

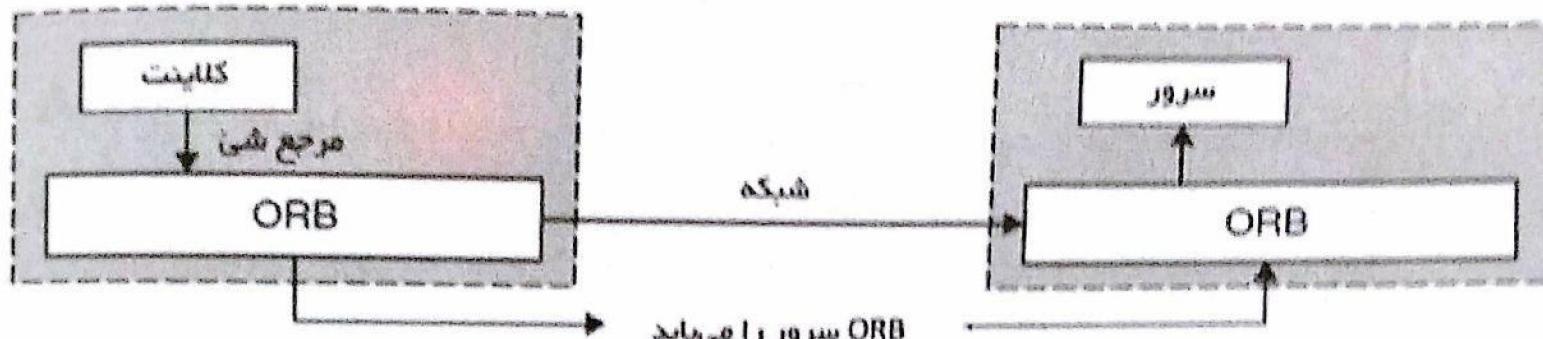


مراحل اصلی برقراری ارتباط بین برنامه های کاربردی از طریق رابط درخواست شی:

۱. هنگامی که یک شی یا مولفه می خواهد از سرویسهای شی یا مولفه دیگری استفاده کند، ابتدا مرجعی برای شی ارائه دهنده سرویس به دست می آورد.
۲. سپس رابط درخواست شی، پیاده سازی شی متناظر را از طرف کلاینت مکان یابی می کند.
۳. هنگامی که سرور مکان یابی شد، رابط درخواست شی اطمینان میدهد که سرور آماده دریافت درخواست است.
۴. رابط درخواست شی در سمت کلاینت، پارامترهای احضار شده متده را می پذیرد و پارامترها را در شبکه تجمعی میکند.
۵. رابط درخواست شی در سمت سرور، پارامترها را تفکیک میکند و آن ها را به شی سرور تحویل میدهد.
۶. مقدار برگشته به ترتیب در سمت سرور و سمت کلاینت به روشهای مشابه، تجمعی و تفکیک میشود.



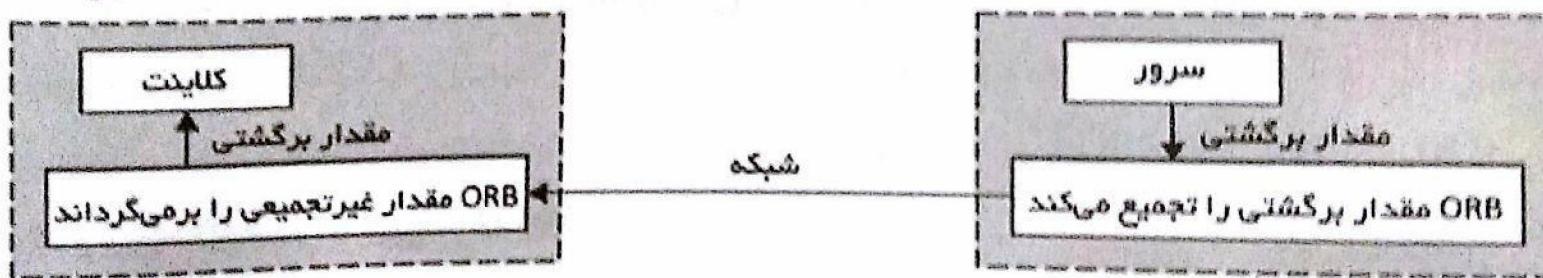
مرحله ۱



مرحله ۲



مرحله ۳



# CORBA مدل زبان تعریف رابط / IDL

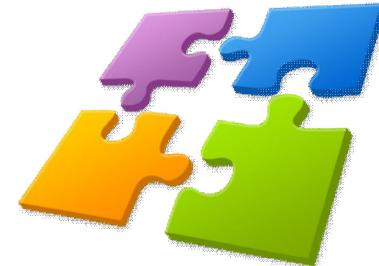


- در روش RPC، رابط‌ها به یک زبان خاص نوشته می‌شود، که برنامه‌های کلاینت و سرور نیز به همان زبان نوشته شده‌اند.

## :IDL •

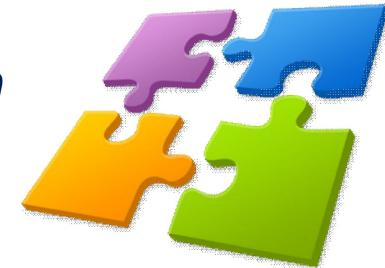
- ابزاری است برای معرفی رابط‌ها به روشهای مستقل از زبان.
- یک زبان برنامه نویسی نیست و فقط می‌تواند برای تعریف رابط‌ها استفاده شود.
- صرف نظر از جزئیات تنها نام و امضای متدها را تعیین می‌کند.
- وابسته به هیچ زبان برنامه نویسی خاصی نیست.
- مسئول تبادل صحیح داده بین زبان‌های متفاوت است، زیرا رابط‌های تعریف شده توسط IDL، می‌توانند به هر زبانی نگاشت شوند.

# CORBA مدل زبان تعریف رابط / ... IDL



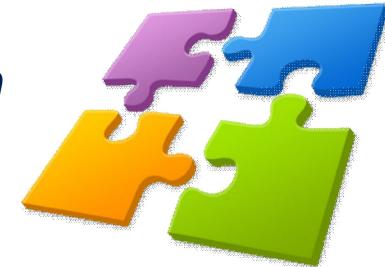
- نگاشت های استانداردی برای برخی زبان های برنامه نویسی رایج مانند C، C++، Java و ... وجود دارد.
- این نگاشت ها، ساختارهای زبان IDL را به ساختارهای یک زبان ویژه نگاشت میکنند
- به عنوان مثال در Java، رابط IDL به رابط جاوا نگاشت می شود و در C++ رابط IDL به کلاس نگاشت میشود.
- این نگاشت با استفاده از یک ابزار خودکار انجام میشود که فایل IDL را به عنوان ورودی میگیرد و تعدادی فایل تولید می کند که برای توسعه کلاینت و سرور به زبانی معین استفاده می شوند.
- به عنوان مثال، ابزار idl2j فایلهای مناسب را برای توسعه کلاینت و سرور به زبان جاوا تولید میکند.

# CORBA مدل ربان تعریف رابط / ... IDL



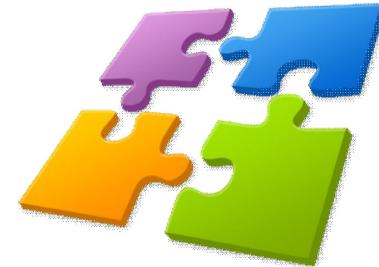
```
Listing 5.1: A sample of IDL interface definition
module Test {
    interface square {
        attribute double arg1;
        double getSquare (in double arg1);
    };
};
```

# CORBA مدل CORBA مدل شی



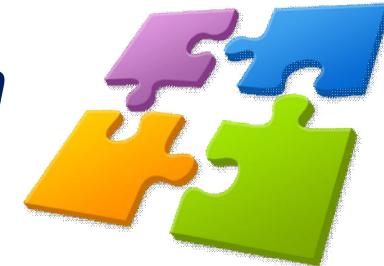
- عناصر یک شی مدل :CORBA
- پشتیبانی از توزیع تقریباً شفاف اولیه
- ارجاعات شی
- آدپتورهای شی (که باعث میشود اشیای توزیع شده با استفاده از رابهای درخواست شی با یکدیگر ارتباط برقرار کنند.)

# CORBA مدل ... CORBA مدل شی



- توزیع شفاف:
  - توزیع اشیا در صورتی شفاف است که کلاینت از سرویس های ارائه شده توسط شی دیگری که از مکان آن تقریبا بی اطلاع است، استفاده میکند.
  - این شی دیگر ممکن است روی ماشین راه دور یا به صورت محلی قرار گیرد.
- دو روش برای برقراری ارتباط مولفه یک برنامه کاربردی دیگر، در اشیایی توزیع شده:
  - ارسال با مقدار / Passing By Value
  - ارسال با مرجع / Passing By Reference
- CORBA از روش ارسال با مرجع استفاده میکند.

