

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

بخش های اساسی و تشکیل دهنده ی UML

مدرس : مهندس افشین رفوآ

## دوره آموزش UML

### بخش های اساسی و تشکیل دهنده ی UML

از آنجایی که UML سیستم های بی وقفه و زمان واقعی (real time) تعریف می کند، ایجاد یک مدل ذهنی (conceptual model) پیش از هر کاری از اهمیت بالایی برخوردار است. جهت فراگیری مدل ذهنی UML، لازم است بر المان های اساسی آن اشراف داشته باشید:

1. اجزا و المان های اصلی تشکیل دهنده ی UML

2. قواعد به هم متصل کردن و ربط دادن این اجزا

3. مکانیزم و سازوکارهای معمول UML

این مبحث تمامی اجزای اصلی زبان مدل سازی UML را شرح می دهد. اجزای اصلی این زبان عبارتند از:

1. اشیا (Things)

2. رابطه ها (Relationships)

3. نمودارها (Diagrams)

### 1 اشیا (Things)

اشیا (thing) مهمترین اجزا UML هستند. اشیا می توانند:

1. ساختاری (structural)

2. رفتاری (behavioral)

3. grouping (مربوط به گروه بندی)

4. annotational (مربوط به نشان گذاری و حاشیه نویسی)

باشند.

## Structural things

بخش ایستا (static) مربوط به مدل را تعریف می کند. Things نشانگر المان های فیزیکی و ذهنی هستند. در زیر توضیحات مختصری درباره ی اشیا ساختاری/structural things مشاهده می کنید:

## Class (کلاس)

کلاس نشانگر مجموعه ای از اشیاست که دارای مسئولیت های مشابه هستند.



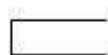
## Interface (رابط)

رابط یا interface یک سری عملیات تعریف می کند که مسئولیت هر کلاس را مشخص می نماید.



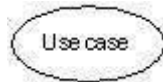
## Collaboration (همکاری)

Collaboration همکاری یا تعامل بین المان ها را تعریف می کند.



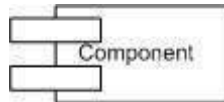
## Use case (مورد کاربرد)

Use case نشانگر یک سری عملیات است که توسط سیستم برای نیل به هدف خاصی اجرا می شود.



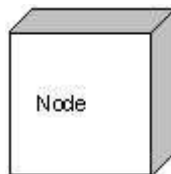
## Component (مولفه)

مولفه نشانگر یا تعریف کننده ی بخش فیزیکی سیستم می باشد.



## Node (گره)

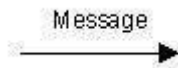
**Node** را می توان یک عنصر فیزیکی در نظر گرفت که در زمان اجرا بوجود می آید.



## Behavioral things

**Behavioral thing** متشکل هست از بخش های پویا (**dynamic part**) مدل های **UML Behavioral thing** ها خود به مولفه های زیر تقسیم می شوند:

1. **interaction** (تعامل): **Interaction** یک رفتار یا **behavior** است که از تعدادی پیغام تشکیل می شود. این پیغام ها در راستای نیل به هدف خاصی بین المان ها رد و بدل می شوند.



2. **state machine** (ماشین وضعیت): زمانی بکار می آید که وضعیت یک شی در چرخه ی حیاتش (**life cycle**) از اهمیت خاصی برخوردار باشد. **state machine** دنباله ای از وضعیت های مختلف است که یک شی در پاسخ به رخدادهای (**event**) متفاوت تجربه می کند. رخدادها عوامل خارجی هستند که تغییر در وضعیت (شی) از آن ها ناشی می شود.



## Grouping thing (شی گروه بندی)

شی گروه بندی عبارت است از یک مکانیزم که وظیفه ی گروه بندی المان ها یک مدل UML را بر عهده دارد. تنها یک شی گروه بندی وجود دارد و آن Package می باشد.

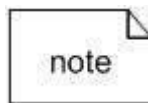
Package : تنها وسیله ای که می توان با استفاده از آن Behavioral thing و Structural thing را گروه بندی کرده و به صورت یک بسته درآورد.



## Annotational things (شی حاشیه نویسی)

یک سازوکار برای ضبط description (توصیفات)، comment (توضیحات)، remark (نظرات) المان های مدل UML می باشد. Note تنها ابزار حاشیه نویسی می باشد.

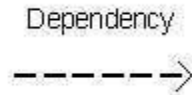
Note : یک شی یا ابزار است که امکان نمایش (render) توضیحات، محدودیت ها (constraint) مربوط به المان UML را فراهم می آورد.



## Relationship (2) (رابطه)

Relationship یکی دیگر از اجزای اساسی زبان مدل سازی UML می باشد. Relationship نشان می دهد چگونه المان ها با یکدیگر رابطه دارند و به هم متصل هستند، این رابطه نیز قابلیت برنامه را تعریف می کند. در کل چهار نوع relationship وجود دارد که به شرح زیر می باشد.

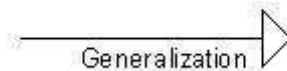
**Dependency** (رابطه ی وابستگی): **Dependency** یک رابطه بین دو شی است که در آن دو شی به هم وابسته هستند و تغییر در یکی سبب تغییر در دیگری می شود.



**Association** (رابطه ی انجمنی): **association** در اصل یک سری پیوند است که عناصر یک مدل **UML** را به هم متصل می کند. این نوع رابطه همچنین بیان می کند چه تعداد شی در آن رابطه شرکت دارند.



**Generalization** (رابطه وراثت): **Generalization** یک نوع رابطه هست که یک المان اختصاصی را به یک المان عمومی ربط می دهد (بیانگر رابطه ی جز به کل می باشد). در واقع رابطه ی وراثت را در جهان اشیا توصیف می کند.



**Realization** (رابطه ی تحقق): **Realization** را می توان یک رابطه تعریف کرد که در آن دو المان به هم وصل هستند. یک رابطه **Realization** از عنصر مبدا (عنصر محقق سازی) به عنصر مقصد (عنصر خصوصیت) نشان می دهد که عنصر مبدا حداقل همه عملیات عنصر مقصد را پشتیبانی می کند. (عنصر مبدا می تواند صفات یا پیوندهای عنصر مقصد را پشتیبانی کند). این رابطه را می توان در **interface** سراغ گرفت.

### 3) نمودارهای UML

نمودارهای **UML** خروجی یا همان نتیجه ی نهایی کل این مبحث محسوب می شود. تمامی المان ها، رابطه ها باهم ترکیب شده و یک نمودار کامل **UML** و نمودار هم در نتیجه سیستم را تشکیل می دهد.

آن نموداری که به عنوان خروجی مشاهده می کنید و تمامی عناصر ذکر شده در تکمیل آن دخیل بوده و شرکت دارند، مقصود نهایی بوده و مهم ترین بخش پروسه می باشد.

**UML** در مجموع 9 نمودار به عنوان خروجی عرضه می کند. در مباحث بعدی جزئیات مربوط به این نه نمودار را به تفصیل شرح خواهیم داد.

1. نمودار کلاس (**class**)

2. نمودار شی (**object**)

3. نمودار مورد کاربرد (**use case**)

4. نمودار توالی (**sequence**)

5. نمودار همکاری (**collaboration**)

6. نمودار فعالیت (**activity**)

7. نمودار وضعیت (**statechart**)

8. نمودار **deployment** (اسقرار)

9. نمودار اجزا (**component**)