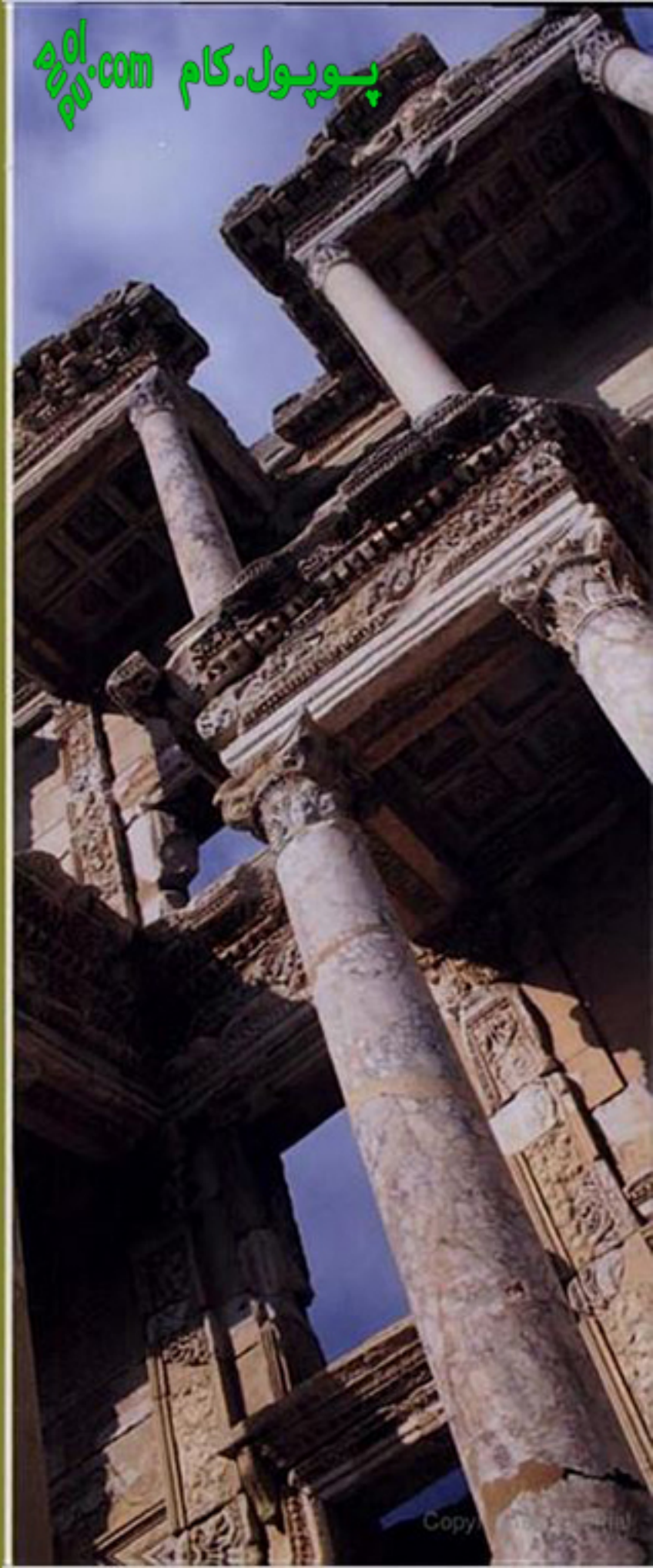
UML Rational rose

به زبان فارسی



پوپول مرجع دانشگاه و مدرسه
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه

preview

بزرگترین مرجع کتابهای الکترنیکی فارسی وانگلیسی
بزرگترین مرجع نرم افزارهای کاربردی و تخصصی
بزرگترین مرجع دانلود کلیپهای موبایل

www.IranMeet.com

Ramin.Samad @yahoo.com

www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

باسمه تعالی

آموزش
UML-Rational Rose

تهیه کننده:

مسعود حاجی اسماعیلیان

آبان ۱۳۸۳

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

Rena .....

فهرست مطالب

۱) مقدمه‌ای بر UML
۷ .....
۱-۱) UML و شی گرایی
۷ .....
۱-۲) مدلسازی بصری(Visual Modeling)
۸ .....
۱-۳) نمودارهای UML
۹ .....
۱-۴) مدلسازی بصری، توسعه نرم افزار و پردازش تولید
۱۱ .....

۲) گردشی در برنامه Rational Rose
۱۴ .....
۲-۱) بخشهای صفحه نمایش
۱۵ .....
۲-۲) کار با برنامه Rational Rose
۱۷.....

۳) Use Caseها و Actorها
۱۸.....
۳-۱) روش کار با نمودارها، عناصر مدل و دیگر موارد عمومی
۱۹.....
۳-۲) کار با Use Caseها
۲۲.....
۳-۳) افزودن Use Case موجود به نمودار Use Case
۲۴.....
۳-۴) مشخصات یک Use Case
۲۴.....
۳-۵) کار با عامل
۲۴.....
۳-۶) چگونگی کار با رابطه‌ها
۲۵.....

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

۴) محاورات آبجکتی
۲۷.....
۴-۱) نمودارهای تعامل
۲۷.....
۴-۲) نمودارهای توالی
۲۸.....
۴-۳) نمودارهای همکاری
۲۸.....
۴-۴) کار با آبجکت‌ها
۲۸.....
۴-۵) کار با پیغام‌ها
۲۹.....
۴-۶) کار با اسکریپت‌ها
۳۱ .....
۴-۷) سوئیچ کردن بین نمودارهای توالی و همکاری
۳۲.....
۴-۸) روش‌های دو گذره(Two-Pass) برای نمودارهای تعامل
۳۲.....

۵) کلاسها و بسته‌ها
۳۳.....
۵-۱) نمودارهای کلاس
۳۳.....
۵-۲) یافتن کلاسها
۳۳.....
۵-۳) ساختن نمودارهای کلاس
۳۴.....
۵-۴) کار با کلاسها
۳۴.....
۵-۵) افزودن کلاسهای معمولی
۳۴.....
۵-۶) مشخصات کلاس
۳۴.....
۵-۷) استفاده از کلاس‌های تو در تو(Nested)
۳۷.....

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

۵-۸) دیدن نمودارهای تعامل شامل یک کلاس
۳۷.....
۵-۹) کار با بسته‌ها
۳۷.....
۵-۱۰) انواع ویژه کلاس
۳۷.....

۶) صفات(Attributes) و عملیات(Operations)
۴۰.....
۶-۱) یافتن صفات
۴۰.....
۶-۲) افزودن صفات
۴۱.....
۶-۳) مشخصات صفت
۴۱.....
۶-۴) کار با عملیات
۴۳.....
۶-۵) یافتن عملیات
۴۳.....
۶-۶) افزودن عملیات
۴۴.....
۶-۷) مشخصات عملیات
۴۴.....
۶-۸) نمایش صفات و عملیات بر روی نمودارهای کلاس
۴۶.....

۷) رابطه‌ها(Relationships)
۴۸.....
۷-۱) پیدا کردن رابطه‌ها
۴۸.....
۷-۲) Associationها
۴۸.....
۷-۳) Dependencyها
۴۹.....
۷-۴) Aggregationها
۵۰.....

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

Rena .....

۷-۵) Generalitationها
۵۰.....
۷-۶) مشخصات رابطه‌ها
۵۱.....

۸) رفتار آبجکت
۵۴.....
۸-۱) نمودارهای تغییر حالت
۵۴.....
۸-۲) اضافه کردن حالت
۵۵.....
۸-۳) اضافه کردن گذرها(Transitions)
۵۶.....
۸-۴) اضافه کردن حالات خاص
۵۷.....
۸-۵) استفاده از حالات تو در تو(Nested States)
۵۷ .....

۹) نمای Component
۵۹.....
۹-۱) انواع کامپونت‌ها
۵۹.....
۹-۲) نمودارهای کامپونت
۶۰.....
۹-۳) مشخصات کامپونت
۶۰.....
۹-۴) اضافه کردن رابطه‌های Dependency در کامپونت‌ها
۶۱.....

۱۰) نمای Deployment
۶۲.....
۱۰-۱) نمودار Deployment
۶۲.....
۱۰-۲) افزودن پردازنده‌ها(Processor)
۶۲.....

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

۱۰-۳) اضافه کردن ابزارها(Devices)
۶۳.....
۱۰-۴) اضافه کردن رابطه‌ها
۶۴.....
۱۰-۵) نشان دادن فرآیندها و نوع زمانبندی آنها در نمودار Deployment
۶۴.....

۱۱) مقدمه‌ای بر تولید کد(Code Generation)
۶۵.....
۱۱-۱) چک کردن مدل
۶۵.....
۱۱-۲) ایجاد کامپوننت‌ها
۶۶.....
۱۱-۳) نگاشت کلاسها به کامپوننت‌ها
۶۶.....
۱۱-۴) تعیین خصوصیات تولید کد
۶۶.....
۱۱-۵) انتخاب کلاس، کامپوننت یا بسته جهت تولید کد
۶۷.....
۱۱-۶) تولید کد
۶۷.....
۱۱-۷) چه چیزی تولید می‌شود
۶۷.....

۱۲) تولید کد C++ و Visual C++ و مهندسی معکوس در Visual C++
۶۹.....
۱۲-۱) نحوه تولید کد در C++
۶۹.....
۱۲-۲) خصوصیات(Properties) تولید کد C++
۷۰.....
۱۲-۳) نحوه تولید کد در Visual C++
۷۲.....

۱۳) ماخذ
۷۶.....

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

فصل اول

مقدمه‌ای بر UML

UML یک زبان مدلسازی است که بطور گسترده‌ای در نمادهای مدلسازی سیستم‌های شی گرا استفاده شده است.

۱-۱) UML و شی گرایی

UML یک زبان کاملا شی گرا است.

تفاوت روش شی گرایی و Data-Centric: دومی متمرکز روی اطلاعات است و مخصوص طراحی پایگاه داده و گرفتن اطلاعات خیلی مهم است و در کاربردهای تجاری جواب نمی‌دهد زیرا درخواست‌های سیستم در کاربردهای تجاری مکرر تغییر می‌کند. در متد شی گرایی هم بر اطلاعات و هم بر رفتار متمرکز می‌شویم.

۱-۱-۱) Encapsulation (نهان سازی): به معنی بسته بندی اطلاعات و رفتارها در یک آبجکت است و مزیت آن محدود کردن ت أثیرات اعمال شده ناشی از تغییرات به سیستم است.

۱-۱-۲) Inheritance(وراثت): ایجاد آبجکت‌های جدید بر پایه آبجکت‌های قدیمی را گویند. این ایده از طبیعت گرفته شده است. از مزایای آن سهولت در نگهداری است.(تغییرات فقط در آبجکت پدر داده می‌شود)

۱-۱-۳) Polymorphism(چند ریختی): داشتن شکلها و پیامدهای مختلف از یک تابع را گویند. در روش قبل یک تابع داشتیم و در داخل آن تفکیک صورت می‌گرفت و حالا هر آبجکت تابع خودش را دارد.

۱-۲) مدلسازی بصری(Visual Modeling)

مدل‌ها همان طرحهای کلی سیستم می‌باشند و باعث می‌شوند که تغییرات درخواستی به راحتی اعمال شود. گرفتن نیازهای تجاری و تبدیل آن‌ها به

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

مدل‌های بصری که در نهایت می‌تواند به کد تبدیل شود را مدلسازی بصری گویند. این روش تضمین می‌کند که در برگشت از کد به مدل و تغییر نیازها چیزی گم نشود. مدلسازی بصری با استفاده از مجموعه‌ای از عناصر گرافیکی استاندارد صورت می‌گیرد.

هدف اصلی مدلسازی بصری ارتباط میان کاربران، برنامه‌نویسان، تحلیل گران، آزمایش کننده‌ها، مدیران و ... است.

مدلسازی بصری باعث می‌شود که پیچیدگیهای سیستم را بهتر از حالتی که فقط متنی نوشته شده است درک کنیم. با مدلسازی بصری میتوان تعاملات بین کاربران و سیستم، همچنین بین آبجکت‌ها در محدوده یک سیستم و همچنین بین سیستم‌ها را مدل کنیم.

بعد از تهیه مدل هر کسی به فراخور می‌تواند از مدل استفاده کرده و اطلاعات مورد نیاز را بدست آورد.

۱-۲-۱) BOOCH، OMT، UML: هر کدام یک نماد گرافیکی هستند. لازم است کل سیستم از یک نماد گرافیکی استفاده کنند. OMT مخفف Object Modeling Technology است. Rational از هر ۳ پشتیبانی می‌کند.

BOOCH از نام مخترعش بنام Grady Booch گرفته شده است.(دانشمند ارشد شرکت Rational) در نماد Booch آبجکت‌ها با ابر نمایش داده می‌شود.

نماد OMT توسط دکتر Rumbaugh(نویسنده چندین کتاب درباره تجزیه، تحلیل و طراحی سیستم) بوجود آمده. OMT نسبت به BOOCH از اشکال گرافیکی بیشتری استفاده می‌کند.

UML حاصل تلاش Booch، Rumaugh، Jacobson و... می‌باشد که به این ۳ نفر ۳ تفنگدار می‌گویند و هر ۳ در Rational کار می‌کنند.

UML با دو نماد دیگر متناسب بوده و شامل عناصری از نمادهای دیگر می‌باشد. نمایش سیستم و کاربر در هر ۳ نماد مشابه است. نماد شی و کلاس در OMT و UML مشابه است.

به هم آمیختن متدولوژی به منظور ایجاد UML از ۱۹۹۳ شروع و در ۱۹۹۵ اولین نسخه(۰/۸) با نام Unified Method بیرون آمد و در ۱۹۹۶ به UML تغییر نام داد. نسخه ۱ در سال ۱۹۹۷ به OMG (object Management Group) داده شد و بعد از آن شرکتها شروع به استفاده از آن کردند. در این سال OMG نسخه ۱.۱ از UML را به عنوان یک استاندارد معرفی کرد.

۱-۳) نمودارهای UML

نمودارهای UML جنبه‌های مختلف سیستم را نشان می‌دهند.

۱-۳-۱) نمودارهای Use Case: محاورات میان سیستم‌ها(Use Caseها که از این به بعد با عنوان اختصاری UC می‌آید) و عامل‌ها(کاربران یا سیستم‌های خارجی) را نشان می‌دهد و UCها درخواست‌های

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

سیستم را نشان می‌دهد. عامل‌ها به UCها مقدار اولیه می‌دهند یا اطلاعات را از آن دریافت می‌کنند.(فلش از عامل به UC و فلش از UC به عامل)

این نمودار می‌تواند مورد استفاده مدیران، کاربران و... قرار گیرد. جهت مشاهده مثال به صفحه ۳۷ از کتاب مراجعه شود.

۱-۳-۲) نمودار Sequence(توالی): جریان عملیات را در یک UC نشان می‌دهد. عامل و آبجکت‌ها در بالای نمودار نشان داده می‌شوند، هر فلش یک پیغام ارسالی بین عامل و آبجکت یا بین آبجکت و آبجکت را نشان می‌دهد. در نمودار توالی می‌توان از یک آبجکت به خودش فلش داشت. نمودار توالی آبجکت‌ها را نشان می‌دهد و نه کلاس‌ها را.

این نمودار می‌تواند مورد استفاده طراحان و... قرار گیرد. جهت مشاهده مثال به صفحه ۳۹ از کتاب مراجعه شود.

۱-۳-۳) نمودار Collaboration(همکاری): نشان دهنده همان اطلاعات نمودار توالی با روش و هدفی متفاوت است. در نمودار توالی آبجکت‌ها و ارتباطات به ترتیب زمان توضیح داده می‌شوند و در این نمودار بدون توجه به عامل زمان. در صورتیکه تمرکز بیش از حد روی آبجکت‌های خاصی وجود داشته و پراکندگی به درستی صورت نگرفته باشد این نمودار آنرا نشان می‌دهد.

این نمودار می‌تواند مورد استفاده طراحان و... قرار گیرد. جهت مشاهده مثال به صفحه ۴۰ از کتاب مراجعه شود.

۱-۳-۴) نمودارهای کلاس(Class): ارتباطات بین کلاسها را نشان می‌دهد. در شکل گرافیکی کلاس قسمت بالا نام کلاس، قسمت وسط صفات و قسمت پایین عملکردها را نشان می‌دهد. خطوط بین کلاسها وابستگی آنها به یکدیگر را نشان می‌دهد. Rose چارچوب کلاس را در کد برای ما تولید می‌کند و بقیه جزئیات با برنامه‌نویس است.