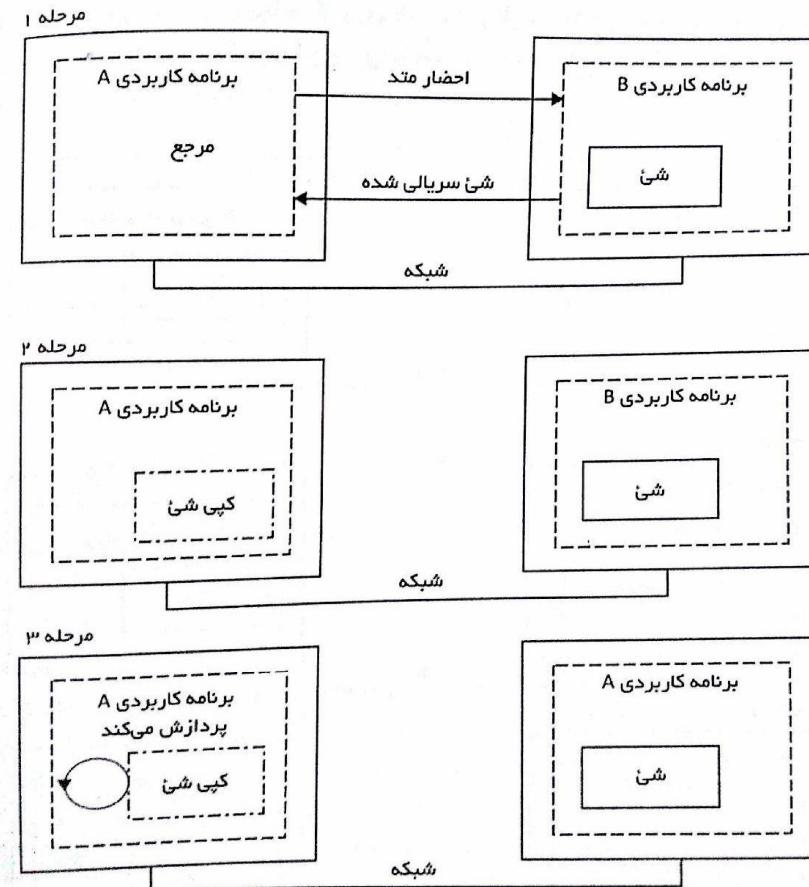
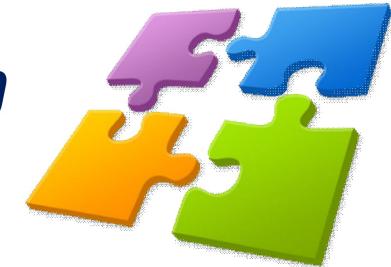
# CORBA مدل ... CORBA مدل شی



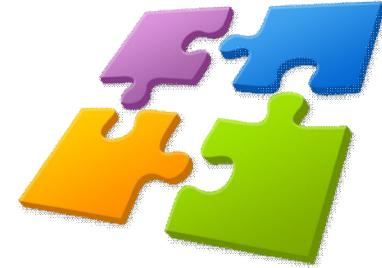
## ارسال با مقدار / Passing By Value

- رونوشتی از شی فراخوانی شده، ایجاد میشود که با فرآیندی به نام سریالیزه کردن به برنامه کلاینت ارسال میشود.
- برنامه کلاینت متدى روی رونوشت شیئی که از سرور دریافت کرده، احضار میکند.
- در این روش، عملیات انجام شده منجر به تغییر وضعیت شی در سرور نمی شود.
- به عنوان مثال:
  - برنامه A، متدى روی شی راه دور در برنامه B را احضار میکند.
  - برنامه B رونوشتی از شی فراخوانی شده را از طریق سریالیزه کردن ایجاد میکند و رونوشت ایجاد شده را به برنامه A ارسال میکند.
  - برنامه A رونوشتی از شی به صورت محلی ایجاد میکند و متدى روی این رونوشت فراخوانی میکند.

