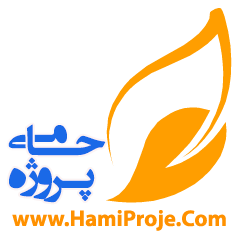
**تجزیه و تحلیل صرف غذا در رستوران با UML**

****

UML چیست؟

UML شامل تعدادي عنصر گرافيكي است كه از تركيب آنها نمودارهاي UML شكل مي گيرند. هدف استفاده از نمودارهاي مختلف در UML ، ارائه ديدگاه هاي گوناگون از سيستم است. همانطور كه مهندسين عمران جهت ساختن يك ساختمان پلانهاي مختلفي از ساختمان تهيه مي كنند ، ما با استفاده از نمودارهاي UML نماهاي مختلفي از نرم افزار مورد نظر را تهيه مي كنيم.  
  
نكته اي كه بايد حتما به آن توجه كنيد اين است كه : مدل UML آنچه كه يك سيستم بايد انجام دهد را توضيح مي دهد، ولي چيزي درباره نحوه پياده سازي سيستم نمي گويد.  
  
با توجه به رشد نرم افزارهاي پشتيباني كننده UML امروزه با استفاده از نرم افزارهايي مانند Visio ، Enterprise Architecture و Rational Rose شما مي توانيد بعد از كشيدن نمودارهاي UML مستقيما نمودارهاي خود را به بانك اطلاعاتي و كد تبديل كنيد (البته اين نرم افزارها ساختار كد شما را برايتان توليد مي كنند!) اين نرم افزارها همچنين كد برنامه شما را گرفته و نمودارهاي UML برنامه را توليد مي كنند. پس از آشنايي با مفاهيم شيء گرايي، (توضیحات بیشتر در سایر مقالات سایت میکرو رایانه) در اينجا زبان مدلسازي UML را معرفي کرده و خواهيم ديد چگونه اين زبان مفاهيم شيء گرايي را پشتيباني مي كند.  
  
زبان مدل سازي يكنواخت ( Unified Modeling Language ) یا UML يك زبان مدلسازي است كه براي تحليل و طراحي سيستم هاي شی گرا به كار مي‌رود. UML اولين بار توسط شركت Rational ارائه شد و پس از آن از طرف بسياري از شركت هاي كامپيوتري و مجامع صنعتي و نرم افزاري دنيا مورد حمايت قرار گرفت؛ به طوري كه تنها پس از يك سال، توسط گروه Object Management Group، به عنوان زبان مدلسازي استاندارد پذيرفته شد. UML توانايي ها و خصوصيات بارز فراواني دارد كه مي‌تواند به طور گسترده‌اي در توليد نرم‌افزار استفاده گردد.

****

**تاريخچة UML**   
ديدگاه شي گرايي (Object Oriented) از اواسط دهه 1970 تا اواخر دهه 1980 در حال مطرح شدن بود. در اين دوران تلاش هاي زيادي براي ايجاد روش هاي تحليل و طراحي شي گرا صورت پذيرفت. در نتيجه اين تلاش ها بود كه در طول 5 سال يعني 1989 تا 1994، تعداد متدولوژي هاي شي گرا از كمتر از 10 متدولوژي به بيش از 50 متدولوژي رسيد. تكثر متدولوژي ها و زبانهاي شي گرايي و رقابت بين اينها به حدي بود كه اين دوران به عنوان دوران جنگ متدولوژيها لقب گرفت.  
  
از جمله متدولوژي هاي پركاربرد آن زمان مي توان ازBooch، OOSE، OMT، Fusion، Coad-Yourdan، Shlayer-Mellor و غيره نام برد. فراواني و اشباع متدولوژيها و روشهاي شي گرايي و نيز نبودن يك زبان مدلسازي استاندارد، باعث مشكلات فراواني شده بود. از يك طرف كاربران از متدولوژيهاي موجود خسته شده بودند، زيرا مجبور بودند از ميان روشهاي مختلف شبيه به هم كه تفاوت كمي در قدرت و قابليت داشتند يكي را انتخاب كنند. بسياري از اين روشها، مفاهيم مشترك شي گرايي را در قالب هاي مختلف بيان مي کردند كه اين واگرايي و نبودن توافق ميان اين زبانها، كاربران تازه كار را از دنياي شي گرايي زده مي‌ کرد و آنها را از اين حيطه دور مي‌ساخت. عدم وجود يك زبان استاندارد، براي فروشندگان محصولات نرم افزاري نيز مشكلات زيادي ايجاد كرده بود.  
  