یکدسته از نمودارهای کلاس شامل کلاس‌های ذیل یک UC و تعداد نمودارهای جامع‌اند که شامل کل سیستم و فراتر از یک UC می‌باشد.

این نمودار می‌تواند مورد استفاده برنامه‌نویسان قرار گیرد. جهت مشاهده مثال به صفحه ۴۱ از کتاب مراجعه شود.

۱-۳-۵) نمودارهای حالت(State Transition): حالت‌های مختلف یک آبجکت را مدل کرده و رفتارهای پویای یک کلاس را نشان می‌دهد. رفتار کلاس یا آبجکت در هر یک از حالت‌ها متفاوت است و چگونگی انتقال از یک حالت به حالت دیگر نمایش داده می‌شود. انتقال می‌تواند در اثر یک رخداد(Event) صورت گیرد و خود نیز می‌تواند رخداد جدیدی را تولید کند. می‌توان برای انتقال شرط معین کرد. حالت شروع و حالت پایان هم وجود دارد که از اولی به تعداد ۱ و از دومی به تعداد نامحدود می‌توانیم

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

داشته باشیم یا نداشته باشیم. action اتفاقاتی است که هنگامی که آبجکت در یک حالت خاصی قرار دارد، می‌افتد. معمولا فقط کلاسهای پیچیده نمودار حالت دارد و نه هر کلاسی. این نمودار ت أثیری در تولید کد ندارد و فقط در مستندسازی استفاده می‌شود.

برنامه‌نویسان از آنها در زمان تولید کد استفاده می‌کنند. جهت مشاهده مثال به صفحه ۴۳ از کتاب مراجعه شود.

۱-۳-۶) نمودارهای اجزاء(Component): یک دید فیزیکی از مدل، اجزای نرم افزاری و روابط بین آنها را نشان می‌دهد.دو نوع کامپوننت در نمودار داریم یکی کامپوننت‌های قابل اجرا و دیگری کتابخانه‌های کد. هر کلاس به یک یا دو کامپوننت(بسته به زبان مورد استفاده) نگاشته می‌شود.ارتباطات بین کامپوننت‌ها نشان دهنده وابستگی‌های زمان اجرا(Run Time) و زمان کامپایل(Compile Time) می‌باشد.

فلش‌های خط چین وابستگی کامپونتت‌ها را نشان می‌دهد به این معنا که کامپیونتی که در انتهای فلش قرار دارد کلاس آن اول باید کامپایل شود و الی آخر.

بسته به تعداد زیر سیستم‌ها(UCها) یا قابلیت اجرایی می‌توان چندین نمودار اجزا داشت. این نمودار بعد از تولید کد ایجاد می‌شوند.

برنامه‌نویسان از آنها در زمان تولید کد استفاده می‌کنند. جهت مشاهده مثال به صفحه ۴۵ از کتاب مراجعه شود.

۱-۳-۷) نمودارهای Deployment(گسترش): لایه فیزیکی شبکه، سرورها، بانکهای اطلاعاتی، وسایل جانبی و ارتباطات شبکه محلی را نشان می‌دهد و مورد استفاده مدیر پروژه، کاربران، طراحان و مابقی پرسنل برنامه نویسی قرار می‌گیرد. جهت مشاهده مثال به صفحه ۴۶ از کتاب مراجعه شود.

۱-۴) مدلسازی بصری، توسعه نرم افزار و پردازش تولید

- انواع پردازش تولید وجود دارد از قبیل پردازش آبشاری و شی‌گرایی.
- مدل آبشاری برای مدتی طولانی مورد استفاده برنامه‌نویسان بوده که به ترتیب درخواستها را تجزیه و تحلیل، سیستم را طراحی و تولید و سپس آزمایش می‌کردیم. همانطوریکه از اسم این روش پیداست نمی‌توان برخلاف جریان حرکت کرد. این مدل در پروژه‌های بی‌شماری استفاده شده است.
- یکی از اشکالات مدل آبشاری نیاز به بازگرد از وسط مراحل به خارج از مجموعه پروژه می‌باشد. در هر یک از مراحل ممکن است نیاز به تغییر در تحلیل و طراحی وجود داشته باشد.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- پروژه تا به نتیجه برسد مدت زمان زیادی طول می‌کشد و احتمالا در این مدت اصول تجارت عوض شده است. در نتیجه کاربر می‌گوید: «این فقط آن چیزی است که من تقاضا کردم نه آن چیزی که دوست دارم».
- **روش تولید مکرر(Iterative Development):** در این روش و با استفاده از پردازش شی‌گرا ما کارها را بارها و بارها انجام می‌دهیم بدین صورت که در سراسر مراحل تجزیه و تحلیل، طراحی، تولید و تست در مقاطع کوچک بارها حرکت می‌کنیم.
- این غیر ممکن است که همه درخواستها را در طول بخش نخست پروژه بفهمیم. لذا آبشار را به یکسری از آبشارهای کوچک تقسیم می‌کنیم. هر کدام از این آبشارهای کوچک که طراحی شدند به اندازه کافی بزرگ هستند تا پایان یک بخش مهم پروژه را نشان دهند و از طرفی آنقدر کوچک هستند که نیاز به بازگرد(Backtracking) را به حداقل برسانند.
- هر پروژه را در طی ۴ فاز به انجام می‌رسانیم که به ترتیب انجام می‌شوند. اولی فقط یکبار انجام شده و ۳ تای بعدی می‌توانند تکرار شوند.

۱-۴-۱) Inception(شناخت): فاز شروع پروژه است که اطلاعات را جمع‌آوری کرده و برداشت کلی خود را در مورد پروژه بیان می‌کنیم. پایان این فاز تصمیم درباره اجرا یا عدم اجرای پروژه است.(همان امکان سنجی است)

- در این مرحله UCها و عامل‌ها(Actors) شناخته می‌شوند ولی بدون جزئیات. نمودارهای UC در این مرحله کشیده می‌شود. هم چنین تخمینی از کل پروژه در این مرحله زده می‌شود.(از نظر زمانی و حجم کار و هزینه‌ها)
- این فاز دنباله‌دار(متوالی) و غیر تکراری است(در داخل آن بازگرد نداریم) در حالیکه ما بقی فازها تکراری هستند.
- این فاز فقط یکبار انجام می‌شود و UCهایی که قرار است با هم یک آبشار کوچک تشکیل دهند دسته‌بندی می‌شوند. دسته‌بندی باید طوری باشد که هر یک بتواند بترتیب بعد از دیگری اجرا شود.

۱-۴-۲) Elaboration(مهارت): شامل مقداری تجزیه و تحلیل، طراحی، کدنویسی و تست است.

- این فاز برای هر UC انجام می‌شود.(که هر UC خود جزئی از Iteration است) در این فاز UC تکمیل می‌شود. نمودارهای Interaction (Collaboration و Sequence) در این فاز تکمیل می‌شوند.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- SRS یا مشخصات مورد نیاز و درخواست شده نرم‌افزار(Software Requirement Specification) سندی است که شامل کلیه درخواستها و جزئیات UCها می‌شود.
- در این فاز تخمین‌های اولیه(در Inception) اصلاح می‌شوند.
- نمودارهای Class, Collaboration, Sequence و State در این فاز برای هر UC کامل می‌شود. در انتهای این فاز سیستم آماده برنامه‌نویسی است.

۱-۴-۳) Construction(ساختار): تولید(برنامه نویسی) و تست نرم‌افزار در این فاز انجام می‌شود.

- این فاز نیز برای هر UC در طی تکرار(Iteration) انجام شده و درگیر تصمیم‌های طراحی نمی‌شود و لذا تولید برنامه می‌تواند بصورت موازی انجام شود.
- جهت تولید کد نیاز به تعریف کامپوننت‌ها و رسم نمودار کامپوننت می‌باشد. تولید کد شامل تعریف کلاس، تعریف صفات، تعریف توابع و تعریف محدوده صفات و توابع و دستورات وراثتی می‌باشد.
- بعد از این که کد تکمیل شد باید توسط یک گروه همتراز از برنامه نویسان بازبینی شود تا از حیث استاندارد و رعایت مسائل مطروحه در طراحی مطمئن شوند.
- بعد از بازبینی کد آبجکت‌ها توسط تضمین کیفیت بازبینی می‌شود تا در صورتیکه صفات و توابع جدیدی در فاز ساختار اضافه شده‌اند یا تعاملات بین آبجکت‌ها تغییر کرده، کد جدید از طریق مهندسی معکوس مدل Rose را بروز رسانی کند.
- فاز ساختار زمانی به پایان می‌رسد که نرم افزار کامل شده و تست شده باشد و هم چنین مدل با کد سنکرون شده باشند.

۱-۴-۴) Transition(انتقال): در این فاز محصول نرم افزاری کامل شده به سوی کاربر بر می‌گردد.

- وظایف این فاز مشتمل است بر کامل کردن محصول نرم افزاری نهایی، تکمیل تست ت أیید نهایی، کامل کردن مستندسازی و فراهم کردن آموزش کاربر.
- نمودار Deployment در این فاز رسم می‌شود.
- مدلی که در این فاز وجود دارد آخرین مدل است که به مد پشتیبانی می‌رود و ممکن است چند ماه بعد به کار آید لذا باید کاملا با کد سنکرون باشد.

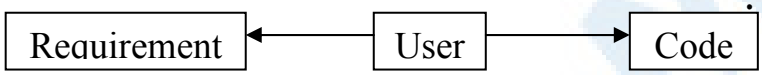
Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

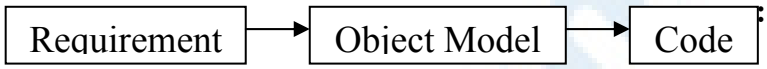
گردشی در برنامه Rational Rose

- Rose ابزار قدرتمندی است که به تجزیه و تحلیل و طراحی سیستم‌های نرم‌افزاری شی‌گرا کمک می‌کند.

طرح قدیمی وابسته به فرد:



طرح جدید وابسته به مدل:



- چه کسی از چه نموداری استفاده می‌کند:
  - مشتری و مدیر پروژه از نمودار UC
  - مدیر پروژه از مستندات نمودار UC
  - تحلیل گر ومشتری از مستندات نمودار UC
  - نویسندگان تکنیکی از مستندات نمودار UC
  - تحلیل‌گران و برنامه نویسان از نمودارهای Interaction(تعامل)
  - کارمندان تضمین کیفیت از اسناد UC و نمودارهای تعامل
  - برنامه نویسان از نمودارهای کلاس و حالت
  - کارمندان Deployment از نمودارهای Deployment و Component
- نسخه‌های مختلف نرم افزار Rose عبارتند از:
  - **Rose Modeler:** فقط اجازه ایجاد مدل را می‌دهد، تولید کد و مهندسی معکوس منتفی است.
  - **Rose Professional:** فقط اجازه تولید کد را می‌دهد.
  - **Rose Enterprise:** ایجاد مدل، تولید کد و مهندسی معکوس امکان پذیر است.
- نصب Rational Rose: (نصب Visual Studio از قبل الزامی است)

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- گزینه Compact فقط Modeler و Help را نصب می‌کند.
- گزینه Typical بقیه اجزاء Rose را هم نصب می‌کند.(بعلاوه همه Add-Inها)
- گزینه Custom اجازه می‌دهد هر Add-In را که خواستیم نصب کنیم.
- اگر زبان Analysis انتخاب شود عملا زبان پیش فرض نخواهیم داشت. هر زبان دیگری انتخاب شود کتابخانه‌های آبجکت آن زبان نیاز اضافه خواهد شد.

۲-۱) بخشهای صفحه نمایش

۲-۱-۱) Browser(مرورگر): محلی است که ساختار سلسله مراتبی را جهت حرکت در طول مدل و نمایش عناصر آن شامل می‌شود. با استفاده از امکانات این پنجره میتوان کارهای ذیل را انجام داد:

- اضافه کردن عناصر مدل و هم چنین حذف یا تغییر نام آنها و نیز دیدن روابط بین آنها
- اضافه کردن عناصر مدل به نمودار و باز کردن نمودار
- گروه‌بندی عناصر مدل و ...

پنجره مرورگر شامل ۴ نمای مختلف است که عبارتند از:

۲-۱-۱-۱) نمای Use Case: این نما می‌تواند شامل عناصری از مدل از قبیل: عامل‌ها، UCها، Associationها، مستندسازی‌های UC، نمودارهای UC ، توالی و همکاری و هم چنین بسته‌ها(Package) باشد.

- این نما امکانی است برای نگاه سطح بالا به سیستم بدون اینکه بخواهیم درگیر جزئیات بشویم.
- با توجه به اینکه نمودارهای توالی و همکاری هم در این نما و هم در نمای منطقی می‌توانند بیایند لذا نمودارهایی از این نوع که وابسته به زبان برنامه‌نویسی نیستند در این نما و آنهایی که وابسته هستند در نمای منطقی می‌آیند.
- استفاده کنندگان اصلی نمای Use Case عبارتند از: مشتریان، تحلیل گران و مدیران پروژه

۲-۱-۱-۲) نمای Logical(منطقی): این نما می‌تواند شامل عناصری از مدل از قبیل: کلاس‌ها، بسته‌ها، نمودارهای کلاس، توالی، همکاری و حالت، Associationها باشد.

- نمای منطقی بر روی این متمرکز می‌شود که چگونه سیستم رفتار را در UCها پیاده می‌کند.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- نمودارهای تعامل اگر در این نما بیایند بیشتر بر روی کلاس‌ها متمرکز می‌شوند.
- Reuse یکی از ملاحظات اصلی است لذا می‌بایستی کلاسها، روابط بین آنها و دسته‌بندی آن به دقت تعریف شوند.
- استفاده کنندگان اصلی این نما طراحان و برنامه نویسان می‌باشند.

۲-۱-۱-۳) نمای Component: این نما شامل عناصری از مدل از قبیل: کامپونت‌ها، نمودارهای کامپونت و بسته‌ها می‌باشد.

- نمای کامپونت به ما اجازه می‌دهد تا ارتباطات بین ماژول‌های کد را ببینیم
- کاربران اصلی این نما مسئولین کنترل کد و ارتقاء دهندگان و برنامه نویسان می‌باشند

۲-۱-۱-۴) نمای Deployment: این نما شامل عناصری از مدل از قبیل: پردازنده‌ها، وسایل و ابزار جانبی و نمودار Deployment می‌شود. کاربران اصلی این نما کارمندان Deployment هستند.

۲-۱-۲) پنجره Document(مستندات): این پنجره مستندات عنصر فعال را نشان می‌دهد.

۲-۱-۳) پنجره Tool Bar(نوار ابزار): جهت دستیابی سریع به فرمانهای عمومی می‌باشد.

- دو نوع نوار ابزار وجود دارد یکی استاندارد که همیشه وجود دارد و دیگری نوار ابزار نمودار که برای هر نوع نمودار تغییر می‌کند.
- دکمه‌های نوار ابزار استاندارد عبارتند از: New, Open, Save و...
- دکمه‌های نوار ابزار نمودار در محل خود توضیح داده شده است.
- هر یک از نوار ابزارها را می‌توان مخفی کرد.
- هر یک از نوار ابزارها را می‌توان سفارشی کرد.

۲-۱-۴) پنجره نمودار(Diagram): جهت نمایش و ویرایش انواع نمودار بکار میرود.

۲-۱-۵) پنجره Log: جهت دیدن خطاها و گزارش نتایج فرمان بکار میرود. این پنجره را نمیتوان بست و فقط میتوان مینیمم کرد.

۲-۲) کار با برنامه Rational Rose

- هر مدل در یک فایل با پسوند MDL ذخیره می‌شود.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- مدل جدید را میتوان از روی یک Template ساخت.
- میتوان محتویات پنجره Log را در یک فایل ذخیره کرد.
- میتوان کل مدل، یک بسته از کلاس‌ها یا یک کلاس خاص را Export یا Import کرد که در Reuse بسیار مفید است.
- میتوان یک مدل را بر روی وب منتشر کرد.(از مدل، خروجی وب گرفت)

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

Use Caseها و Actorها

- UCها و عامل‌ها محدوده سیستمی را که در حال ساخت آن هستید مشخص می‌کنند.
  - UCها شامل تمام آن چیزهایی است که درون سیستم قرار دارد.
  - عامل‌ها شامل تمام آن چیزهایی است که خارج از سیستم قرار دارد.
- در صورتیکه دارای چند بسته و نمودار باشیم نمودار Main فقط شامل بسته‌ها خواهد بود.

- نکاتی درباره نمودار UC که باید رعایت شوند:
- ارتباط عامل با عامل مجاز نیست(اگر چه امکان پذیر است) بجای آن می‌توان از نمودار Workflow استفاده کرد.
- UCها با هم ارتباط داده نمی‌شوند.(بجز در ارتباطات Extend یا Include) نمودار UC روند و ترتیب اجرای UC را نشان نمی‌دهد. برای چنین منظوری می‌توان از نمودار فعالیت(Activity) استفاده کرد. استفاده از ارتباط بین UCها برای نمایش جریان اطلاعات مجاز نیست.
- هر UC باید توسط یک عامل شروع بکار کند(فلش از عامل به UC) بجز در روابط Extend و Include.

۳-۱) روش کار با نمودارها، عناصر مدل و دیگر موارد عمومی(

این روش‌ها در مورد تمامی نمودارها و عناصر مدل Rose صادق است و لذا فقط یک‌بار ذکر می‌شود)

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

۳-۱-۱) کار با نمودارها

- **۳-۱-۱-۱) روش‌های ایجاد نمودار:** کلیک راست روی نمایی که نمودار مورد نظر می‌تواند در آن ایجاد شود(مثلا نمای Use Case برای نمودار Use Case و نمای منطقی برای نمودار کلاس) یا بسته ذیل آن و انتخاب آیتم نمودار همنام(مثلا Use Case Diagram) از منوی New

۳-۱-۱-۲) روش‌های باز کردن نمودار

- دوبار کلیک کردن روی نام نمودار در پنجره مرورگر
- انتخاب آیتم نمودار همنام(مثلا ... Use Case Diagram) از منوی Browse
- انتخاب آیکون نمودار از نوار ابزار اصلی

در دو روش اخیر پنجره‌ای باز می‌شود که لیست نماها و بسته‌ها را در سمت چپ و لیست نمودارهای هر نما یا بسته را در سمت راست نشان می‌دهد و می‌توان نمودار مورد نظر خود را انتخاب کرده و باز کرد.

- **۳-۱-۱-۳) روش‌های حذف نمودار**(نمودار اصلی هر نما(با نام Main) را نمیتوان حذف نمودار حذف شده را با Undo برگرداند)
- کلیک راست روی نام نمودار در پنجره مرورگر و انتخاب آیتم Delete
- انتخاب آیتم نمودار همنام از منوی Browse، انتخاب نمودار مورد نظر از لیست نمودارها در پنجره باز شده و فشردن کلید Delete
- انتخاب آیکون نمودار از نوار ابزار اصلی، انتخاب نمودار مورد نظر از لیست نمودارها در پنجره باز شده و فشردن کلید Delete

۳-۱-۱-۴) روش‌های تغییر نام نمودار

- کلیک راست روی نام نمودار در پنجره مرورگر و انتخاب آیتم Rename
- انتخاب آیتم نمودار همنام از منوی Browse، انتخاب نمودار مورد نظر از لیست نمودارها در پنجره باز شده و فشردن کلید Rename
- انتخاب آیکون نمودار از نوار ابزار اصلی، انتخاب نمودار مورد نظر از لیست نمودارها در پنجره باز شده و فشردن کلید Rename
- دو بار کلیک روی نام نمودار در پنجره مرورگر

Info@IRANMEET.COM
www.pupuol.com

۳-۱-۲) کار با عناصر مدل

۳-۱-۲-۱) روش‌های ایجاد عناصر مدل

- کلیک راست روی نمایی که عنصر از نوع مورد نظر می‌تواند در آن ایجاد شود یا بسته ذیل آن و انتخاب آیتم عنصر همنام(مثلا Use Case) از منوی New. این روش عنصر مورد نظر را به نمودار اضافه نمی‌کند.
- انتخاب منوی Tools، منوی Create، آیتم عنصر همنام و کلیک کردن در جایی از نمودار.(نمودار باید از قبل باز شده باشد) این روش عنصر مورد نظر را هم به مدل و هم به نمودار اضافه می‌کند.
- انتخاب آیکون عنصر همنام از نوار ابزار نمودار و کلیک کردن در جایی از نمودار.(نمودار باید از قبل باز شده باشد) این روش عنصر مورد نظر را هم به مدل و هم به نمودار اضافه می‌کند.

۳-۱-۲-۲) روش‌های باز کردن پنجره مشخصات عنصر مدل

- کلیک راست روی نام عنصر در پنجره مرورگر و انتخاب آیتم Open Specification ...
- کلیک راست روی عنصر در نمودار و انتخاب آیتم Open Specification ...
- کلیک روی عنصر در نمودار، انتخاب منوی Browse، آیتم Specification ...
- کلیک مجدد(Double Click) روی عنصر در نمودار
- کلیک روی عنصر در نمودار و فشار دادن کلید Ctrl+B

۳-۱-۲-۳) روش‌های حذف عنصر مدل

(می‌توان با Undo از منوی Edit عنصر حذف شده را برگرداند)

کلیک روی عنصر در نمودار و فشار دادن کلید Del(این روش فقط عنصر را از نمودار حذف می‌کند)

کلیک راست روی عنصر در نمودار و انتخاب آیتم Delete از منوی Edit(این روش فقط عنصر را از نمودار حذف می‌کند)

کلیک روی عنصر در نمودار و انتخاب آیتم Delete از منوی Edit(این روش فقط عنصر را از نمودار حذف می‌کند)

کلیک راست روی نام عنصر در پنجره مرورگر و انتخاب آیتم Delete (این روش عنصر را از نمودار و از مدل حذف می‌کند)

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- کلیک روی عنصر در نمودار و فشار دادن کلید Ctrl+D(این روش عنصر را از نمودار و از مدل حذف می‌کند)
- کلیک راست روی عنصر در نمودار و انتخاب آیتم Delete from Model از منوی Edit(این روش عنصر را از نمودار و از مدل حذف می‌کند)
- کلیک روی عنصر در نمودار و انتخاب آیتم Delete from Model از منوی Edit(این روش عنصر را از نمودار و از مدل حذف می‌کند)

۳-۱-۲-۴) روش‌های تغییر نام عنصر مدل

- دو بار کلیک روی نام عنصر در پنجره مرورگر
- کلیک راست روی نام عنصر در پنجره مرورگر و انتخاب آیتم Rename
- کلیک روی عنصر در نمودار
- باز کردن پنجره مشخصات عنصر و تغییر فیلد Name از برگه General

۳-۱-۲-۵) روش اضافه کردن یک عنصر موجود در مدل به نمودار

- انتخاب عنصر در پنجره مرورگر، کشیدن و رها کردن آن در نمودار
- **۳-۱-۲-۶) روش‌های اضافه کردن مستندات(Documentation) به عنصر مدل**
- کلیک روی نام عنصر در پنجره مرورگر و نوشتن مستندات ربوطه در پنجره مستندات
- کلیک روی عنصر در نمودار و نوشتن مستندات مربوطه در پنجره مستندات
- باز کردن پنجره مشخصات عنصر و تغییر فیلد Documentation از برگه General
- **۳-۱-۲-۷) تخصیص یک Stereotype به عنصر مدل(یا به مشخصات آن) :** در UML، Stereotype برای کمک به طبقه‌بندی عناصر مدل بکار میرود. بیشترین کاربرد آنها در کلاس‌ها و ارتباطات است. برای تعیین آن کافی است پنجره مشخصات عنصر مدل را باز کرده و از لیست کشویی Stereotype در برگه General آنرا انتخاب کرده یا یک طبقه جدید معین کنیم.

۳-۱-۳) کار با فایل و URLهای الصاقی به نمودارها و عناصر مدل

۳-۱-۳-۱) روش‌های الصاق فایل و URL به نمودارها و عناصر مدل

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- کلیک راست روی نام نمودار یا عنصر مدل در مرورگر و انتخاب آیتم File یا URL از منوی New
- در پنجره مشخصات نمودار یا عنصر مدل، انتخاب برگه Files، کلیک راست روی لیست، انتخاب آیتم Insert File یا Insert URL
- برای باز کردن کافی است روی فایل یا URL کلیک دوباره کنیم یا از منوی Open استفاده کنیم.

۳-۱-۳-۲) روش‌های حذف فایل و URLهای الصاقی از نمودارها و عناصر مدل

- کلیک راست روی نام فایل یا URL در پنجره مرورگر و انتخاب آیتم Delete
- در پنجره مشخصات نمودار یا عنصر مدل، انتخاب برگه Files، کلیک راست روی فایل یا URL مورد نظر در لیست، انتخاب آیتم Delete

۳-۱-۴) کار با توضیحات(Notes): برای ارائه توضیحات اضافی به یک عنصر مدل در نموداراستفاده می‌شود و دو نوع یادداشت توضیحی وجود دارد.

- **۳-۱-۴-۱) کادر متن:** که جهت گذاشتن عنوان برای یک نمودار یا توضیحی مختصر برای یک رابطه بکار میرود. کادر متن با استفاده از آیکون ABC ایجاد می‌شود.

۳-۱-۴-۲) یادداشت: که جهت گذاشتن توضیحات اضافه برای یک عنصر خاص تعریف می‌شود.

- یادداشت با استفاده از آیکون Note ایجاد می‌شود. یادداشت هیچ مشخصه‌ای غیر متنی که در داخل کادر آن قرار می‌گیرد ندارد.
- برای افزودن یک یادداشت به یک عنصر خاص از آیکون Anchor Note To Item استفاده می‌شود.

۳-۱-۵) کار با بسته‌ها(Packages): بسته‌ها برای ایجاد دسته‌بندی و گروه‌بندی آیتم‌ها و نمودارها استفاده می‌شود.

- می‌توان بسته‌های تو در تو ایجاد کرد.
- می‌توان یک بسته را به یک نمودار اضافه کرد. با کلیک دوباره روی یک بسته در یک نمودار، نمودار دیگری از همان نوع داخل بسته ایجاد می‌شود.
- روش‌های ایجاد، حذف و تغییر نام بسته مشابه عناصر مدل می‌باشد.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- وقتی یک بسته از مدل حذف می‌شود تمام اجزا داخل آن نیز از مدل حذف می‌شود.

۳-۲) کار با UCها

- UC را می‌توان با نگاه به سندی که مشتری خواسته‌های خود را ذیل آن بیان کرده استخراج کرد.
- در هنگام تفکیک UC کاری به چگونگی اجرا نداریم و آنچه که باید باشد را در نظر می‌گیریم.
- تعداد UC نباید بیش از حد باشد.(بین ۲۰ تا ۵۰ خوب است)
- نامگذاری UC باید از دیدگاه تجاری باشد و نه تکنیکی.(معمولا فعل یا اصطلاحات شامل فعل می‌باشد)
- مستندسازی جریان رخدادها: مستند کاملی از کلیه رخدادهایی که در سراسر UC باید اتفاق بیافتد را باید مکتوب کرد. این مستند شامل آنچه که کاربر سیستم و نیز خود سیستم انجام می‌دهد می‌باشد. در این قسمت هم کاری به زبان پیاده‌سازی نخواهیم داشت.
- **۳-۲-۱) تعریف(Description):** شامل تعریف کوتاه و مختصری است از آنچه که UC انجام می‌دهد.
- **۳-۲-۲) پیش شرایط(Precondition):** لیستی از شرایطی است که قبل از شروع کار UC باید بررسی شوند. بطور مثال اینکه آیا یک UC دیگر اجرا شده یا کاربر حق دستیابی مستقیم را دارد یا خیر. از این قسمت میتوان برای نشان دادن ترتیب اجرای UCها استفاده کرد. این قسمت الزامی نیست.
- **۳-۲-۳) جریان رخداد اصلی و فرعی:** مشخصات کامل UC و آنچه که در اجرای عملیات در UC پیش خواهد آمد در جریان رخداد اصلی و فرعی تعریف می‌شود. این جریان مشتمل است بر:
- چگونگی شروع به کار UC
- مسیرهای متنوع منتهی به UC
- در طول UC هر گونه انحرافی از جریان رخداد اصلی به عنوان روز فرعی شناخته می‌شود.
  - هر روند دارای خطا
- چگونه UC به پایان می‌رسد.
  - می‌توان جریان رخدادها را بصورت لیست‌های شماره‌دار، متون پاراگراف بندی شده، لیست‌های بالت شده و یا فلوچارت نشان داد.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- هنگام نوشتن جریان رخداد باید از جزئیات چگونگی کار دوری کرد.
- تمرکز باید بر روی اطلاعات رد و بدل شده بین کاربر و سیستم و نحوه چگونگی عملکرد سیستم باشد.
- **۳-۲-۴) شرایط پسین(Post Conditions):** شرطهایی هستند که باید بعد از اجرای UC ست باشند. برای حفظ ترتیب اجرای UCها می‌توان از این شرایط استفاده کرد. این شرطها الزامی نیستند.

۳-۳) افزودن UCموجود به نمودار UC: برای این‌کار بجز روش معمول می‌توان از آیتم... Add Use Cases از منوی Queryهم استفاده کرد.

۳-۴) مشخصات(Specification) یک UC

۳-۴-۱) تخصیص اولویت به یک UC: اولویت درجه اهمیت UC را نشان می‌دهد.

۳-۴-۲) ساختن UCهای مجرد(Abstract): UC مجرد آنی است که مستقیما توسط یک عامل شروع به کار نمی‌کند و در عوض برخی عملیات اضافی که می‌تواند توسط UCهای دیگر استفاده شود را فراهم می‌کند.(آنهایی که مورد Include یا Extend واقع می‌شوند)

۳-۴-۳) مشاهده نمودارها برای یک UC: می‌توان هر یک از نمودارهای پنجگانه Use Case، توالی، همکاری، کلاس و حالت را ذیل یک UC تعریف کرد.

۳-۴-۴) مشاهده شرکت کنندگان در یک UC: شامل کلاسها، عملیات یا کامپوننت‌هایی که در یکی از دیدگرام‌های ذیل UC تعریف شده‌اند. برای مشاهده شرکت کنندگان در یک UC آیتم ... Show Participants in UC را از منوی Report انتخاب می‌کنیم.

۳-۴-۵) مشاهده رابطه‌های(Relations) متعلق به یک UC: رابطه‌ها عبارت است از رابطه‌های بین یک UC با UCها یا عامل‌های دیگر.

- برای دیدن رابطه(در هر دو طرف) از برگه Relationدر پنجره مشخصات استفاده شود.
- برای دیدن رابطه(یک طرفه) از آیتم Show Usage از منوی Report استفاده شود.

۳-۵) کار با عامل(Actor): هر کس یا هر چیزی که با سیستم تعامل داشته باشد عامل نامیده می‌شود. سه نوع اصلی از عامل‌ها وجود دارد که عبارتند از:

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- کاربران سیستم که یک انسان است و بیشتر عامل‌ها از این نوع هستند. برای نام‌گذاری بهتر است از نام نقش استفاده شود تا نام موقعیت.
- یک سیستم یا UC دیگر. این سیستم باید خارج از سیستم ما باشد و امکان هیچ تغییری را هم در آن نخواهیم داشت.
- زمان. چون زمان خارج از کنترل ماست پس یک عامل است. هنگامی زمان تبدیل به یک عامل می‌شود که زمان در حال گذر باعث ایجاد رخدادی در سیستم گردد مانند اینکه ATM در نیمه شب بخواهد یک پردازش خاصی انجام دهد.

۳-۵-۱) مشخصات عامل: شامل صفات ویژه یک عامل می‌باشد. با توجه به اینکه عامل نوع خاصی از کلاس است پنجره مشخصات عامل با کلاس مشابه است. برخی از برگه‌های موجود در پنجره مشخصات خاص کلاس‌ها هستند و برای عامل کاربردی ندارند.(فقط برگه‌های Relations, Detail, General و Files)

- **تنظیم کاردینالیتی برای یک عامل:** مشخص می‌کند به چند نمونه از عامل خاصی نیاز داریم. انواع کاردینالیتی عبارتند از: n، 0..0، 0..1، 0..n، 1..1، 1..n . می‌توان دو کاردینالیتی را با هم تلفیق کرد مثلا: 1..3 و 5 . جایگاه این صفت در برگه Detail می‌باشد.
- **عامل مجرد:** عاملی است که هیچ مصداق واقعی ندارد و کاردینالیتی آن صفر است و کاربرد آن در ارث بری است.(مثلا کارمند از نوع مجرد است در مقابل کارمند ساعتی) جایگاه این صفت در برگه Detail است.
- **مشاهده رابطه‌های متعلق به یک عامل:** شامل رابطه‌هایی است که یک عامل با UCها یا عامل‌های دیگر دارد.
- **مشاهده یک نمونه عامل :** میتوان لیستی از نمودارهای تعامل را پیدا کرد که این عامل در آنها بکار رفته است. این عمل با استفاده از آیتم Show Instances یا Show Usage از منوی Report انجام می‌شود.

۳-۶) چگونگی کار با رابطه‌ها: رابطه بین UCها و عامل‌ها انواع مختلفی دارد که جداگانه به هر یک می‌پردازیم.

۳-۶-۱) رابطهAssociation : رابطه‌ای است که بین UC و عامل برقرار می‌شود.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- ابتدای فلش نشان دهنده کسی است که ارتباط از آنجا شروع می‌شود، تبادل اطلاعات در هر دو جهت می‌باشد. فلش می‌تواند از عامل به UC یا بر عکس باشد.
- برای ایجاد این ارتباط از آیکون Unidirectional Association استفاده می‌شود.

۳-۶-۲) رابطه Include: این نوع از رابطه به یک UC اجازه استفاده از عملیات مهیا شده توسط یک UC دیگر را می‌دهد. مثلا برداشت پول از حساب و واریز پول به حساب هر دو نیاز UC مربوط به شناسایی مشتری و شماره PIN آن دارد.

جهت فلش از UC استفاده کننده به UC استفاده شونده است.

UC که استفاده می‌شود از نوع مجرد خواهد بود.(معقول‌تر است)

برای ایجاد این ارتباط از آیکون Dependency استفاده می‌شود.

نوع ارتباط با استفاده از صفت Stereotype و نوشتن یا انتخاب کردن Include مشخص می‌شود.

برای اینکه عنوان "Include" بالای نام رابطه نشان داده شود بایستی Stereotype Label در منوی کلیک راست چک خورده باشد.

۳-۶-۳) رابطه‌های Extend: مشابه رابطه Include بوده و به یک UC اجازه میدهد عملیات مهیا شده توسط UC دیگر را بسط دهد. مثلا برداشت پول از حساب می‌تواند با بسط برداشت سریع از حساب حاصل شود.

جهت فلش از UC استفاده شونده به UC استفاده کننده(بسط دهنده) می‌باشد.

UC که استفاده می‌شود از نوع مجرد خواهد بود.(معقول‌تر است)

برای ایجاد این ارتباط از آیکون Dependency استفاده می‌شود.

نوع ارتباط با استفاده از صفت Stereotype و نوشتن یا انتخاب کردن Extend مشخص می‌شود.

برای اینکه عنوان "extend" بالای نام رابطه نشان داده شود بایستی Label Stereotype در منوی کلیک راست چک خورده باشد.

۳-۶-۴) ارتباط Actor Generalization: ارتباط وراثتی بین عامل‌ها را نشان می‌دهد.

- در صورتیکه دو عامل با UCهای یکسانی ارتباط داشته باشند ارزشی ندارد که از این ارتباط استفاده کنیم.
- عاملی که مجرد است و مورد ارث بری واقع شده در انتهای فلش قرار می‌گیرد.

برای ایجاد ارتباط از آیکون Generalitation استفاده می‌شود.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

محاورات آبجکتی

نمودارهای تعامل در این فصل بررسی می‌شوند. هر دو نوع این نمودارها آبجکت‌های شرکت کننده در روند یک UC و پیغام‌هایی که بین آبجکت‌ها رد و بدل می‌شوند را نشان می‌دهند. نمودارهای توالی بر حسب زمان مرتب می‌شوند در حالیکه نمودارهای همکاری خودشان را در اطراف آبجکت‌ها سازماندهی می‌کنند.

۴-۱) نمودارهای تعامل: برای هر یک از رخدادهای اصلی و فرعی UC یک نمودار تعامل رسم می‌شود. هر نمودار باید دارای یک عاملی(همانی که در نمودار UC می‌آید) باشد که کار را شروع کند.

- آبجکت آن چیزی است که اطلاعات و روش‌ها را در خود کپسوله می‌کند.

بخش‌های اطلاعاتی که توسط آبجکت نگهداری می‌شود صفات یا Attribute نامیده می‌شوند.

رفتارهای یک آبجکت به عنوان عملیات یا Operation نامیده می‌شود.

- کلاس آن چیزی است که طرح کلی آبجکت را فراهم می‌کند.

۴-۱-۱) یافتن آبجکت‌ها: چطور می‌توان آبجکت را برای یک نمودار تعامل پیدا کرد؟

- یک راه خوب در نظر گرفتن نام‌ها در جریان رخدادها است.
- یک جای خوب دیگر سناریوی اسناد می‌باشد. سناریو حالت خاصی از جریان رخدادهاست. در واقع هر نمودار تعامل یکی از این سناریوها را شرح می‌دهد. ممکن است بعضی از اسامی صفت باشد(بجای آبجکت)، باید دید آیا رفتار هم دارد یا فقط اطلاعاتی را نگهداری می‌کند.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- همه آبجکت‌ها در جریان رخدادها وجود نخواهند داشت(مانند فرم‌ها) ولی باید در نمودار ظاهر شوند. آبجکت‌های کنترل هم در جریان رخدادها وجود ندارند ولی باید در نمودار بیایند.

۴-۲) نمودارهای توالی: هر نمودار توالی یک روند را در در UC نشان می‌دهد.

- عامل معمولا اولین آبجکت نمودار است که در قسمت بالا سمت چپ ظاهر میشود.
- هر آبجکت یک خط عمر دارد که بصورت خطوط عمودی خط چین در زیر آن کشیده می‌شود
- یک پیغام بین دو خط عمر قرار داده می‌شود تا ارتباط بین آبجکت‌ها را نشان دهد.

هر پیغام نشان دهنده این است که یک آبجکت توسط آبجکت دیگر صدا زده شده است و تبدیل به یک عملیات(Operation) می‌شود.
پیغام‌ها می‌توانند بازتابی(Reflexive) باشند که نشان دهنده این است که آبجکت یکی از عملیات‌های خودش را صدا زده است.

۴-۳) نمودارهای همکاری: مانند نمودار توالی است بجز اینکه تمرکز بیشتر روی آبجکت وجود دارد. این نمودار به راحتی به نمودار توالی تبدیل می‌شود(کلید F۵ یا آیتم Create Collaboration Diagram از منوی Browse) در واقع هر تغییری در یک نمودار تعامل در دیگری نیز منعکس می‌شود.

۴-۴) کار با آبجکت‌ها

۴-۴-۱) نگاشت(Assign) یک آبجکت به یک کلاس: میتوان آبجکت را به یک کلاس موجود نگاشت کرد یا یک کلاس جدید ایجاد کرد. برای اینکار میتوان کلاس را کشید و روی آبجکت رها کرد یا در پنجره مشخصات آبجکت فیلد Class از برگه General را تنظیم کرد.

- نام کلاس در آیتم آبجکت و بعد از کولن می‌آید.

برای اطمینان یافتن از اینکه همه آبجکت‌ها به کلاس نگاشت شده‌اند از آیتم Show Unresolved object از منوی Report استفاده می‌شود.

۴-۴-۲) تنظیم پایداری(Persistence) آبجکت: ۳ گزینه برای پایداری وجود دارد.

- **Persistent:** آبجکتی است که در یک بانک اطلاعاتی و یا هر مخزن پایدار دیگر نگهداری می‌شود. با از بین رفتن برنامه این آبجکت از بین نمی‌رود.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- **Static:** تا زمانیکه برنامه از بین نرفته وجود خواهد داشت(ممکن است پشت صحنه باشد)
- **Transient:** موقتی است و برای مدت کوتاهی(تا زمانیکه نمودار توالی وجود داشته باشد) در حافظه باقی می‌ماند. این صفت را در پنجره مشخصات فیلد Persistence می‌توان تنظیم کرد.

۴-۴-۳) استفاده از نمونه‌های چندگانه یک آبجکت: به معنی لیستی از آبجکت(مثلا لیستی از کارمندان) بجای یک آبجکت است. در این حالت نماد آبجکت عوض می‌شود. برای چندگانه کردن یک آبجکت در پنجره مشخصات گزینه Multiple Instances باید چک شود.

۴-۵) کار با پیغام‌ها: یک پیغام برقرار کننده ارتباط بین آبجکت‌ها است که در آن آبجکت درخواست کننده از آبجکت توزیع کننده تقاضای انجام کاری را می‌کند. نهایتا یک پیغام به یک تابع(Operation) تبدیل می‌شود.

۴-۵-۱) افزودن پیغام به یک نمودار توالی: ابتدا باید آیکون Object Message را انتخاب کرد و از یک خط عمر به خط عمر دیگر کشید. پیغام‌ها از بالا به پایین به ترتیب زمانی نشان داده می‌شوند و میتوان ترتیب آنرا تغییر داد.

- برای اضافه کردن یک پیغام بازتابی باید آیکون Message to Self را انتخاب کرد و روی یک خط عمر کلیک کرد.
- برای اینکه شماره‌گذاری پیغام‌ها را فعال یا غیر فعال کنیم میتوان از منوی Tools، آیتم ... Options(که از این به بعد با عنوان پنجره امکانات یاد می‌شود)، برگه Diagram، آیتم Sequence Numbering استفاده کرد.
- میتوان مرکز کنترل(مستطیل کوچکی در انتهای پیغام) را فعال یا غیر فعال کرد. این مستطیل برای نشان دادن این است که در یک زمان مشخص کدام آبجکت کنترل را در دست گرفته است. برای فعال یا غیر فعال کردن آن باید در پنجره امکانات برگه Diagram، از آیتم Focus of Control استفاده کرد.

۴-۵-۲) افزودن پیغام به نمودار همکاری: قبل از اینکه پیغام‌هایی را به نمودار همکاری اضافه کنیم باید یک مسیر ارتباطی را بین دو آبجکت برقرار کنیم.

- مسیر ارتباطی لینک نامیده شده و با استفاده از آیکون Object Link ایجاد می‌شود.
- برای افزودن پیغام باید از آیکون Link Message یا Reverse Link Message استفاده کرد و روی مسیر ارتباطی کلیک کرد.

Info@IRANMEET.COM
www.pupuol.com

- برای افزودن پیغام بازگشتی باید از آیکون Link to self و کلیک روی آبجکت استفاده کرد.
- برای فعال یا غیر فعال کردن شماره‌گذاری پیغام‌ها در نمودار همکاری میتوان در پنجره امکانات برگه Diagram از Collaboration Numbering استفاده کرد.

۴-۵-۳) افزودن جریان داده‌ای(Data Flow) به نمودار همکاری: جریان‌های داده‌ای برای نشان دادن اطلاعات برگشتی وقتی که آبجکتی به آبجکت دیگر پیغام ارسال می‌کند بکار میرود. این عنصر در نمودارهای توالی استفاده نمی‌شود.

- استفاده بی‌مورد از این عنصر فقط باعث شلوغی نمودار می‌گردد.
- هر جریان داده‌ای باید همراه یک پیغام بیاید.
- برای اضافه کردن جریان داده‌ای از آیکون Data Flow یا Reverse Data Flow استفاده کرده و روی پیغام کلیک می‌کنیم.

۴-۵-۴) مشخصات پیغام(Message Specification)

- هنگام نامگذاری اولیه باید طوری عمل شود که بعدها راحت تبدیل به نام تابع شود.
- در مستندات می‌توان از شبه کدها(Pseudo-Code) که عملکرد تابع را نشان می‌دهند استفاده کرد. مستندسازی پیغام در تولید کد ظاهر نمی‌شود ولی مستندات تابع در تولید کد ظاهر می‌شود.

۴-۵-۵) نگاشت یک پیغام به یک تابع: قبل از اینکه کدسازی کنیم باید تمام پیغام‌ها به توابعی از کلاس نگاشت شود.(تا زمانیکه آبجکت‌ها را به کلاسها نگاشت نکرده باشیم این مرحله امکان‌پذیر نخواهد بود)

- برای نگاشت پیغام به تابع روی پیغام کلیک راست کرده و از منوی میانبر نام تابع را انتخاب کرده یا از طریق آیتم new operation تابع جدیدی را ایجاد کرده و نگاشت می‌کنیم.
- برای حذف تابع نگاشت شده کافی است پنجره مشخصات را باز کرده و نام تابع را حذف کنیم.
- برای اطمینان از اینکه هر پیغام به یک تابع نگاشت شده از آیتم Show Unresolved Message از منوی Report استفاده می‌کنیم.

۴-۵-۶) تنظیم گزینه‌های همزمان سازی پیغام: پنج گزینه برای همزمان سازی وجود دارد که عبارتست از: (برای تنظیم گزینه همزمان سازی از برگه Detail در پنجره مشخصات استفاده می‌شود)

- **Simple:** مقدار پیش فرض است و مشخص می‌کند پیغام در یک جهت از کنترل اجرا می‌شود.

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- **Synchronous:** بدین معنی است که درخواست کننده پیغام را فرستاده و صبر می‌کند تا توزیع کننده نسبت به پیغام واکنش نشان دهد.
- **Balking:** بدین معنی است که درخواست کننده پیغام را می‌فرستد و اگر توزیع کننده آماده دریافت پیغام نبود روند ارسال پیغام متوقف می‌شود.
- **Timeout:** در این حالت درخواست کننده مدت کوتاهی بعد از ارسال پیغام صبر می‌کند و اگر در این مدت توزیع کننده آماده دریافت نبود ارسال پیغام را متوقف می‌کند.
- **Asynchronous:** با انتخاب این گزینه درخواست کننده پیغام را می‌فرستد و بدون اینکه صبر کند آیا پیغامی دریافت شده یا خیر به پردازش خویش ادامه می‌دهد.

۴-۵-۷) تنظیم تکرار پیغام: به ما اجازه میدهد پیغام‌هایی را که مرتبا در فواصل زمانی فرستاده می‌شوند تعیین کنیم. دو گزینه برای این قسمت وجود دارد که عبارتست از: (برای تنظیم گزینه تکرار از برگه Detail از پنجره مشخصات پیغام استفاده می‌کنیم)

- **Periodic:** به معنی این است که پیغام به طور مرتب در فواصل زمانی معین فرستاده شود.
- **Aperiodic:** این گزینه نشان میدهد که پیغام ممکن است فقط یکبار فرستاده شده یا در زمان‌های نامعینی فرستاده شود.

۴-۶) کار با اسکریپت‌ها: در زمانی که یادداشت برای افزودن توضیحی به آبجکت استفاده می‌شود، اسکریپت توضیح را به پیغام اضافه می‌کند. اسکریپت‌ها فقط در نمودارهای توالی استفاده می‌شوند و در سمت چپ نمودار روبروی پیغام مربوطه ظاهر می‌شوند.

- از اسکریپت می‌توان برای نشان دادن معنی یک پیغام یا ورود برخی منطق‌های شرطی به نمودار استفاده کرد.
- اسکریپت کدی را تولید نمی‌کند.
- برای اضافه کردن اسکریپت ابتدا باید با استفاده از آیکون ABC یک کادر متن ایجاد کرد سپس این کادر متن و پیغام مربوطه را همزمان انتخاب کرد(با استفاده از کلید Shift) و سپس از منوی Edit آیتم Attach Script را انتخاب کرد.
- برای جدا کردن اسکریپت از پیغام کافی است اسکریپت را انتخاب کرد و سپس از منوی Edit آیتم Detach Script را انتخاب کرد.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- برای نشان دادن IF های پیچیده و تو در تو بجای استفاده از اسکریپت بهتر است نمودارها را از هم جدا کنیم.

۴-۷) سوئیچ کردن بین نمودارهای توالی و همکاری: در Rose کافی است یکی از آنها را بسازیم و دیگری را خود Rose برای ما می‌سازد. برای سوئیچ کردن از کلید F۵ یا آیتم Create Collaboration Diagram از منوی Browse استفاده می‌کنیم.

۴-۸) روش‌های دو گذره(Two-Pass) برای نمودارهای تعامل: ساخت نمودارهای تعامل معمولا دو گذر را طی می‌کند که عبارتند از:

۴-۸-۱) مسیر اول(گذر اول): که در طی آن تمرکز اصلی بر روی اطلاعات سطح بالایی است که مشتریان با آنها درگیر هستند و هنوز پیغام‌ها به عملیات و آبجکت‌ها به کلاس‌ها نگاشت نشده‌اند. این نمودارها فقط به تحلیل‌گرها، مشتریان و تمام کسانی که با روند تجاری سیستم در ارتباط هستند چگونگی جریان منطقی درون سیستم را نشان می‌دهد.