# CORBA مدل ... CORBA مدل شی



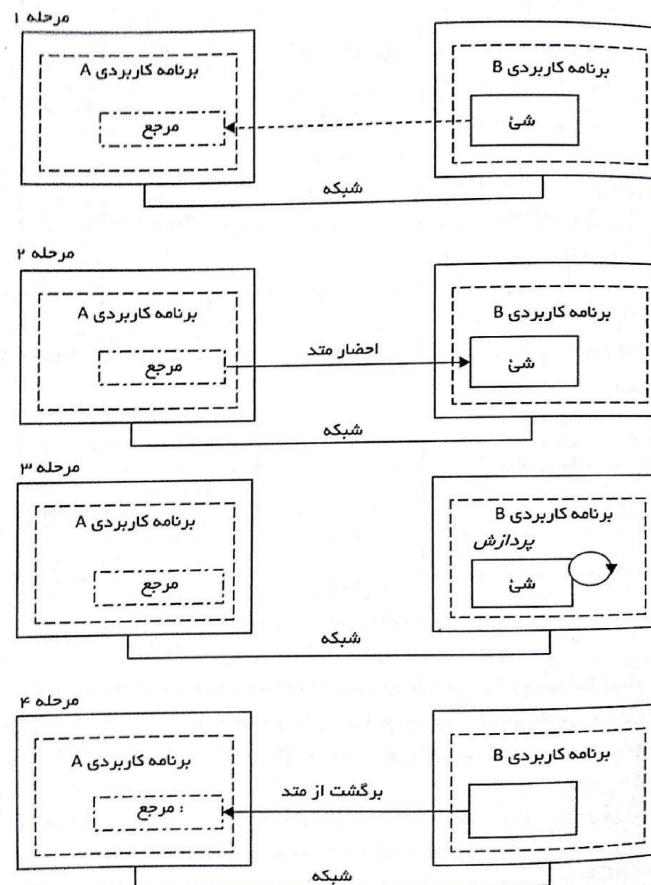
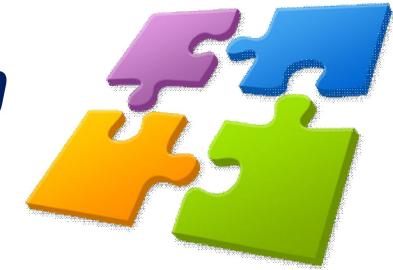
# CORBA مدل ... CORBA مدل شی



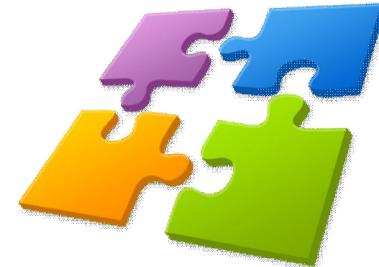
## ارسال با مرجع / Passing By Reference

- در این روش، شی در اختیار برنامه سرور قرار میگیرد و مرجع شی به کلاینت ارسال میشود.
- به عنوان مثال:
- ابتدا برنامه A مرجعی به شی راه دور در برنامه B دریافت میکند.
- سپس برنامه A از این مرجع برای فراخوانی متدهای دور، روی یک شی راه دور استفاده میکند.
- سپس برنامه B متدهای دور را پردازش میکند.
- سرانجام شی راه دور، پاسخ را به برنامه A ارسال میکند.