اولين تلاشهاي استانداردسازي از اكتبر 1994 آغاز شد، زماني كه آقاي Rumbaurgh صاحب متدولوژي OMT به آقاي Booch در شركت Rational پيوست و اين دو با تركيب متدولوژيهاي خود، اولين محصول تركيبي خود به نام `روش يكنواخت` را ارائه دادند. در سال 1995 بود كه با اضافه شدن آقاي Jacobson به اين دو، روش يكنواخت ارائه شده با روش OOSE نيز تركيب شد و اين خود سبب ارائه UML نسخة 0.9 در سال 1996 گرديد. سپس اين محصول به شركتهاي مختلفي در سراسر جهان به صورت رايگان ارائه شد و استقبال شديد شركت ها از اين محصول و تبليغات گسترده شركت Rational، سبب آن شد كه گروه OMG، نسخة 1.0 UML را به عنوان زبان مدلسازي استاندارد خود بپذيرد. تلاشهاي تكميلي UML استاندارد ادامه پيدا كرد و نسخة 1.1 آن در سال 1997 و نسخه 1.3 آن در سال 1999 ارائه گرديد.   
  
UML يا زبان مدلسازي يكنواخت، زباني است براي مشخص كردن (Specify)، مصورسازي (Visualize)، ساخت (Construction) و مستندسازي (Documenting) سيستمهاي نرم افزاري و غير نرم افزاري و نيز براي مدلسازي سيستمهاي تجاري.

**چرا مدل و مدلسازي ؟**   
ايجاد يك مدل براي سيستمهاي نرم افزاري قبل از ساخت يا بازساخت آن، به اندازه داشتن نقشه براي ساختن يك ساختمان ضروري و حياتي است. بسياري از شاخه هاي مهندسي، توصيف چگونگي محصولاتي كه بايد ساخته شوند را ترسيم مي كنند و همچنين دقت زيادي مي كنند كه محصولاتشان طبق اين مدلها و توصيفها ساخته شوند. مدلهاي خوب و دقيق در برقراري يك ارتباط كامل بين افراد پروژه، نقش زيادي مي توانند داشته باشند. شايد علت مدل كردن سيستمهاي پيچيده اين باشد كه تمامي آن را نمي توان يك باره مجسم كرد، بنابراين براي فهم كامل سيستم و يافتن و نمايش ارتباط بين قسمتهاي مختلف آن، به مدلسازي مي‌پردازيم. UML زباني است براي مدلسازي يا ايجاد نقشه توليد نرم افزار. به عبارت ديگر، يك زبان، با ارائه يك فرهنگ لغات و يك مجموعه قواعد، امكان مي دهد كه با تركيب كلمات اين فرهنگ لغات و ساختن جملات، با يكديگر ارتباط برقرار كنيم. يك زبان مدلسازي، زباني است كه فرهنگ لغات و قواعد آن بر نمايش فيزيكي و مفهومي آن سيستم متمركزند. براي سيستمهاي نرم افزاري نياز به يك زبان مدلسازي داريم كه بتواند ديدهاي مختلف معماري سيستم را در طول چرخه توليد آن، مدل كند.  
فرهنگ واژگان و قواعد زباني مثل UML به شما مي گويند كه چگونه يك مدل را بسازيد و يا چگونه يك مدل را بخوانيد. اما به شما نمي گويند كه در چه زماني، چه مدلي را ايجاد كنيد. يعني UML فقط يك زبان نمادگذاري (Notation) است نه يك متدولوژي. (توضیحات بیشتر در سایر مقالات سایت میکرو رایانه) يك زبان نمادگذاري شامل نحوه ايجاد و نحوه خواندن يك مدل مي باشد، اما يك متدولوژي بيان مي كند كه چه محصولاتي بايد در چه زماني توليد شوند و چه كارهايي با چه ترتيبي توسط چه كساني، با چه هزينه‌اي، در چه مدتي و با چه ريسكي انجام شوند.

**نمودارهای UML**  
 در این بخش به معرفی نمودارهای UML می‌پردازیم:



**نمودار موردکاربرد (Usecase Diagram)** این نمودار، تعامل کاربران خارجی و سیستم را مدل می کند و از جهاتی شبیه نمودار سطح صفر DFD می باشد که جنبه های رفتاری سیستم را نمایش می دهد. این نمودار نقطه‌ ورودی برای تمامی نمودارهای دیگری است که به تشریح نیازمندیها و معماری و پیاده سازی سیستم می پردازند.