۴-۸-۲) گذر دوم: که وقتی مشتری خودش را با روند گذر اول وفق داد تیم جزییات بیشتری را به نمودار اضافه می‌کند. در این نقطه ممکن است نمودار حالت مفید خود را برای مشتری از دست داده باشد ولی برای برنامه نویسان و تست کننده حالت مفیدتری پیدا خواهد کرد.

معمولا هر نمودار تعامل دارای آبجکت کنترل(مسئول کنترل تناوب در طول پروژه) می‌باشد. ممکن است این آبجکت بین تمام نمودارهای تعامل مربوط به یک UC به اشتراک گذارده شود.

آبجکت کنترل در واقع نقش واسط و انتقال پیغام از آبجکتی به آبجکت دیگر را دارد. آبجکت کنترل مسئول هماهنگ سازی کارهای آبجکت‌های دیگر و تعیین و محول کردن مسئولیت‌ها به آبجکت‌هاست به همین دلیل گاهی اوقات گاهی آبجکت‌های مدیر نامیده می‌شوند.

مزیت وجود آبجکت‌های کنترل در این است که در صورت ثابت ماندن منطق تجاری و با تغییر منطق تناوب تنها کافی است آبجکت کنترل را تغییر دهیم. شبیه به آبجکت‌های کنترل آبجکت‌های با کاربرد امنیت، تحت کنترل نگه داشتن خطاها و یا اتصال به پایگاه داده را می‌توان یک بار ساخته و به تعدّد استفاده کرد.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

کلاسها و بسته‌ها

۵-۱) نمودارهای کلاس: یک نمودار کلاس برای نمایش تعدادی از کلاس‌ها و بسته‌های کلاس بکار میرود. این نمودار یک تصویر ایستا از قطعات سیستم و ارتباطات بین آنها را به ما می‌دهد.

- بطور پیش فرض نمای منطقی و هر بسته‌ای که در این نما ایجاد شود یک نمودار کلاس اصلی با نام Main دارد.
- در نمودار اصلی نمای منطقی معمولا بسته‌ها ظاهر میشوند و کلاسها در نمودار اصلی بسته‌ها می‌آیند.

۵-۲) یافتن کلاسها: یک محل خوب برای پیدا کردن کلاسها جریان رخدادهای UC است. وقتی به اسامی نگاه می‌کنیم یکی از چهار چیز خواهند بود:

- عامل
- کلاس
- یک صفت از کلاس
- یک اصطلاح که عامل، کلاس یا صفت نیست

۵-۳) ساختن نمودارهای کلاس

۵-۳-۱) سازماندهی عناصر مدل روی یک نمودار کلاس

- با استفاده از آیتم Layout Diagram از منوی Format می‌توان عناصر مدل را مرتب کرد.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- با استفاده از آیتم All Autosize از منوی Format می‌توان اندازه عناصر روی نمودار را به اندازه لازم برای نام کلاس، صفات و عملگر بصورت خودکار در آورد.

۵-۴) کار با کلاسها: انواع کلاسها عبارتند از:

- معمولی(Regular)
- پارامتری شده(Parametrited)
- نمونه (Instantiated)
- یوتیلیتی(Utility)
- Parametrited Utility
- Instantiated Utility
- Meta

۵-۵) افزودن کلاسهای معمولی: علاوه بر روش‌های معمول ساخت عناصر مدل از طریق نمودارهای تعامل در هنگام نگاشت کلاس به آبجکت هم می‌توان کلاس جدید ساخت.

۵-۶) مشخصات کلاس: در نامگذاری کلاس بهتر است استانداردی وجود داشته باشد. شامل فضای خالی نبوده و زیاد طولانی نباشد.

۵-۶-۱) انواع طبقه‌بندی کلاس: در UML سه نوع اساسی برای طبقه‌بندی کلاس وجود دارد که هر یک نماد مجزایی دارند و عبارتند از:

- کلاسهای Boundary: کلاسهایی هستند که روی حاشیه بین سیستم ما و جهان خارج قرار می‌گیرند. فرمها، گزارشها، واسطهای سخت افزاری از قبیل چاپگرها، اسکنرها و... مثالهایی از این طبقه از کلاسها هستند. یک راه برای پیدا کردن کلاسهای Boundary محل اتصال عامل با UC در نمودار UC است. یک کلاس Boundary در موارد متعدد می‌تواند نقش فوق را باز می‌کند.
- کلاسهای Entity: این کلاسها شامل اطلاعاتی هستند که آنها را در انباری طولانی ذخیره خواهیم کرد. این کلاسها معمولا در جریان رخدادها و رسم نمودارهای تعامل پیدا می‌شوند. معمولا متناظر با یک کلاس از نوع Entity یک جدول در بانک اطلاعاتی ساخته می‌شود.
- کلاسهای Control: این نوع از کلاس مسئول هماهنگ کردن تلاش‌های کلاس‌های دیگر هستند. بطور ویژه برای هر UC یک کلاس کنترل وجود دارد که دنباله‌ای از رخدادها را در طول UC

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

کنترل می‌کند. کلاسهای کنترل توابع زیادی برای صدا زدن ندارند در عوض خودشان توابع زیادی از کلاسهای دیگر را صدا می‌زنند. ممکن است کلاسهای کنترلی داشته باشیم که در میان چند UC به اشتراک گذاشته شده باشند مثل کلاس Security Manager یا Transaction Manager.

۵-۶-۱-۱) نمایش آیکون نوع کلاس

- برای اینکه طبقه‌بندی بصورت متن و بین << >> بیاید کافی است روی کلاس کلیک راست کرده و از منوی Options، منوی Sterotype Display، آیتم Label انتخاب شود.
- اگر بخواهیم کلاس را در نمودار به شکل آیکون خاص هر طبقه نشان دهد می‌بایستی از منوی اخیر آیتم Icon انتخاب شود.
- برای غیر فعال نمودن نمایش Stereotype از نام کلاس بایستی از منوی اخیر آیتم None را انتخاب کرد.
- برای اینکه بگوییم در کل مدل آیا Stereotype نمایش داده شود یا خیر باید در پنجره امکانات، برگه Diagram، آیتمShow Stereotypes را فعال کنیم.
- می‌توان یک طبقه جدید، آیکون جدید برای نمودار، آیکون دکمه کوچک جدید برای نوار ابزار، آیکون بزرگ جدید برای نوار ابزار و آیکون جدید برای مرورگر را برای نوع خاصی کلاس ایجاد کرد به نحوی که بتوان در تمامی مدل‌های Rose از آن استفاده کرد. برای اطلاع از جزئیات چگونگی انجام مراحل لازم به صفحات ۲۷۴-۲۶۹ از کتاب مرجع کامل UML مراجعه شود.

۵-۶-۲) تنظیم Visibility کلاس: این گزینه محدوده دیده شدن کلاس توسط کلاسهای دیگر را معین می‌کند و ۲ گزینه برای آن وجود دارد.

- **Public:** کلاس به وسیله همه کلاسهای دیگر سیستم دیده می‌شود.
- **Implementation:** فقط توسط کلاسهای تعریف شده در همان بسته دیده می‌شود.

۵-۶-۳) تنظیم کاردینالیتی کلاس: تعداد نمونه‌هایی از کلاس که انتظار داریم ایجاد شود را معین می‌کند. حالتهای این گزینه شبیه گزینه مشابه برای عامل می‌باشد.

۵-۶-۴) تنظیم حافظه مورد نیاز: می‌توان حافظه فیزیکی نسبی یا مطلقی که کلاس نیاز دارد را ذخیره کرد. این گزینه در پنجره مشخصات کلاس، برگه Detail، فیلد Space قرار دارد.

Info@IRANMEET.COM
www.pupuol.com

WWW.IRANMEET.COM

۵-۶-۵) تنظیم پایداری(Persistence) کلاس: کاربرد این گزینه زمانی است که بخواهیم از مدلمانDDL(Data Definition language) را تولید کنیم. با توجه به اینکه DDL ساختار پایگاه داده را تعریف می‌کند این گزینه توسط Rose بررسی می‌شود. این گزینه را نمی‌توان برای کلاسهای از نوع Utility(همه انواع آن) بکار برد. انواع پایداری عبارتست از:

- **Persistent:** پیشنهاد می‌کند کلاس خارج از اجرای برنامه زندگی کند به عبارت دیگر آبجکت‌های کلاس در یک پایگاه داده ذخیره شود.
- **Transient:** پیشنهاد می‌کند آبجکت‌های کلاس در انباره دائمی ذخیره نشود. این حالت پیش فرض است.

۵-۶-۶) تنظیم همروندی(Concurrency) کلاس: توضیح می‌دهد کلاس در حضور چند رشته(Thread) اجرایی چگونه رفتار می‌کند. انواع آن عبارتند از: (تنظیم همروندی از طریق برگه Detail در پنجره مشخصات انجام می‌شود)

- **Sequential:** پیش فرض است و نشان میدهد کلاس به طور نرمال رفتار می‌کند. زمانیکه چند رشته وجود داشته باشند و بخواهند عملگرهای این کلاس را صدا بزنند رفتار آن ضمانت شده نیست.
- **Guarded:** در حالت چند رشته‌ای نیاز است خود کلاس‌ها با یکدیگر همکاری کنند تا مطمئن شوند برخوردی پیش نمی‌آید.
- **Active:** پیشنهاد می‌کند که کلاس رشته خودش را داشته باشد.
- **Synchronous:** در حالت چند رشته‌ای هم رفتار کلاس تضمین شده است و نیازی به همکاری با دیگر کلاس‌ها وجود ندارد. حل مشکل کلاس با خودش از طریق حذف مشترک(Mutual Exclusion) انجام می‌شود.

۵-۶-۷) ساختن یک کلاس مجرد: کلاس مجرد کلاسی است که هیچ نمونه‌ای نخواهد داشت. کاربرد این کلاس در ساختارهای وراثتی است و آن اطلاعات و رفتاری را نگه می‌دارد که در دیگر کلاسها مشترک است.

- نام کلاس مجرد بصورت ایتالیک نمایش داده می‌شود.
- تنظیم گزینه در برگه Detail از پنجره مشخصات انجام می‌شود.

۵-۷) استفاده از کلاس‌های تو در تو(Nested): می‌توان یک کلاس را داخل کلاس دیگر قرار داد، هم چنین می‌توان کلاسهای ضمیمه را داخل کلاس تو در تو قرار داد.

- تعریف کلاسهای تو در تو در برگه Nested از پنجره مشخصات و با استفاده از منوی کلیک راست و Insert صورت می‌گیرد.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- نام کلاس داخلی که به نمودار اضافه می‌شود با یک توضیح داخل پرانتز که نشان میدهد در چه کلاسی قرار دارد همراه می‌شود.

۵-۸) دیدن نمودارهای تعامل شامل یک کلاس: برای دیدن لیستی از نمودارهای تعامل که یک کلاس خاصی در آنها بکار رفته است بعد از فعال کردن کلاس مربوطه از آیتم Show Instances از منوی Report استفاده می‌کنیم.

۵-۹) کار با بسته‌ها: بسته‌بندی کلاسها دلخواه است ولی یکراه مفید می‌تواند دسته‌بندی بر اساس طبقه کلاس باشد. روش مفید دیگر گروه‌بندی بر اساس کارایی است مثل گروه امنیت.(Security) بهترین بسته‌بندی(و بسته‌بندی تو در تو) آن است که قابلیت استفاده مجدد را به حداکثر برساند.

۵-۱۰) انواع ویژه کلاس

۵-۱۰-۱) کلاس پارامتری شده(Parametrized): کلاسی است که برای ساختن یک خانواده از کلاس‌های دیگر استفاده شده است. مخصوصا کلاس پارامتری شده تعداد مرتبی از ظرف(Container) است و هم چنین به عنوان یک الگو شناخته می‌شود. مثلا List می‌تواند یک کلاس پارامتری شده باشد که با استفاده از آن بتوان انواع کلاسهای لیست را ساخت.

- در Rose کلاس پارامتری شده با استفاده از یک مستطیل خط چین شده در قسمت بالا سمت راست کلاس ظاهر می‌شود.
- برای افزودن کلاس پارامتری شده می‌توان از آیکون مربوطه در نوار ابزار استفاده کرد یا فیلد Type در پنجره مشخصات کلاس را به Parametrized Class تغییر داد.

۵-۱۰-۱-۱) تنظیم آرگومان‌ها برای یک کلاس پارامتری شده: پارامتری را مشخص می‌کند که از کلاس پارامتری شده لیست‌های جدید بر اساس تفاوت بین مقادیر آرگومان‌ها می‌سازد.

- برای اضافه کردن آرگومان در برگه Detail از پنجره مشخصات در قسمت Formal Argument باید از منوی میانبر Insert استفاده کرد. آرگومان اضافه شونده دارای نام، Type و مقدار پیش فرض می‌باشد.
- آرگومان‌های اضافه شده در کادر مستطیل خط چین شده ظاهر می‌شوند.

۵-۱۰-۲) کلاس نمونه(Instantiated Class): کلاس نمونه یک کلاس پارامتری شده است که مقادیر واقعی را برای آرگومان‌ها دارد. با مشخص

Info@IRANMEET.COM
www.pupuol.com

شدن یک مقدار مشخص برای آرگومان‌ها، یک لیست خاصی(مثلا لیست کارمندانی که سن آنها ۳۰ سال است) ساخته می‌شود.

- عنوان کلاس نمونه در نمودار کلاس ما بین< >واقع می‌شود.
- برای اضافه کردن یک کلاس نمونه می‌توان آیکون مربوطه را از نوار ابزار انتخاب کرد یا فیلد Type از یک کلاس معمولی اضافه شده را به Instantiated Class تغییر داد.
- تنظیم آرگومان‌ها مشابه همین مبحث در کلاسهای پارامتری شده است.

۵-۱۰-۳) ابزار کلاس(Class Utility): یک ابزار کلاس مجموعه‌ای از عملگرهاست. مثلا ممکن است تعدادی توابع ریاضی داشته باشیم که در طول برنامه استفاده شده باشد. می‌توان این توابع را جمع کرده و در یک ابزار کلاس قرار داد تا توسط کلاسهای دیگر استفاده شود.

- ابزار کلاس بصورت یک کلاس سایه‌دار در نمودار کلاس نمایش داده می‌شود.
- برای اضافه کردن ابزار کلاس می‌توان آیکون مربوطه را از نوار ابزار انتخاب کرده و یا فیلد Type در یک کلاس معمولی اضافه شده را به Class Utility تغییر داد.

۵-۱۰-۴) ابزار کلاس پارامتری شده(Parametrized Class Utility):
این نوع از کلاس یک کلاس پارامتری شده است که شامل مجموعه‌ای از عملگرهاست.

- شکل ابزار کلاس پارامتری شده روی نمودار تلفیقی از کلاس پارامتری شده و ابزار کلاس یعنی دارای یک کادر مستطیل خط چین و دارای سایه می‌باشد.
- برای اضافه کردن یک ابزار کلاس پارامتری شده می‌توان آیکون مربوطه را از نوار ابزار انتخاب کرده و یا فیلد Type در یک کلاس معمولی اضافه شده را به Parametrized Class Utility تغییر داد.
- تنظیم آرگومان در این نوع از کلاس مشابه همین مبحث در کلاس‌های پارامتری شده است.

۵-۱۰-۵) ابزار کلاس نمونه(Instantiated Class Utility): عبارت است از یک ابزار کلاس پارامتری شده که مجموعه‌ای از مقادیر واقعی را برای آرگومان‌ها(پارامترها) دارد.

- شکل ابزار کلاس نمونه در نمودار تلفیقی از شکل ابزار کلاس و کلاس نمونه است یعنی عنوان کلاس میان< > آمده و سایه‌دار است.

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- برای اضافه کردن یک ابزار کلاس نمونه می‌توان آیکون مربوطه را از نوار ابزار انتخاب کرده و یا فیلد Type از یک کلاس معمولی اضافه شده را به Instantiated Class Utility تغییر داد.
- تنظیم آرگومان شبیه همین مبحث در کلاس‌های پارامتری شده است.

۵-۱۰-۶) کلاس متا(Meta Class): کلاسی است که نمونه‌های آن بیشتر از آنکه آبجکت باشند از نوع کلاس هستند. کلاس‌های متا یکسری عملیات(توابع) را برای مقدار دهی اولیه متغیرها فراهم می‌کنند و بعنوان انبانی برای نگهداری متغیرهای کلاسی که یک مقدار واحد را برای تمامی آبجکت‌های کلاس ت أمین می‌کنند، استفاده می‌شود. تمامی زبانهای برنامه‌نویسی مستقیما این نوع از کلاس را پشتیبانی نمی‌کنند.

برای اضافه کردن یک کلاس متا می‌توان آیکون مربوطه را از نوار ابزار انتخاب کرده و یا فیلد Type از یک کلاس معمولی اضافه شده را به Meta Class تغییر داد.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

صفات(Attributes) و عملیات(Operations)

۶-۱) یافتن صفات: راه‌های مختلفی برای یافتن صفات وجود دارد که بعضی از آنها عبارتند از:

- می‌توانیم به مستندسازی UC نگاهی بیندازیم و در جریان رخدادها اسامی را پیدا کنیم. برخی از اسامی می‌توانند صفت باشند.
- یک جای مناسب دیگر نگاه کردن به سند نیازمندیها است. هر قطعه از اطلاعاتی که باید توسط سیستم جمع‌آوری شود می‌تواند یک صفت از کلاس باشد.
- راه دیگر بررسی ساختار بانک اطلاعاتی است. در صورتیکه بانک اطلاعاتی به طور کامل تعریف شده باشد فیلدهای موجود در جداول می‌توانند ایده خوبی درباره صفات به ما بدهند.(همیشه یک تناظر یک به یک بین جداول بانک اطلاعاتی و کلاسهای Entity وجود دارد)
- در حالت عادی تعداد صفات یک کلاس نباید بیش از ۱۵-۱۰ عدد باشد و در صورت لزوم باید به دو یا چند کلاس شکسته شود. همینطور می‌توان در مورد کلاسهایی که تعداد معدودی صفت دارند قضاوت کرد و در صورت لزوم آنها را با یکدیگر ترکیب کرد.
- مواقعی که بین صفت و کلاس شک می‌کنیم باید ببینیم آیا عنصر ت أثیر و عملکردی در برنامه دارد یا خیر. اگر داشت کلاس و در غیر این صورت صفت خواهد بود.

۶-۲) افزودن صفات

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- سه بخش اصلی از اطلاعات هر صفت عبارتست از: نام صفت، نوع داده‌ای و مقدار اولیه
- برای افزودن صفت ۳ راه وجود دارد:
  - مستقیما درون نمودار روی شکل کلاس تایپ کنیم.
  - در پنجره مرورگر روی نام کلاس کلیک راست کرده و از منوی New آیتم Attribute را انتخاب کرد.
  - از طریق پنجره مشخصات کلاس در برگه Attribute و با از استفاده از منوی میانبر Insert صفت را به کلاس بیافزاییم.
- برای صفت هم میتوان مانند دیگر عناصر مدل مستنداتی ذکر کرد.
- با حذف صفت از کلاس، صفت مذکور از مدل هم حذف می‌شود.

۶-۳) مشخصات صفت

۶-۳-۱) نوع داده‌ای صفت(Type): می‌تواند یکی از انواع داده‌ای استاندارد زبانهای برنامه‌نویسی یا کلاسهای تعریف شده در مدل باشد.

- برای تنظیم نوع داده‌ای باید فیلد Type در برگه General را تغییر داد.
- اگر بخواهیم کلاسهای تعریف شده در مدل هم در لیست انواع داده‌ای بیاید باید چک باکس Show Classes را فعال کرد.

۶-۳-۲) تنظیم مقدار اولیه صفت: این مشخصه اختیاری بوده و جایگاه آن در فیلد Initial Value در برگه General از پنجره مشخصات است.

۶-۳-۳) تنظیم Visibility صفت: محدوده دیده شدن صفت توسط کلاسهای دیگر را مشخص می‌کند.

- Public: نشان دهنده این است که صفت برای تمامی کلاسهای دیگر قابل رؤیت است.
- Private: نشان می‌دهد این صفت فقط برای توابع همین کلاس قابل رؤیت است.
- Protected: حاکی از دیده شدن صفت توسط توابع همین کلاس و نیز کلاسهای ارث برنده از این کلاس است.
- Implementation: نشان می‌دهد صفت فقط توسط کلاسهایی که در بسته جاری قرار دارند می‌تواند دیده شود.

۶-۳-۳-۱) نماد Visibility صفت: دو دسته نماد در این زمینه وجود دارد:

۱- نماد UML: علامت + برای Public، -برای Private و علامت # برای Protected و بدون علامت برای Implementation

۲- نماد Rose که بصورت پیش فرض انتخاب شده است.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- برای تغییر نماد می‌توان در پنجره امکانات، برگه Notation آیتم Visibility As Icon استفاده کرد. تغییر نماد از این به بعد اعمال می‌شود.
- برای ذکر اینکه نمادهای Visibility نمایش داده شوند یا خیر می‌توان روی کلاس در نمودار کلیک راست کرده و از منوی Options، آیتم Show Visibility را انتخاب کرد.

۶-۳-۴) تنظیم محدودیت‌های صفت: نشان می‌دهد که صفت چگونه در کلاس ذخیره شده است. ۳ انتخاب برای این مشخصه وجود دارد که عبارتست از: (برای تنظیم محدودیت صفت باید به برگه Detail از پنجره مشخصات صفت رجوع کنیم)

- **By Value(توسط مقدار):** نشان میدهد که مقدار صفت درون کلاس قرار دارد مثلا اگر صفت از نوع String باشد کل رشته درون کلاس تعریف می‌شود.
- **By Reference(توسط مرجع):** نشان می‌دهد محتوای صفت خارج از کلاس قرار دارد و فقط یک اشاره‌گر(Pointer) به آن در کلاس قرار دارد.
- **Unspecified(نامعین):** هنوز مشخص نشده است. در زمان تولید کد Rose انتخاب By Value را در نظر می‌گیرد.

۶-۳-۵) ایستا نمودن یک صفت(Static): یک صفت ایستا صفتی است که توسط تمام آبجکت‌های کلاس بتواند استفاده شود در حالیکه در مورد صفت معمولی هر آبجکت از کلاس کپی صفت خود را دریافت خواهد کرد.

برای تنظیم ایستا نمودن یک صفت باید به برگه Detail از پنجره مشخصات صفت مراجعه کرد. صفتی که ایستا باشد در شکل نمودار با یک علامت $ قبل از آن ظاهر میشود.

۶-۳-۶) تعریف یک صفت مشتق شده(Derived Attribute): صفتی را مشتق شده گوییم که از یک یا چند صفت دیگر ساخته شده باشد، بطور مثال ممکن است صفتی با نام Area داشته باشیم که از دو صفت دیگر Height , Width مشتق شده باشد.

برای تنظیم مشتق بودن یک کلاس باید به برگه Detail از پنجره مشخصات صفت مراجعه کرد. بعد از ست کردن صفت به عنوان مشتق شده نام آن با یک علامت / قبل از آن در شکل نمودار ظاهر میشود.

۶-۴) کار با عملیات(Operations): عملیات یا توابع، رفتار و عملکرد مربوط به یک کلاس می‌باشد. یک عملیات ۳ قسمت دارد: نام عملیات،

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

پارامترهای ورودی عملیات و نوع برگشتی عملیات. در UML عملیات با استفاده از فرمت زیر نشان داده می‌شود:

Operation Name(argument1 : argument1 data type, ...) : return type

هنگام اضافه کردن عملیات جدید می‌توان با تبعیت از فرمت فوق الذکر و نوشتن آن روی جایگاه عملیات در شکل کلاس، عملیات جدیدی به کلاس اضافه کرد. همچنین باید نکات زیرا را به خاطر سپرد:

- به هر کلاسی که فقط یک یا دو عملیات داشته باشد باید به دیده تردید نگریست، ممکن است هیچ اشکالی در تعریف کلاس نباشد و یا نیاز به ترکیب دو یا چند کلاس باشد.
- کلاسی که هیچ عملیاتی نداشته باشد بهتر است به صفات تشکیل دهنده آن شکسته شود.
- با کلاسهایی که تعداد زیادی عملیات دارند باید با احتیاط رفتار کرد. هر کلاس باید مجموعه قابل مدیریتی از مسئولیتها را داشته باشد. ممکن است تقسیم کلاس کار را ساده‌تر کند.

۶-۵) یافتن عملیات: زمانیکه نمودارهای توالی و همکاری را می‌سازیم بیشترین عملیات مربوط به یافتن عملیات را انجام داده‌ایم. چهار نوع متفاوت از عملیات داریم که عبارتند از:

۶-۵-۱) عملیاتهای مجری(Implementor): عملیاتهای مجری برخی وظایف تجاری را انجام می‌دهند و نوعا با تست نمودارهای تعامل بدست می‌آیند. هر پیغامی روی نمودار تعامل معمولا به یک عملیات مجری نگاشت می‌شود. این نوع از عملیات باید دارای قابلیت تبدیل به یک نیاز را داشته باشد و معمولا از نوع عمومی(Pubic) هستند.

۶-۵-۲) عملیاتهای مدیریتی(Manager): ساخته شدن و خراب شدن آبجکتها را کنترل می‌کنند. توابع سازنده(Constructor)و مخرب(Destructor) در این دسته می‌گنجند.

۶-۵-۳) عملیاتهای دسترسی(Access): با توجه به اینکه صفات معمولا اختصاصی(Private) یا محافظت شده هستند برای دسترسی به مقدار صفات یا تغییر آنها از این نوع از عملیات استفاده می‌شود. نامگذاری این توابع بدین صورت است که کلمات Get و Set را به ابتدای نام صفت مربوطه اضافه می‌کنیم.

۶-۵-۴) عملیاتهای امدادرسانی(Helper): عملیاتی هستند که کاربرد داخلی در کلاس دارند و لازم به دانستن آنها توسط کلاسهای بیرونی نمی‌باشد. عملیاتهای امدادرسانی معمولا به صورت پیغامهای

Info@IRANMEET.COM
www.pupuol.com

بازگشتی بر روی نمودارهای تعامل ظاهر میشوند. Visibility این نوع از عملیات معمولا اختصاصی یا محافظت شده است.

۶-۶) افزودن عملیات: عملیات را نیز مانند صفات می‌توان از طریق پنجره مرورگر و یا از طریق برگه Operations در پنجره مشخصات کلاس به کلاس اضافه کرد. مستنداتی که به عملیات اضافه می‌شود در کد نهایی تولید شده نیز ظاهر میشوند.

۶-۷) مشخصات عملیات

۶-۷-۱) تنظیم کلاس برگشتی عملیات(Return Class): نوع داده‌ای بازگشتی عملیات را معرفی می‌کند که می‌تواند یکی از داده‌های استاندارد زبان‌های برنامه‌نویسی یا یکی از کلاس‌های تعریف شده در مدل باشد.

- جایگاه این مشخصه در برگه General و با عنوان Return Class می‌باشد.
- جهت نشان دادن کلاسهای تعریف شده در مدل در لیست انواع داده‌ای باید چک باکس Show Classes را در برگه General فعال کرد.

۶-۷-۲) تنظیم Visibility عملیات: این گزینه نیز دقیقا شبیه گزینه مشابه در صفات می‌باشد. برای توضیح بیشتر به قسمت تنظیم Visibility صفت مراجعه شود.

۶-۷-۳) افزودن آرگومان‌ها به یک عملیات: هر آرگومان یا پارامتر ورودی دو قطعه اطلاعاتی را داراست اول نام آرگومان و دوم نوع آن.

- برای اضافه کردن یک آرگومان جدید باید به برگه Detail از پنجره مشخصات عملیات مراجعه کرده و با استفاده از کلید میانبر Insert در قسمت Arguments آرگومان جدید را اضافه کرد.
- برای حذف آرگومان می‌توان از کلید میانبر Delete در قسمت فوق الذکر استفاده کرد.

۶-۷-۴) تعریف پروتکل(Protocol) عملیات: معین می‌کند در ارتباط با اجرای این تابع(عملیات) چه اتفاق دیگری باید افتاده باشد، مثلا اگر تا زمانیکه عملیات B اجرا نشده نباید عملیات A انجام شود، باید این موضوع را در فیلد پروتکل عملیات A یادداشت کرد. توضیح وارد شده در کد نهایی ظاهر میشود. برای تنظیم گزینه می‌توان به برگه Detail از پنجره مشخصات و به فیلد Protocol مراجعه کرد.

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

۶-۷-۵) تعریف Qualifications(شرط) عملیات: این فیلد به ما اجازه تعریف شروط مختص به زبان برنامه‌نویسی را می‌دهد. توضیح وارد شده در کد نهایی ظاهر می‌شود. جایگاه این تنظیم فیلد Qualification از برگه Detail در پنجره مشخصات عملیات می‌باشد.

۶-۷-۶) تعریف Exceptions(استثنائات) عملیات: مکانی است که می‌توان موارد استثنایی که ممکن است عملیات با آن روبرو شود را ذکر کنیم. توضیح وارد شده در کد نهایی ظاهر می‌شود. جایگاه این تنظیم فیلد Exceptions از برگه Detail در پنجره مشخصات عملیات می‌باشد.

۶-۷-۷) تعریف اندازه(Size) عملیات: معین می‌کند عملیات در زمان اجرا به چه مقدار حافظه نیاز دارد. این فیلد هم در کد نهایی بصورت یک توضیح ظاهر می‌شود. جایگاه تنظیم، فیلد size از برگه Detail در پنجره مشخصات عملیات می‌باشد.

۶-۷-۸) تعریف زمان(time) عملیات: مقدار زمان تخمینی است که عملیات برای اجرا نیاز دارد. این فیلد هم در کد نهایی بصورت یک توضیح ظاهر می‌شود. جایگاه تنظیم، فیلد time از برگه Detail در پنجره مشخصات عملیات می‌باشد.

۶-۷-۹) تعریف همروندی(Concurrency) عملیات: مشابه مبحث همروندی در کلاس می‌باشد با این تفاوت که گزینه همروندی در این جا به Sequential، Synchronous و Guarded محدود می‌شود. همروندی عملیات بصورت یک توضیح در کد نهایی ظاهر میشود. جایگاه این تنظیم، فیلد Concurrency در برگه Detail از پنجره مشخصات عملیات می‌باشد.

۶-۷-۱۰) تعریف پیش شرایط(Preconditions) عملیات: شرایطی هستند که قبل از اینکه عملیات بتواند اجرا شود باید true باشد.

- جایگاه تعریف پیش شرایط برگه Preconditions از پنجره مشخصات عملیات می‌باشد.
- اگر یک نمودار تعامل داشته باشیم که پیش شرایط عملیات را نشان دهد می‌توانیم آنرا در برگه Preconditions از پنجره مشخصات عملیات و در فیلد Interaction Diagram ذکر کنیم. در این جایگاه می‌توان یکی از نمودارهای موجود را انتخاب یا نمودار جدیدی ایجاد کرد.

۶-۷-۱۱) تعریف شرایط پسین(Postconditions) عملیات: شرایطی هستند که همیشه باید بعد از خاتمه اجرای عملیات true باشند.

- جایگاه تعریف شرایط پسین در برگه Postconditions از پنجره مشخصات عملیات می‌باشد.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- اگر یک نمودار تعامل داشته باشیم که شرایط پسین عملیات را نشان دهد می‌توانیم آنرا در برگه Postconnditions از پنجره مشخصات عملیات و در فیلد Interaction Diagram ذکر کنیم. در این جایگاه می‌توان یکی از نمودارهای موجود را انتخاب یا نمودار جدیدی ایجاد کرد.

۶-۷-۱۲) تعریف Semantic(معنای) عملیات: جایگاهی است که می‌توان منطق کاری عملیات را توضیح داد. در این جا می‌توان از شبه کد(Pseudo-Code) یا فقط تعاریف استفاده کرد. هر آنچه در این جایگاه وارد شود نهایتا به صورت یک توضیح در کد تولید خواهد شد.

- جایگاه تعریف معنای عملیات در برگه Semantics از پنجره مشخصات عملیات می‌باشد.
- اگر یک نمودار تعامل داشته باشیم که معنای عملیات را نشان دهد می‌توانیم آنرا در برگه Semantics از پنجره مشخصات عملیات و در فیلد Interaction Diagram ذکر کنیم. در این جایگاه می‌توان یکی از نمودارهای موجود را انتخاب یا نمودار جدیدی ایجاد کرد.

۶-۸) نمایش صفات و عملیات بر روی نمودارهای کلاس: برنامه Rational بسیار انعطاف پذیر است و اجازه می‌دهد فقط آن جزئیاتی که می‌خواهیم را نمایش دهد. می‌توان نمودار کلاس را به گونه‌ای تغییر داد که:

- تمام صفات و عملیات را نشان دهد.
- صفات را پنهان سازد.
- عملیات را پنهان سازد.
- صفات و عملیات انتخاب شده را نمایش دهد.
- نشان‌های عملیات و یا فقط نام‌های عملیات را نشان دهد.
- Visiblityهای صفات و عملیات را پنهان یا آشکار سازد.
- Stereotypeهای صفات و عملیات را پنهان یا آشکار سازد.

۶-۸-۱) نمایش صفات

۶-۸-۱-۱) نمایش یا پنهان کردن همه صفات یک کلاس خاص:
برای این کار میتوان بعد از انتخاب(فعال کردن) کلاس:

- از منوی میانبر راست کلیک روی کلاس، منوی Options، گزینه Show All Attributes را انتخاب کرد.
- از منوی Format، گزینه Show All Attributes را انتخاب کرد.

۶-۸-۱-۲) نمایش صفات منتخب یک کلاس خاص: برای این کار بعد از فعال کردن آن می‌توان:

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- از منوی میانبر کلیک راست، منوی Options، گزینه Select Compartment Item را انتخاب کرد. در دیالوگ باز شده می‌توان صفات مورد نظر را انتخاب کرده و به لیست اضافه کرد.
- از منوی Edit گزینه Compartment را انتخاب کرد. در دیالوگ باز شده می‌توان صفات مورد نظر را انتخاب کرده و به لیست اضافه کرد.

۶-۸-۱-۳) لغو صفات(عدم نمایش خود صفت و نیز کادر شامل آن) یک کلاس خاص: برای این کار بعد از فعال کردن آن می‌توان:

- از منوی میانبر کلیک راست، منوی Options، گزینه Suppress Attributes را انتخاب کرد.
- ۲ز منوی Format گزینه Suppress Attributes را انتخاب کرد.

۶-۸-۱-۴) تغییر گزینه پیش فرض برای تمامی کلاس‌هایی که از این به بعد ایجاد می‌شوند: برای این کار باید از منوی در پنجره امکانات، برگه Diagram، گزینه‌های Show All Attributes و Suppress Attributes را انتخاب کرد.

۶-۸-۲) نمایش عملیات: کلیه کارهایی که با نمایش صفات امکان پذیر بود اینجا برای عملیات هم امکان پذیر است. نحوه عمل به همان شکل است با این تفاوت که بجای بعضی گزینه‌ها، باید گزینه‌های دیگری را تصور کرد.

- Show All Operations بجای Show All Attributes
- Suppress Operations بجای Suppress Attributes

۶-۸-۳) نمایش Visibility

۶-۸-۳-۱) نمایش یا عدم نمایش Visibility برای یک کلاس خاص: برای این کار بعد از فعال کردن آن می‌توان:

- از منوی میانبر کلیک راست، منوی Options، گزینه Show Visibility را انتخاب کرد.
- از منوی Format گزینه Show Visibility را انتخاب کرد.

۶-۸-۳-۲) نمایش یا عدم نمایش Visibility کلیه کلاس‌هایی که از این به بعد ایجاد می‌شوند: برای این کار می‌توان از پنجره امکانات، برگه Diagram، گزینه Show Visibility را انتخاب کرد.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

رابطه‌ها (Relationships)

رابطه یک ارتباط معنایی بین کلاسهاست که به یک کلاس اجازه میدهد درباره صفتها، عملیات و ارتباطات کلاس دیگر چیزهایی بداند. برای اینکه یک آبجکت از طریق نمودار توالی بتواند پیغامی به آبجکت دیگر بفرستد باید یک رابطه بین کلاسهای آنها وجود داشته باشد.

۷-۱) پیدا کردن رابطه‌ها: چند راه برای یافتن رابطه بین کلاسها وجود دارد که عبارتند از:

- نمودارهای توالی و همکاری را بررسی کنیم. هر پیغامی بین دو آبجکت به معنای ارتباط بین کلاسهای آنهاست.
- کلاسهای یافت شده را بررسی کنیم و بدنبال رابطه‌های کل به جزء(Whole-Part) بگردیم.
- کلاسهای یافت شده را بررسی کنیم و سعی کنیم کلاسهایی را پیدا کنیم که ممکن است انواع مختلف داشته باشد. هم چنین سعی کنیم کلاسهایی را پیدا کنیم که مشترکات زیادی با هم دارند.

۷-۲) Associationها: یک رابطه معنایی بین کلاسهاست که به یک کلاس امکان می‌دهد درباره صفتها و عملیات عمومی کلاس دیگر بداند. وجود پیغام بین دو آبجکت در نمودار توالی به معنای وجود رابطه Association بین کلاس-های آنهاست. این رابطه می‌تواند یکطرفه یا دو طرفه باشد که در تولید کد به معنای تعریف یک آبجکت از یک کلاس در زمره متغیرهای کلاس دیگر می‌باشد.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- رابطه‌های Association دو طرفه پیشنهاد نمی‌شود با توجه به اینکه امکان استفاده مجدد(Reuse) را کاهش میدهد.
- کلاس‌هایی که در انتهای فلش قرار می‌گیرد به راحتی می‌توانند استفاده مجدد شوند.
- رابطه Association می‌تواند بازتابی باشد که حاکی از این است که یک نمونه از یک کلاس به سایر نمونه‌های همان کلاس وابسته است.
- برای ایجاد رابطه Association یکطرفه باید آیکون Unidirectional Association را از نوار ابزار انتخاب کرده و بین دو کلاس کشید.
- برای ایجاد رابطه Association دو طرفه باید آیکون Association را از نوار ابزار انتخاب کرده و بین دو کلاس کشید.
- برای اینکه فلش فقط در انتهای یک طرف رابطه یا در انتهای دو طرف رابطه باشد باید در انتهای طرف مربوطه کلیک راست کرده و گزینه Navigable را از منوی میانبر انتخاب کرد.
- برای ایجاد یک Association بازتابی باید آیکون Association را انتخاب کرده و روی کلاس مزبور کلیک کرد، سپس آنرا به فضایی خارج از کلاس کشید و دوباره روی آن رها کرد.
- Associationهای اضافه شده به نمودار در مدل(پنجره مرورگر) و ذیل Associations بصورت یک عنصر مدل اضافه می‌شوند. حذف این رابطه از نمودار باعث حذف آن از مدل نمی‌شود.

۳-۷) Dependencyها: این نوع از رابطه نیز دو کلاس را به هم وصل می‌کند اما با یک تفاوت کوچک نسبت به Associationها و اینکه Dependencyها همیشه یکطرفه هستند و نشان می‌دهد که یک کلاس به تعاریف کلاس دیگر بستگی دارد.

- هنگام تولید کد هیچ صفتی از یک کلاس در کلاس دیگر تعریف نمی‌شود و فقط کد لازم(مثلا در C++ دستور Include) جهت شناسایی آورده می‌شود.
- در کد نهایی تولید شده اگر یک کلاس بخواهد به متغیری از کلاس دیگر(که به آن وابسته است) دسترسی داشته باشد سه راه وجود خواهد داشت:
  - تعریف یک متغیر Global از آن کلاس
  - ایجاد متغیر محلی در داخل تابع از آن کلاس
  - پاس کردن متغیری از آن کلاس به درون تابع

Info@IRANMEET.COM
www.pupuol.com

- برای ایجاد این رابطه باید آیکون Instantiated or Dependency را انتخاب کرده و بین دو کلاس کشید.
- رابطه Dependency را میتوان بین دو بسته نیز کشید. معنی Dependency بین دو بسته این است که بعضی از کلاسهای یک بسته با بعضی از کلاسهای بسته دیگر یک رابطه یکطرفه دارد. استفاده مجدد از بسته‌ای که در انتهای فلش قرار گرفته آسان است ولی بسته دیگر حتما باید همراه به بسته وابسته شده به آن بیاید.
- از رابطه Dependency حلقه‌ای(دو طرفه) بین دو بسته باید پرهیز کرد و در صورت لزوم با تجزیه کردن یکی از آنها(یا هر دو آنها) به بسته‌های دیگر صرفا رابطه یکطرفه بین بسته‌ها ایجاد کرد.

۷-۴) Aggregationها: این رابطه یک فرم قویتر Association است و رابطه بین یک واحد کل(Whole) و اجزاء(Part) آن است. مثلا کلاس Car یک رابطه از این نوع با Tire و Engine دارد.

- رابطه Aggregation هم می‌تواند بازتابی باشد و بدین معنی است که یک نمونه از یک کلاس می‌تواند شامل چند نمونه از همان کلاس باشد.
- هنگام تولید کد صفتهایی از کلاس‌های Part در کلاس Whole ایجاد می‌شود.
- برای ایجاد این رابطه باید از آیکون Aggregation استفاده کرد و از کلاس Whole به کلاس Part کشید. در شکل رابطه از این نوع در لوزی کوچک در طرف کلاس Whole قرار می‌گیرد.
- برای تغییر دادن نوع این رابطه به نوعی دیگر میتوان از منوی Edit، گزینه Change Into استفاده کرد.
- برای ایجاد Aggregation بازتابی باید بعد از انتخاب آیکون مربوطه روی کلاس کلیک کرده و به مکانی خارج از کلاس کشید و دوباره روی کلاس آنرا رها کرد.
- رابطه Aggregate ایجاد شده در مدل(پنجره مرورگر) ذیل Associations قرار می‌گیرد.

۷-۵) Generalitationها: یک رابطه وراثتی بین دو کلاس است که به یک کلاس اجازه میدهد تا صفتها و عملیات عمومی و محافظت شده کلاس دیگر را به ارث برد.

- هر زیر کلاس(SubClass) می‌تواند صفات و عملیات منحصر به خود را داشته باشد.

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- هنگام تعریف این نوع از رابطه می‌توان ساختار وراثت از بالا به پایین(کلاسهایی که ممکن است انواع مختلفی داشته باشند) یا پایین به بالا(کلاسهایی که وجوه مشترک دارند) را ساخت.
- حفظ و نگهداری و تغییر در رابطه‌های وراثتی که تعداد لایه‌های زیادی دارند مشکل است.
- برای ایجاد این رابطه باید از آیکون Generalitation استفاده کرد و از کلاس فرزند به کلاس پدر کشید.

۷-۶) مشخصات رابطه‌ها

۷-۶-۱) تنظیم Multiplicity: نشان می‌دهد که چه تعداد نمونه از یک کلاس به یک تک نمونه یا کلاس دیگر مرتبط است. نشانگر Multiplicity در انتهای رابطه قرار می‌گیرد و می‌توانند در دو طرف رابطه وجود داشته باشد.

- انواع این مشخصه مشابه کلاس می‌باشد.
- مقدار این مشخصه بین فرم‌ها، صفحه‌ها یا پنجره‌ها در هر طرف رابطه معمولا 0..1 است که نشان می‌دهد هر فرم می‌تواند مستقل از فرم‌های دیگر وجود داشته باشد.
- برای تنظیم این مشخصه می‌توان در هر طرف رابطه کلیک راست کرده و از منوی Multiplicity مقدار لازم را انتخاب کرد. هم چنین می‌توان با استفاده از پنجره مشخصات رابطه، برگه Role Detail (Role Aمربوط به رابطه منتهی به یکی از کلاس‌ها و Role B مربوط به رابطه منتهی به کلاس دیگر می‌باشد) تعیین کرد.

۷-۶-۲) نام‌های رابطه: خود رابطه یک نام و هر یک از Roleهای آن(رابطه منتهی به هر یک از کلاس‌ها) می‌توانند نام مربوط به خود را داشته باشند. برای تنظیم نام Role می‌توان از پنجره مشخصات رابطه برگه Role A Detail یا برگه Role B Detail استفاده کرد.

۷-۶-۳) تنظیم Export Control: معین می‌کند هنگام تولید صفتی که در یک کلاس تولید می‌شود دارای چه Visibility باشد.

- انواع Visibility عبارتند از: Private، Public، Protected و Implementation.. در صورتیکه رابطه دو طرفه باشد باید برای هر طرف این مشخصه را جداگانه تعریف کرد.
- برای تنظیم این مشخصه باید به قسمت Export Control از برگه Role General مراجعه کرد.

۷-۶-۴) رابطه‌های ایستا: تعیین می‌کند آیا صفتهایی که در اثر وجود رابطه در کد نهایی تولید می‌شوند باید استاتیک باشد یا خیر.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- برای تنظیم این مشخصه باید چک‌باکس Static را از برگه Role Detail انتخاب کرد.
- رابطه‌ای که از نوع استاتیک باشد کنار نام Role مربوطه علامت $ ظاهر می‌شود.

۷-۶-۵) رابطه‌های Friend: تعیین می‌کند آیا کلاس مشتری(Client) به صفتهای غیر عمومی کلاس Supplier(کلاسی که در انتهای فلش قرار می‌گیرد) دسترسی داشته باشد یا خیر. این عمل در کد نهایی با استفاده از کلمه کلیدی Friend میسر می‌شود.

۷-۶-۶) تنظیم Containment: تعیین می‌کند که آیا صفتهای تولید شده در کد نهایی از نوع By Value هستند یا By Reference.

- اعمال این گزینه از طریق گزینه Containment of... در برگه Role Detail یا از طریق کلیک راست در یکی از دو انتهای رابطه، منوی، Containment of...، انتخاب گزینه مناسب می‌باشد.
- در رابطه Aggregation در صورتیکه رابطه از نوع By Value باشد لوزی به صورت توپر نمایش داده می‌شود.

۷-۶-۷) Qualifierها: پارامتری است که حوزه(Scope) یک رابطه را کاهش می‌دهد مثلا اگر رابطه‌ای بین شخص و شرکت داشته باشیم می‌توان یک Qualifier به نام Person ID داشت که به ازای یک مقدار مفروض از آن فقط دو شرکت داشته باشیم.

- برای اضافه کردن Qualifier باید از قسمت Keys/Qualifiers در برگه Role Detail استفاده کرد و با کلیک راست و منوی Insert پارامتر لازم را اضافه کرد.
- راه دیگر برای اضافه کردن Qualifier انتخاب منوی میانبر(کلیک راست) و انتخاب گزینه New Key/Qualifier می‌باشد. در پی این انتخاب پنجره‌ای ظاهر میشود که میتوان Qualifier مربوطه را اضافه کرد.
- کنار کلاسی که دارای Qualifier باشد کادری بصورت مستطیل ظاهر می‌شود که نام Qualifier در آن قرار دارد.

۷-۶-۸) عناصر پیوند(Link Elements): یک عنصر پیوند که هم چنین با نام یک کلاس Association نیز شناخته می‌شود مکانی است که در آن صفتهای مربوط به یک رابطه ذخیره می‌شود. مثلا اگر دو کلاس دانشجو و درس داشته باشیم صفت نمره را در هیچ کدام نمیتوان قرار داد. برای حل این مشکل نمره را در کلاس دیگری قرار می‌دهیم که با نام Association Class شناخته می‌شود.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- برای ایجاد یک عنصر پیوند می‌توان از آیکون Association calss استفاده کرده و از کلاس سوم(مثلا کلاس شامل نمره) به ارتباط بین دو کلاس کشید.
- یک راه دیگر انتخاب گزینه Link Element در برگه Detail از پنجره مشخصات رابطه است.

۷-۶-۹) محدودیتها(Constraints): عبارت از شرطهایی هستند که باید صحیح باشند و می‌توانند به رابطه یا به هر یک از رل‌های آن اعمال شوند. توضیح مشخص شده در این قسمت در کد تولید شده ظاهر می‌شود.

برای تنظیم این قسمت می‌توان قسمت Constraints را از برگه Detail یا برگه Role Detail انتخاب کرد و توضیح لازمه را در آن ثبت نمود.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

رفتار آبجکت(Object Behavior)

در این فصل به جزئیات رفتار یک آبجکت یا کلاس می‌پردازیم. یک آبجکت می‌تواند در یک یا چند حالت مختلف وجود داشته باشد. مثلا یک کارمند می‌تواند استخدام شده، اخراج شده یا در طی دوره آزمایشی باشد. مبحث این فصل درباره نمودارهای حالت، چگونگی تغییر یک آبجکت از حالتی به حالت دیگر و اینکه آبجکت چگونه در هر کدام از حالات رفتارها و عملکردهای متفاوت دارد، می‌باشد.

