بسم الله الرحمن الرحیم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

معرفی UML

مدرس : مهندس افشین رفوآ

دوره آموزش UML

معرفی UML

UML یک زبان استاندارد برای تعریف، نمایش گرافیکی، ساختن و مستندسازی اجزای یک سیستم نرم افزاری کمک گرفت.

این زبان در سال 1997 هنگامی که پیش نویس مشخصات آن به گروه مدیریت شی (OMG) ارائه شد، با ویرایش 1 پا به عرصه گذاشت.

OMG سعی بر هر چه بهتر کردن این زبان دارد. در زیر توضیحاتی را درباره ی این زبان مشاهده می کنید:

1. UML سرواژه ی Unified Modeling Language می باشد.
2. UML از دیگر زبان های رایج برنامه نویسی مانند C++، Java و COBOL متفاوت است.
3. UML یک زبان تصویری، نمایشی است که از آن جهت مدل سازی و ساخت برنامه ی کار نرم افزار استفاده می شود.

بنابراین UML را می توان به عنوان یک زبان دیداری مدل سازی همه منظور تلقی کرد که توسط آن سیستم نرم افزاری نمایش، تعریف، ساخته و مستندسازی می شود. اگرچه UML بیشتر جهت مدل سازی سیستم ها نرم افزاری بکار می رود، اما می توان از آن در زمینه های دیگر مانند مدل سازی جریان پردازش در یک واحد تولید بهره گرفت.

UML به خودی خود یک زبان برنامه نویسی نیست اما ابزاری است که با استفاده از نمودارهای آن می توان به زبان های مختلف کد نوشت. UML یک رابطه ی مستقیم با تجزیه و تحلیل، طراحی شی گرا دارد. پس از کمی متعارف سازی، UML به ی استاندارد OMG تبدیل شده است.

اهداف UML

مفایم شی گرا بسیار قبل تر از UML معرفی شدند. در آن زمان هیچ روشی برای سازمان دهی و تحکیم برنامه نویسی شی گرا وجود نداشت. UML برای این منظور پا به عرصه گذاشت.

UML برای اهداف مختلفی تعبیه و عرضه شد، اما مهمترین آن تعریف یک زبان همه منظوره ی مدل سازی بود که تمامی طراحانی که مدل سازی انجام می دهند بتوانند از آن استفاده کنند.

نمودارهای UML تنها توسط برنامه نویسان یا توسعه دهندگان استفاده نمی شود، بلکه تمامی طراحان و همچنین مردم عادی که علاقه مند به مدل سازی و فهم سیستم هستند می توانند از آن استفاده کنند. این سیستم می تواند یک سیستم نرم افزاری یا غیر نرم افزاری باشد.

UML را نمی‌توان به عنوان یک روش تولید نرم‌افزار کامل دانست. این زبان شامل فرایند مرحله به مرحله تولید نرم‌افزار نیست، بلکه UML زبانی است که تقریبا تمام شیوه‌های تولید نرم‌افزار از آن استفاده می‌کنند.

در پایان باید گفت که UML یک مکانیزم modeling ساده برای مدل سازی تمامی سیستم های کاربردی در محیط پیچیده ی امروز محسوب می شود.

مدل یا الگوی ذهنی از UML (Conceptual model)

برای اینکه بفهمیم مدل ذهنی از UML چی هست، ابتدا می بایست تعریف روشن و واضحی از آن ارائه کنیم و همچنین علت نیاز به آن را مورد بررسی قرار دهیم.

1. یک conceptual model را می توان یک مدل در نظر گرفت که از مفاهیم و رابطه ی میان آن مفایهم تشکیل شده است.

2. مدل ذهنی اولین گامی است که پیش از ترسیم نمودار UML بایستی به آن پرداخت. این مدل در درک موجودیت ها و نحوه ی تعامل بین آن ها در جهان واقع کمک می کند.

از آنجایی که UML سیستم های زمان واقعی (real time system) تعریف می کند، بایستی ابتدا یک مدل ذهنی از آن ترسیم/ایجاد کرده و از آن جا باقی مراحل را دنبال کرد. لازم است برای درک کامل مدل ذهنی، سه المان اصلی زیر را فرابگیرید:

1. بخش های اصلی و تشکیل دهنده ی UML

2. قوانین لازم برای ربط دادن این بخش های تشکیل دهنده

3. مکانیزم های رایج UML

مفاهیم شی گرا

می توان گفت که UML دنباله روی طراحی، تجزیه و تحلیل شی گرا می باشد.

یکی شی در واقع دربردارنده ی داده ها و متدهایی است که آن داده ها را کنترل می کنند. داده های ذخیره شده در شی، در واقع وضعیت آن را نمایش می دهد. کلاس یک شی تعریف می کند و سلسه مراتبی که بر اساس آن سیستم واقعی مدل سازی می شود را شکل می دهد. سلسله مراتب به صورت وراثت نشان داده می شود و کلاس ها بر اساس نیاز به هم ربط داده می شوند.

اشیا موجودیت های واقعی اطراف ما هستند. مفاهیم پایه ای مانند abstraction، کپسوله سازی (encapsulation)، وراثت (inheritance) و چند ریختی (polymorphism) را می توان به وسیله ی UML نمایش داد.

از این رو می توان گفت که UML چنان قدرتمند است که می تواند تمامی مفاهیم موجود در تجزیه، تحلیل و طراحی شی گرا را نمایش دهد. نمودارهای UML تنها نشانگر مفاهیم پایه ای شی گرا هستند.از این رو می بایست پیش از فراگیری UML، با مفاهیم شی گرا آشنایی پیدا کرد.

در زیر برخی از مفاهیم اساسی دنیای شی گرا شرح داده شده است:

1. شی: شی نشانگر یک موجودیت و عضو اساسی می باشد.

2. کلاس: کلاس طرحی از یک شی است.

3. انتزاع (abstraction): نشانگر رفتار یک موجودیت در دنیای واقعی است.

4. کپسوله سازی (encapsulation): مکانیزم متصل کردن داده ها به یکدیگر و پنهان سازی آن ها از دنیای خارج است.

5. وراثت (inheritance): مکانیزم ایجاد کلاس های جدید از یک کلاس موجود می باشد.

6. چندریختی (polymorphism): مفهوم چندریختی ویژگی است که به رابطها امکان می‌دهد تا برای گروهی از عملیات‌ها مورد استفاده قرار گیرند.

تجزیه و تحلیل oop

تجزیه و تحلیل شی گرا را می توان یک نوع تحقیق و بررسی یا به طور دقیق تر بررسی دقیق اشیا دانست. طراحی یعنی همکاری و رابطه ی بین اشیا شناسایی شود.

از این رو درک تحلیل و طراحی شی گرا از اهمیت بالایی برخوردار است. لازم است توجه داشته باشید که مهم ترین هدف تجزیه و تحلیل شی گرا شناسایی اشیایی است که می بایست طراحی شود. گفتنی است که این تجزیه و تحلیل برای سیستم موجود نیز انجام می شود. حال یک تجزیه و تحلیل موثر صرفا زمانی امکان پذیر می باشد که ما بتوانیم اشیا را شناسایی کنیم. پس از شناسایی اشیا، رابطه ی بین آن ها را شناسایی کرده و در نهایت design را تولید می کنیم.

می توان هدف تجزیه و تحلیل شی گرا را در موارد زیر خلاصه نمود:

1. شناسایی اشیا سیستم.
2. شناسایی رابطه ی میان آن ها.
3. ایجاد یک طرح که بتوان به وسیله ی زبان های شی گرا به فایل های اجرایی تبدیل نمود.

در کل سه گام وجود دارد که طی آن مفاهیم شی گرا اعمال و پیاده سازی می شوند. این گام ها به ترتیب زیر می باشند:

OO Analysis --> OO Design --> OO implementation using OO languages

پیاده سازی شی گرا به وسیله ی زبان های شی گرای← طراحی شی گرای ←تجزیه و تحلیل شی گرا

حال سه گام بالا را به تفصیل شرح می دهیم:

1. در این مرحله (تحلیل شی گرا) هدف اصلی که دنبال می شود، شناسایی اشیا و توصیف آن ها به شیوه ی صحیح است. اگر این اشیا به صورت کارآمد شناسایی شوند، گام بعدی طراحی آسان می باشد. اشیا می بایست همراه با مسئولیت های آن ها شناسایی شوند. مسئولیت ها درواقع کارها یا وظایفی هستند که شی انجام می دهد. هر شی ای مسئولیت های خاص خود را داشته و وظایف مربوط به خود را انجام می دهد. هنگامی که این مسئولیت ها، هماهنگ شده و رابطه ی بین آن ها شکل می گیرد، مقصود اصلی سیستم برآورده می شود.

2. مرحله ی دوم طراحی یا design شی گرا می باشد. در این مرحله تاکیید بر روی نیازها و برآورده کردن آن نیازهاست. همچنین در این گام اشیا با توجه به رابطه ی قبلا تعیین شده همکاری می کنند. با تکمیل رابطه ی بین آن ها، طراحی نیز کامل می شود.

3. مرحله ی سوم پیاده سازی شی گرا است. در این مرحله طراحی به واسطه ی زبان های شی گرا نظیر Java، C و C++ پیاده سازی می شود.

نقش UML در طراحی شی گرا (OO design)

همان طور که پیش تر گفته شد، UML یک زبان مدل سازی سیستم های نرم افزار و غیر نرم افزاری می باشد. اگرچه UML برای سیستم های غیر نرم افزاری نیز بکار می رود، تاکیید آن (کاربرد اصلی آن) بر روی مدل سازی نرم افزارهای کاربردی شی گرا می باشد. بیشتر نمودارهای UML که تاکنون درباره ی آن ها بحث شد، به منظور مدل سازی مفاهیم و جنبه های مختلف همچون static یا dynamic بکار می رفتند. حال می گوییم جنبه هر چه می خواهد باشد، اجزا چیزی به جز اشیا نیستند.

اگر نمودار کلاس، نمودار شی، نمودار همکاری (collaboration diagram) و نمودارهای تعامل را با دقت مورد بررسی قرار دهیم، متوجه می شویم که تمامی آن ها بر اساس اشیا طراحی می شوند.

درک رابطه ی بین طراحی شی گرا و UML بسیار حائز اهمیت می باشد. طراحی شی گرا با توجه به نیاز به نمودارهای UML تبدیل می شوند. پیش از درک کامل UML، می بایست بر مفاهیم شی گرا اشراف پیدا کرد. پس از با

موفقیت پشت سر گذاشتن تحلیل و طراحی شی گرا، گام بعدی بسیار آسان می باشد. خروجی تحلیل و طراحی شی گرا ورودی نمودارهای UML می باشد.