**نمودارهای تعامل (Interaction Diagram)**  
 این نمودارها، بیان کننده تعامل هستند که شامل اشیاء مختلف است و نیز روابط بین آنها و همچنین پیغام هایی که بین آنها رد و بدل می شود.  
 این نمودارها جنبه های پویای یک سیستم را مدل می کنند و خود بر دو نوعند:

* نمودار توالی (Sequence Diagram) که ترتیب زمانی تعامل ها را نشان می دهد.
* نمودار همکاری (Collaboration Diagram) که تاکید بر نمایش ساختاری تعامل ها دارد.

**نمودار فعالیت (Activity Diagram)**  
 این نمودار، نوع خاصی است از نمودار حالت، که انتقال جریان از یک فعالیت به فعالیت دیگر را نمایش می دهد. این نمودار جنبه های پویای یک سیستم را نمایش می دهد. در واقع حالات این نمودار، گام های ترتیبی انجام یک عمل را نمایش می دهند.

**نمودارحالت (Statechart Diagram)**  
 این نمودار، بیان کننده جنبه های رفتاری سیستم می باشد و در واقع توصیف رسمی یک کلاس بوده که شامل حالات، انتقال بین حالات، رخدادها و فعالیت ها می‌باشد. از این نمودارها برای نمایش دادن چرخه حیات اشیاء یک کلاس خاص نیز می توان استفاده کرد.

**نمودار اشیاء (Object Diagram)**   
 این نمودار، اشیاء سیستم و روابط بین آنها را نمایش می دهد. در واقع یک تصویر لحظه‌ای از نمودار کلاس می باشد.

**نمودار کلاس (Class Diagram)**  
 این نمودار، کلاس ها، واسط ها و همکاری و روابط بین آنها را نمایش می دهد. و نمودار اصلی و مرکزی UML می‌باشد. که بیان کننده ساختار ایستای سیستم نرم افزاری می باشد.

**نمودار اجزاء(Component Diagram)**  
 از جمله نمودارهای پیاده سازی می‌باشد و سازمان دهی و روابط بین مجموعه‌ای از اجزاء را نمایش می دهد. این نمودار، جنبه های ایستای پیاده سازی یک سیستم را مدل می کند.

**نمودار استقرار(Deployment Diagram)**  
 پیکربندی گره های پردازشی زمان اجرا را نمایش می دهد. که برای مدل کردن جنبه های ایستای به کار‌گماری یک معماری بکار می رود. همچنین نمایش دهنده اجزای استفاده شده زمان اجرا مثل کتابخانه های DLL، فایل‌های اجرایی، کدهای مبدا و روابط بین آنها می باشد.  
البته این نمودارها تمام نمودارهای UML نیستند بلکه بسته به نیاز و با کمک ابزارهای Case می توان نمودارهای دیگری نیز تعریف و استفاده کرد.

**مقدمه**

مهندسی نرم افزار راهی برای مدیریت زمان و منابع با استفاده از نرم افزارهای مدیریتی و کمکی می باشد و همچنین ما می توانیم با تحلیل درست یک سیستم در وقت؛ هزینه و ... صرفه جویی داشته باشیم. در این پروژه ما می خواهیم سیستم رستوران را با استفاده از نمودارهای usecase diagram, class diagram, sequence diagram, activity diagram, swimlane که همگی این نمودار ها در نرم افزار rational rose رسم می شود؛ تحلیل کنیم. که در نمودارها شیوه خرید غذا توسط مشتریان و سرو غذا و تحویل آنها به مشتریان توسط کارمندان رستوران مورد تجزیه و تحلیل قرار گرفته است.

**فهرست**

نمودار use case ................................................................................................................................. 5

نمودار sequence .............................................................................................................................. 6

نمودار activity .................................................................................................................................... 8

نمودار swim lane ........................................................................................................................... 10

نمودار class ........................................................................................................................................ 12

نمودار Use case diagram:



نمودار sequence:

این نمودار برای آماده سازی غذا برای داخل رستوران می باشد.

نمودار sequence:

این نمودار برای تحویل غذا برای مشتریان ثابت رستوران در منازلشان با استفاده از پیک می باشد.



نمودار Activity:



نمودار Activity:

نمودار swim lane:



نمودار swim lane:



نمودار class:

