

COM چیست؟

COM سرنام های component object model می باشد. COM استانداردهایی برای برقراری ارتباط بین مولفه های مختلف تعریف می کند. هدف اصلی آن این است که به ما این امکان را بدهد تا بتوانیم برنامه هایی با استفاده از قطعات پیش ساخته یا مولفه (component) تولید کنیم. به عنوان مثال یک سیستم سفارش خرید را در نظر بگیرید. در این سیستم ممکن است برای ورود آیتم های مختلف از یک مولفه grid استفاده شده باشد که کاربر لیست آیتم ها را در آن وارد می کند. یک مولفه دیگر نیز داریم که به امکان جستجو در بین آیتم های مختلف را می دهد. همچنین ممکن است مولفه های دیگری نیز مانند مولفه محاسبه مالیات یک سفارش خرید را در سیستم داشته باشیم. حال سوال این است که چگونه می توان ارتباط بین این مولفه های مختلف را برقرار کرد. پاسخی که COM به این سوال می دهد این است که از طریق معرفی یکسری استاندارد. حال ببینیم که برای برقراری ارتباط بین این مولفه ها با چه مشکلاتی مواجه هستیم و به عبارت دیگر چه چیزهایی در برقراری ارتباط بین مولفه های مختلف دخالت دارند که باید برای آنها استاندارد تعریف کنیم.

اولین سوالی که مطرح می شود این است که چگونه به مولفه های مختلف دسترسی پیدا کنیم. برای جواب این سوال باید یک استاندارد تعریف شود تا برنامه نویسی بتواند پراحتی به مولفه های مختلف دسترسی پیدا کند و اشیاء مورد نظر خود را از آنها ایجاد کند. هر مولفه باید یک شناسه (identity) منحصر به فرد داشته باشد که آنرا از بقیه مولفه ها جدا کند. مولفه های مختلف ممکن است توسط شرکت های مختلفی ایجاد شده باشند و این شناسه منحصر بفرد باید به گونه ای باشد که هیچ دو مولفه ایجاد شده دارای شناسه یکسانی نباشند. دومین مشکل نحوه برقراری ارتباط بین اشیاء مختلف می باشد. بدون داشتن یک استاندارد مشخص در خصوص نحوه ارتباط بین اشیاء ممکن است وقت زیادی را لازم داشته باشیم تا بفهمیم که چگونه باید از اشیاء مختلف استفاده کرد و ارتباط بین آنها را برقرار کرد. سوم اینکه، برنامه نویسی نباید نگران این باشد که مولفه ای که قصد استفاده از آنرا دارد با چه زبان برنامه نویسی یا ابزاری تولید شده است. استقلال از زبان برنامه نویسی مشکل بزرگی است. هر چیزی که مربوط به یک شیء می شود باید بگونه ای تعریف شود که مستقل از زبان استفاده شده برای پیاده سازی آن و زبان استفاده شده برای بکارگرفتن آن شیء باشد، مثل تخصیص حافظه به آن، نام متدها، نوع پارامترها، نحوه صدا زدن متدها و غیره.

و بالاخره اینکه، باید بتوان نسخه های دیگری از مولفه های تولید شده را ساخت. باید ایجاد نسخه های بعدی از یک مولفه به گونه ای باشد که برنامه هایی که قبلا با استفاده از این مولفه نوشته شده اند با تغییر نسخه مولفه خود بتوانند باز هم به کار خود ادامه دهند و نیازی به تغییر برنامه خود نداشته باشند.

COM به تمام این سوال ها با معرفی یکسری استاندارد جواب می دهد. مدلی که COM برای برنامه نویسی ارائه می کند بسیار ساده و در عین حال قدرتمند است. برای شروع بحث در مورد مدل برنامه نویسی COM، ابتدا چند اصطلاح بکاربرده شده در این مدل را با هم بررسی می کنیم.

شیء (object)

شیء اصطلاحی است که در برنامه نویسی زیاد با آن سروکار داریم. همانند بسیاری از مدل های شیء گرا، در مدل COM نیز شیء یک نمونه زمان اجرا (run-time instance) از یک کلاس می باشد. اشیاء دارای سه خصیصه می باشند: شناسه، حالت یا وضعیت و رفتار. شناسه یک نام منحصر بفرد است که یک شیء را از بقیه اشیاء متمایز می کند. حالت یک شیء، داده هایی است که یک شیء دارا می باشد و رفتار یک شیء، مجموعه متدهایی است که از طریق آنها امکان تغییر داده های یک شیء یا انجام یک فعالیت توسط شیء امکان پذیر می شود. برای روشن شدن موضوع، اجازه بدهید که از یک مثال استفاده کنیم. یک شیء در زبان برنامه نویسی ++C یک نمونه زمان اجرا از یک کلاس می باشد. کلاس در ++C لیست متغیرها و متدهای اشیائی که از این کلاس ایجاد خواهند شد را مشخص می کند. وقتیکه یک شیء ایجاد می شود، یک فضای از حافظه برای نگهداری مقادیر متغیرهای آن شیء به آن نسبت داده می شود. آدرس آن حافظه نسبت داده شده به شیء، به عنوان شناسه شیء در نظر گرفته می شود. محتویات آن فضای حافظه، حالت شیء را مشخص می کند. رفتار شیء نیز توسط متدهای تعریف شده توسط کلاس مشخص می شود که این متدها در جای دیگری از حافظه نگهداری می شوند.

مدل های اشیاء استفاده شده توسط سایر زبان ها نیز تقریباً مشابه همین مدل استفاده شده توسط ++C می باشد، اما اشیاء COM اندکی متفاوت هستند. در COM، مفاهیم رابط عمومی (interface) و پیاده سازی شیء کاملاً از یکدیگر جدا هستند. برنامه ها می توانند با اشیاء تنها از طریق رابط آن و با استفاده از اشاره گر رابط ارتباط برقرار کنند. در اینجا چون از اشاره گر رابط استفاده می شود، برابری مهم نیست که حالت شیء در کجای حافظه قرار دارد یا به چه ترتیبی ذخیره شده است. فقط باید به گونه ای، شناسه شیء را با اشاره گر رابط مرتبط کرد. همانگونه که دیدیم، رابط نقش مهمی در مدل ارائه شده توسط COM برای برنامه نویسی دارد. در ادامه مفهوم رابط را با جزئیات بیشتری بررسی می کنیم.

رابط (interface)

یک رابط حاوی مجموعه ای از عملیات می باشد که به صورت منطقی به هم مربوط می باشند و یکسری رفتار را تعریف می کنند. در هنگام تعریف یک رابط، تنها توصیفی از مجموعه ای از عملیات ارائه می شود و هیچ گونه پیاده سازی در آن صورت نمی گیرد. به عبارت دیگر، تعاریفی که در یک رابط مطرح می شوند یک قرارداد بین استفاده کننده از مولفه و پیاده ساز مولفه وضع می کند. موارد زیر در خصوص یک رابط باید در نظر گرفته شود:

- یک رابط باید بوسیله یک شناسه منحصر بفرد تعریف شود.
- یک رابط باید در نهایت از یک رابط خاص به نام IUnknown مشتق شود.
- پس از اینکه یک رابط منتشر شد نباید تغییری در آن صورت بگیرد.

شناسه های COM

همانگونه که برای هر مولفه به یک شناسه منحصر بفرد نیاز داریم، هر رابط نیز باید یک شناسه منحصر بفرد داشته باشد. یکی از روش هایی که می توان بکار برد، استفاده از شناسه رشته ای می باشد. استفاده از شناسه رشته ای چند ایراد دارد. مهم ترین ایراد این است که نمی توان تضمینی داد که شناسه انتخاب شده منحصر بفرد باشد. در COM هرگاه که به یک شناسه منحصر بفرد نیاز داریم، از مفهومی با عنوان GUID که مخفف globally unique identifier می باشد، استفاده می شود. یک GUID یک عدد ۱۲۸ بیتی است. الگوریتمی که برای تولید GUID استفاده می شود، به صورت آماری تضمین می کند که اعداد تولید شده منحصر بفرد می باشند. بنابراین هر رابط با یک GUID شناخته می شود.

تعریف رابط

رابط های COM با استفاده از زبان تعریف رابط (Interface Definition Language) تعریف می شوند. IDL یک زبان مشابه ++C می باشد ولی از آنجاییکه اکثر برنامه نویسان با آن آشنایی زیادی ندارند و از طرفی برخی از نوع های استفاده شده در آن توسط برخی از زبان های برنامه نویسی پشتیبانی نمی شود، بنابراین اکثر ابزارهای برنامه نویسی که از COM پشتیبانی می کنند، این امکان را برای پیاده ساز فراهم می کنند تا با استفاده از نحو همان زبان، تعریف رابط را انجام دهد و تولید IDL و کمپایل آنرا از دید پیاده ساز مخفی می کند. قبل از ادامه بحث، اجازه بدهید از یک مثال استفاده کنیم تا ساختار کلی رابط تولید شده را بهتر متوجه شویم. این مثال تعریف یک رابط در دلفی می باشد:

```
unit ISample  
interface
```

```

const
;'{ISampleGUID : TGUID = '{4A9B897D-D302-43DC-BCF9-74FE9B56DC6D
type
ISampleInterface = interface
['{A9B897D-D302-43DC-BCF9-74FE9B56DC6D4}']
;procedure ShowForm; stdcall
; end
implementation
.end

```

همانگونه که دیده می شود، این رابط حاوی یک GUID می باشد که باعث می شود این رابط از سایر رابط ها تمیز داده شود. رابط از نوع Interface تعریف می شود.

در COM همانگونه که گفته شد باید تمام رابط ها از رابط پایه IUnknown مشتق شده باشند و دلفی خود اینکار را در هنگام تبدیل این رابط به رابطی مطابق استاندارد تعریف شده توسط COM اینکار را انجام خواهد داد. متدها در بدنه رابط تعریف می شوند.

علاوه بر متدهای تعریف شده توسط کاربر، که در اینجا یک متد را تعریف کرده است، سه متد پایه دیگر نیز به رابط اضافه خواهند شد. در حقیقت این متدها در رابط پایه IUnknown تعریف شده اند و از طریق اشتقاق انجام شده، این رابط نیز آنها را دارا خواهد بود. این سه متد عبارتند از : QueryInterface، AddRef و Release.

هر مولفه شامل یکسری کلاس می باشد و یک رابط نیز دارد که از طریق این رابط امکان دسترسی به کلاس های موجود در آن و ساخت یک نمونه از آنها ممکن می شود. ارتباط بین مولفه های مختلف تنها از طریق رابط آنها انجام می شود.

مرجع: Mary Kirtland, Designing Component Based Applications, Microsoft Press, 1998

نوشته شده توسط حسن شجاعی در شنبه ۲۲ بهمن ۱۳۸۴ |