# CORBA مدل ... CORBA مدل شی

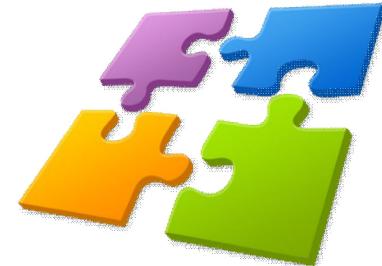


# CORBA مدل ... CORBA مدل شی



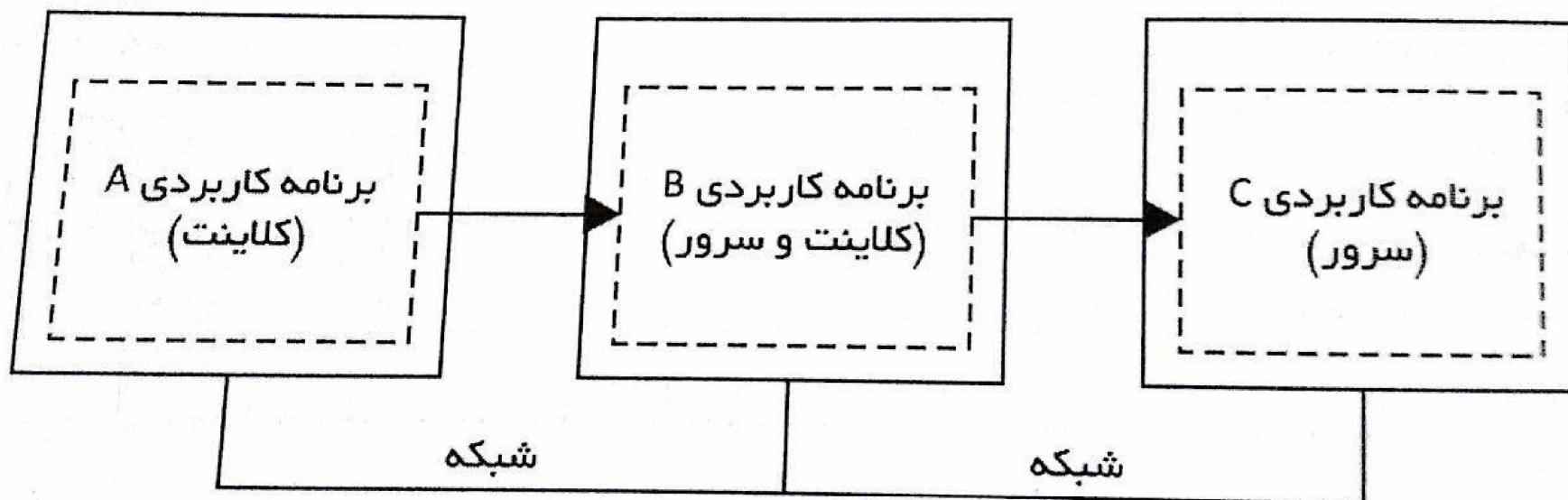
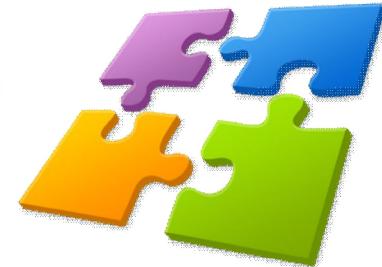
- آداتور شی:
- هدف اصلی آداتور شی، برقراری ارتباط بین پیاده سازی شی با رابط درخواست شی است.
- در CORBA سه نوع آداتور شی استفاده میشود که مهم ترین آنها آداتور شی ابتدایی / BOA است.
- این آداتور به اشیای CORBA، مجموعه رایجی از متدها برای دسترسی به توابع رابط درخواست شی ارائه میدهد.
- این توابع شامل فعال سازی، احراز هویت، و پایداری شی است.
- هر پیاده سازی از رابط درخواست شی مطابق با CROBA، باید یک آداتور شی ابتدایی را ارائه دهد.

# CORBA مدل نقش های کلاینت و سرور در CORBA

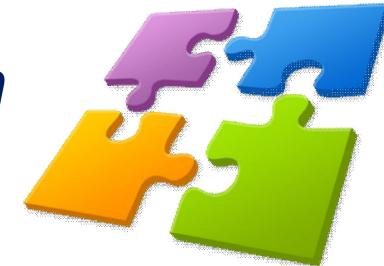


- CORBA مانند RPC مفهوم اصلی کلاینت و سرور رو حفظ میکند اما تمایز بین کلاینت و سرور در CORBA تا حدودی مبهم است. چون روابط آنها به صورت همتا به همتا است.
- سرور: هر برنامه کاربردی که شی را ایجاد میکند و به سایر برنامه های کاربردی امکان مشاهده این شی را میدهد.
- کلاینت: سایر برنامه های کاربردی که از سرویس های ارائه شده توسط سرور، استفاده می کند.
- یک برنامه کاربردی می تواند برای برخی از برنامه های کاربردی به عنوان کلاینت و برای برنامه های کاربردی دیگر به عنوان سرور عمل کند.

# مدل CORBA نقش های کلاینت و سرور در ... CORBA

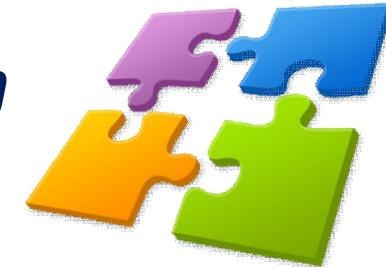


# CORBA مدل ارتباطات

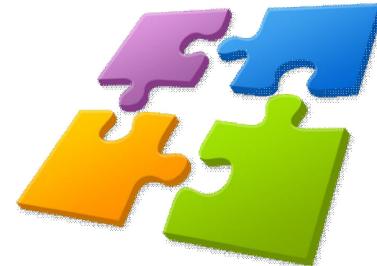


- در یک محیط توزیع شده، برنامه های کاربردی باید از طریق شبکه با هم ارتباط برقرار کنند.
- شبکه در بالای لایه فیزیکی (که میتواند به صورت سیمی یا بی سیم باشد)، از یک لایه شفاف تشکیل شده که مسئول انتقال بسته های داده از یک نقطه به نقطه دیگر است.
- متداولترین پروتکل انتقال، پروتکل TCP/IP و پس از آن پروتکل DCE است.
- CORBA از یک پروتکل استاندارد به نام IIOP، بین رابط های درخواست شی استفاده میکند.
- این پروتکل بر اساس پروتکل TCP/IP ساخته میشود و تعامل پذیری چندمحیطی صحیح را تضمین میکند.

# CORBA مدل سرویس های CORBA



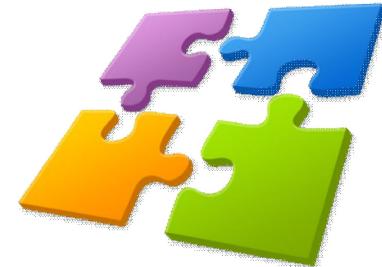
- **سرویس نامگذاری:**
  - باعث میشود اشیای CORBA با یک نام رجیستر و مکان یابی شوند.
  - به اشیای CORBA نام هایی داده میشود که اشیای دیگر با این نام ها میتوانند انها را جستجو کنند.
- **سرویس کنترل همزمانی:**
  - رابطی برای کنترل همزمانی در اشیایی به اشتراک گذاشته شده CORBA ارائه میدهد.
  - این کار با استفاده از چند نوع قفل مانند قفلهای خواننده-نویسنده، و قفل های هدف، انجام میشود.



# CORBA مدل سرویس های ... CORBA

- سرویس تراکنش:
- بخش جدایی ناپذیر هر برنامه کاربردی مهم
- تراکنش باید تجزیه ناپذیر باشد.
- به عنوان مثال برای هماهنگی انتقال بین دو حساب بانکی، عملیات به صورت تراکنش انجام میشود یعنی یا موجودی هر دو حساب به طور صحیح به روز می شود یا هیچ کدام.
- سرویس چرخه حیات:
- مسئول ایجاد، حذف، کپی و انتقال اشیای CORBA

# CORBA مدل ... CORBA سرویس های



- سرویس امنیت:
- امنیت در محیط یکپارچه سازی بسیار اهمیت دارد، زیرا هر برنامه کاربردی در دسترس همه برنامه های کاربردی دیگر سازمان است.
- اجزای سرویس امنیت:
- سرویس احراز هویت:
- برای تایید هویت مورد ادعای کاربر
- سرویس تعیین حق دسترسی:
- کنترل دسترسی به سرویسهای اشیای مختلف

# CORBA مدل سرویس های ... CORBA



- اجزای سرویس امنیت:...
- سرویس ممیزی امنیت:
- حفظ سوابق کلیه اعمال کاربر
- سرویس انکارناپذیری:
- امکاناتی شبیه امضای دیجیتال را ارائه میدهد، به این معنا که منبع ارسال داده، یا دریافت کننده داده، میتواند به طور غیرقابل انکار ثابت شود.



## مقایسه روش RPC و مدل شی توزیع شده

- مزایای مدل شی توزیع شده نسبت به روش **RPC**:

- با استفاده از رابط درخواست شی، مساله استفاده مجدد و ارتباط نقطه به نقطه، حل شد.
- با استفاده از زبان تعریف رابط، مساله وابستگی به زبان حل شد.

- مشکلات **RPC** که در مدل شی توزیع شده نیز حل نشد:

- مساله مقیاس پذیری
- این مساله به دلیل ماهیت ذاتی فراخوانی همگام است که باعث میشود کلاینت از انجام کار بیشتر باز بماند تا زمانی که سرور کارش را تمام کند و کنترل را به کلاینت برگرداند.