۸-۱) نمودارهای تغییر حالت(State Transition Diagram): یک نمودار حالت چرخه زندگی یک آبجکت منفرد را از زمانی که ایجاد می‌شود تا زمانی که از بین می‌رود، نشان می‌دهد. این نمودار روش خوبی برای مدل کردن رفتار دینامیکی یک کلاس است.

- در یک پروژه معمولی لازم نیست برای هر کلاس نمودار حالت ایجاد شود و فقط برای کلاس‌های پیچیده این کار صورت می‌گیرد.
- معمولا وجود یک صفت در کلاس که وضعیت(State) را نشان می‌دهد مبین این نکته است که کلاس نیاز به نمودار حالت دارد.
- هم چنین اگر رابطه‌های یک کلاس را بررسی کنیم رابطه‌هایی که دارای Multiplicity برابر با صفر یا بیشتر باشند نشانگر این هستند که رابطه انتخابی است و در نتیجه کلاس می‌تواند در حالات مختلفی قرار داشته باشد.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- هیچ کدی از نمودار حالت تولید نمی‌شود و فقط جنبه مستندسازی و کمک به درک بهتر رفتار دینامیکی کلاس دارد. مورد استفاده این نمودار برای تحلیل گران و برنامه نویسان می‌باشد.

۸-۲) اضافه کردن حالت: یک حالت یک وضعیت ممکن است که امکان دارد یک آبجکت در آن قرار گرفته باشد.

- اضافه کردن جزئیات به حالت: وقتی یک آبجکت در یک حالت خاص است ممکن است فعالیت‌هایی داشته باشد که آنها را اجرا می‌کند. پنج نوع اطلاعات وجود دارد که می‌تواند در یک حالت قرار داده شود.

۸-۲-۱) فعالیت(Activity): رفتار یا عملکردهایی است که یک آبجکت تا وقتی که در آن حالت است اجرا می‌کند. فعالیت یک عملکرد وقفه‌ای است. فعالیت درون شکل حالت ظاهر می‌شود و حالت بصورت State/do : activity در می‌آید.

۸-۲-۲) action ورودی(Entry action): یک رفتار و عملکرد است که وقتی آبجکت در حال انتقال به حالت است اتفاق می‌افتد. این عمل یک عملکرد غیر وقفه‌ای است. Action ورودی درون شکل حالت ظاهر میشود و حالت بصورت State/entry : action در می‌آید

۸-۲-۳) action خروج(Exit action): شبیه action ورودی است بجز اینکه هنگام خروج از حالت به عنوان بخشی از انتقال اتفاق می‌افتد. این عمل نیز غیر وقفه‌ای است. action خروج درون شکل حالت و بصورت State/exit : action ظاهر می‌شود.

- رفتار و عملکرد هر یک از ۳ گونه فوق می‌تواند شامل فرستادن یک رخداد به آبجکت‌های دیگر باشد که در این وضعیت به صورت:(Do:^Target.Event(Argument ظاهر می‌شود.
- یک فعالیت ممکن است به عنوان نتیجه بعضی رخدادهای دریافت شده اتفاق بیافتد.

۸-۲-۴) پنجره مشخصات حالت

۸-۲-۴-۱) اضافه کردن یک فعالیت(Activity): برای این کار برگه Actions را باز کرده و با استفاده از منوی میانبر گزینه Insert را انتخاب می‌کنیم. بر روی action جدید دوباره کلیک می‌کنیم تا پنجره مشخصات آن ظاهر شود. در کادرWhen گزینه Do را انتخاب می‌کنیم.

۸-۲-۴-۲) اضافه کردن یک Action ورودی: برای این کار سیر قبلی را تکرار کرده و در کادر When گزینه On Entry را انتخاب می‌کنیم.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

۸-۲-۴-۳) اضافه کردن یک Action خروجی: برای این کار مسیر قبلی را تکرار کرده و در کادر When گزینه On Exit را انتخاب می‌کنیم.

۸-۲-۴-۴) ارسال رخداد: برای این کار مسیر قبلی را تکرار کرده و در کادر When گزینه On Event را انتخاب می‌کنیم.

در هر یک از حالات فوق اگر بخواهیم رخدادی به یک آبجکت دیگر ارسال شود کافی است در پنجره مشخصات action در کادر Type گزینه Send Event را انتخاب کرده و فیلدهای Send arguments(پارامترهای ارسالی توسط رخداد) و Send Target(آبجکت مقصد) را پر کنیم.

۸-۳) اضافه کردن گذرها(Transitions): یک گذر، انتقال از یک حالت به حالت دیگر است. در نمودار هر انتقال به صورت یک پیکان از حالت مبدأ به حالت مقصد کشیده می‌شود. گذرها می‌توانند بازتابی نیز باشند بدین معنی که ممکن است چیزی اتفاق بیافتد که باعث شود یک آبجکت به حالتی که در حال حاضر در آن است دوباره بازگردد.

- برای اضافه کردن یک گذر باید آیکون State Transition را انتخاب کرد و بین دو حالت کشید.
- برای اضافه کردن یک گذر بازتابی باید آیکون Transition to Self را از نوار ابزار نمودار حالت انتخاب کرد و روی حالت مورد نظر کلیک کرد.
- ویژگیهای متنوعی وجود دارد که می‌توان آنها را به گذر اضافه کرد که عبارتند از:

۸-۳-۱) رخداد(Event): رخداد چیزی است که اتفاق می‌افتد و باعث گذر از یک حالت به حالت دیگر می‌شود. رخداد در نمودار در طول فلش انتقال نشان داده می‌شود.

- نام رخداد می‌تواند بصورت نام یک عملکرد بوده و دارای آرگومان‌هایی باشد.
- میتوان یک انتقال خودکار(بدون رخداد) داشت اگر چه اکثر انتقالها دارای رخداد هستند. با یک انتقال خودکار به محض اینکه تمام actionهای ورودی، خروجی و فعالیتها اتفاق افتادند انتقال از یک حالت به حالت دیگر اتفاق می‌افتد.

۸-۳-۲) حالت شرط(Guard Condition): یک حالت شرط کنترل می‌کند که چه زمانی یک انتقال می‌تواند اتفاق بیافتد. یک حالت شرط در طول خط انتقال بعد از نام رخداد و درون کروشه ظاهر می‌شود. حالات شرطی انتخابی هستند.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

۸-۳-۳) Action: یک عملکرد غیر وقفه‌ای است که به عنوان بخشی از یک انتقال رخ میدهد.actionهای ورود و خروج در درون حالت نشان داده میشود در حالیکه این نوع در طول خط انتقال نمایش داده می‌شود. یک action در طول خط انتقال بعد از نام رخداد نشان داده می‌شود و قبل از آن نیز یک کاراکتر / قرار می‌گیرد. یک action ممکن است خود یک رفتار باشد یا فرستادن رخداد به آبجکتی دیگر باشد.

۸-۳-۴) تنظیم مشخصات گذر

۸-۳-۴-۱) اضافه کردن یک رخداد: برای این کار باید در کادر Event نام رخداد را وارد کرد.

۸-۳-۴-۲) اضافه کردن آرگومان به رخداد: برای این کار باید در کادر Arguments باید آرگومانهای رخداد را وارد کرد.

۸-۳-۴-۳) اضافه کردن یک حالت شرطی: برای این کار در برگه Detail و در کادر Guard condition حالت شرطی را وارد می‌کنیم.

۸-۳-۴-۴) اضافه کردن یک action: برای این کار در برگه Detail و در کادر Action نام action را وارد می‌کنیم.

۸-۳-۴-۵) ارسال رخداد: برای این کار در برگه Detail از کادرهای Send Event, Send Arguments, Send Target استفاده می‌کنیم.

۸-۴) اضافه کردن حالات خاص

۸-۴-۱) حالت آغازین(Start State): حالتی است که وقتی آبجکت ایجاد می‌شود در آن قرار دارد و به صورت یک دایره توپر نشان داده می‌شود.

- حالت آغازین اجباری است و فقط یک عدد از آن می‌تواند وجود داشته باشد.
- برای اضافه کردن آن از آیکون Start State استفاده می‌کنیم.

۸-۴-۲) حالت پایانی(Stop State): حالتی است که وقتی آبجکت خراب می‌شود در آن قرار می‌گیرد. این حالت بوسیله یک دایره توپر با خطی دور آن نشان داده می‌شود.

- حالت پایانی انتخابی است و به تعداد نامحدود می‌توان از آن در نمودار ایجاد کرد.
- برای اضافه کردن آن به نمودار از آیکون Stop State استفاده می‌کنیم.

۸-۵) استفاده از حالات تو در تو(Nested States): برای جلوگیری از بی‌نظمی در نمودار میتوان یک یا چند حالت را درون یک حالت دیگر قرار

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

دارد. حالات درونی به عنوان زیر حالت(SubState) و به حالت بزرگتر حالت ما فوق(SuperState) گفته می‌شود.

- اگر دو یا چند حالت یک گذر یکسان داشته باشند می‌توانند با هم یک گروه را تشکیل دهند و یک حالت مافوق شوند.
- اگر بخواهیم تاریخچه را در یک حالت مافوق حفظ کنیم(بدین معنی که بدانیم دقیقا چه زیر حالتی باعث گذر شده است) تا بعدا بتوانیم از همان مسیر گذر برگردیم بایستی از تاریخچه حالت(State History) که بصورت یک حرف H داخل یک دایره کنار حالت مافوق ظاهر میشود استفاده کنیم.
- برای تو در تو کردن یک حالت آیکون حالت را از نوار ابزار انتخاب کرده و بر روی حالتی که می‌خواهد حالت مافوق باشد کلیک می‌کنیم.
- برای استفاده از تاریخچه حالت کافی است چک باکس State/activity history را در برگه General از پنجره مشخصات حالت مافوق فعال کنیم.
- برای اینکه تاریخچه حالت به حالت‌های داخلی‌تر نیز تعمیم پیدا کند کافی است چک‌باکس Substate/activity history را در برگه General از پنجره مشخصات حالت مافوق فعال کنیم.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

نمای Component

در نمای کامپوننت بر روی سازماندهی فیزیکی سیستم تمرکز می‌کنیم. ابتدا تصمیم می‌گیریم که چگونه کلاسها در کتابخانه‌های کد سازماندهی شوند سپس به فایل‌های اجرایی مختلف، فایل‌های DLL و سایر فایل‌های زمان اجرا در سیستم نگاهی می‌اندازیم.

- یک کامپوننت یک ماژول فیزیکی کد است. کامپوننت می‌تواند هم حاوی کتابخانه کد منبع و هم فایل‌های زمان اجرا باشد. بطور مثال در C++ فایل‌های H, CPP هر کدام یک کامپوننت می‌باشند.
- قبل از تولید کد باید کلاس‌هایمان را به کامپوننت(های) مناسب نسبت دهیم.

۹-۱) انواع کامپوننت‌ها: دو گروه اصلی از انواع کامپوننت وجود دارد که عبارتست از کامپوننت‌های کتابخانه کد منبع و کامپوننت‌های زمان اجرا. هر کدام از این دو گروه انواع مختلفی دارند که ذیلا به آنها می‌پردازیم.

۹-۱-۱) Subprogram Specification & Body: نشان دهنده یک زیر برنامه که معمولا بعنوان مجموعه‌ای از زیر روال‌ها(Subroutine) است بکار میرود.

۹-۱-۲) Main Program: فایلی است که حاوی ریشه برنامه می‌باشد.(مثلا فایل حاوی کلاس Application)

۹-۱-۳) Package Specification & Body: یک بسته پیاده‌سازی یک کلاس است.Package Specification اشاره به فایل سربار(H.) و Packag Body اشاره به فایل بدنه(Cpp.) می‌کند.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

۹-۱-۴) **Dll File:** اشاره به فایل‌های Dll دارد.
۹-۱-۵) **Task Specification & Body:** نوعا برای فایل‌های اجرایی(Exe.) بکار میرود.

۹-۲) نمودارهای کامپوننت(Component Diagram): یکی از انواع نمودارهای UML است که کامپوننت‌های موجود در سیستم و رابطه‌های Dependency بین آنها را نمایش می‌دهد.

- معمولا متناظر با هر بسته(Package) درون نمای منطقی یک بسته همنام در نمای کامپوننت ساخته می‌شوند.
- بهتر است کلاس‌های متناظر در نمای منطقی با کامپوننت‌ها در نمای کامپوننت همنام باشند.

۹-۲-۱) اضافه کردن یک کامپوننت ساده به نمودار: برای این کار آیکون Component را از نوار ابزار انتخاب کرده و روی نمودار کلیک می‌کنیم.

۹-۲-۲) اضافه کردن انواع دیگر کامپوننت به نمودار: برای این کار می‌توان آیکون‌های مربوط را از نوار ابزار انتخاب کرد یا یک کامپوننت معمولی را اضافه کرد و سپس نوع آنرا از طریق مشخصهStereotype در برگه General از پنجره مشخصات کامپوننت تغییر داد.

۹-۳) مشخصات کامپوننت

۹-۳-۱) **Language:** زبان برنامه نویسی کامپوننت را معین می‌کند و در برگه General قرار دارد.

۹-۳-۲) **Declarations:** در برگه Detail قرار دارد و شامل اعلانهای مکملی است که لازم می‌بینیم در تولید کد نهایی ظاهر شود.(مثلا #Includeهای اضافی در فایل C++)

۹-۳-۳) **Classes:** برای اینکه از یک کلاس کدی تولید شود ابتدا بایستی کلاس مزبور به یک کامپوننت نگاشت شود.

- این قسمت در برگه Realizes قرار دارد و می‌توان یک یا چند کلاس را از لیست انتخاب کرده و با استفاده از منوی میانبر Assign روی آن، آن کلاس‌ها را به کامپوننت نگاشت کرد.(در مورد زبان C++ هر دوی کامپوننت‌های CPP, H باید به کلاس مربوطه نگاشت شوند)

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- اگر بخواهیم ارتباط نگاشت فوق الذکر را قطع کنیم کافی است روی کلاسهای قبلا نگاشت شده کلید میانبر Remove Assignment را انتخاب کنیم.

۹-۴) اضافه کردن رابطه‌های Dependency در Componentها: تنها رابطه‌ای است که بین کامپوننت‌ها وجود دارد و اظهار میدارد که یک کامپوننت به کامپوننت دیگر وابسته است.

- این رابطه بصورت یک فلش خط چین نشان داده می‌شود. کامپوننتی که در ابتدای فلش قرار دارد به کامپوننتی که در انتهای فلش است وابسته می‌باشد.
- Dependency بر کامپایل تأثیر می‌گذارد بدین معنی که اگر A به B وابسته باشد تا زمانیکه B کامپایل نشده A نیز نمی‌تواند کامپایل شود.
- باید از Dependencyهای حلقه‌ای اجتناب کرد. زیرا مانع از تولید کد می‌شوند.
- برای اضافه کردن Dependency آیکون همنام را از نوار ابزار انتخاب کرده و بین دو کامپوننت می‌کشیم.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

نمای Deployment

نمای Deployment از استقرار فیزیکی برنامه کاربردی متأثر می‌شود که شامل موضوعاتی از قبیل طراحی و آرایش شبکه و مکان کامپونت‌ها بر روی شبکه می‌باشد.
برای هر سیستم فقط یک نمودار Deployment در هر مدل Rose وجود دارد.

۱۰-۱) نمودار Deployment: این نمودار تمام گره‌های موجود در شبکه، ارتباطهای بین آنها و فرآیندهایی که بر روی هر کدام از آنها اجرا خواهند شد را نشان می‌دهد.

۱۰-۲) افزودن پردازنده‌ها(Processor): یک پردازنده هر ماشینی است که قدرت پردازش دارد و میتواند به نمودار Deployment اضافه شود. سرویس دهنده‌ها(Servers)، ایستگاههای کاری(Workstations) و سایر ماشینهای دارای پردازنده در قالب پردازنده می‌گنجند.هر پردازنده دارای مشخصاتی است که در پنجره مشخصات می‌آید و عبارتست از:

۱۰-۲-۱) Characteristics(خصیصه‌ها): توضیحات فیزیکی پردازنده می‌باشد مثلا می‌تواند شامل سرعت و حافظه و... باشد.

۱۰-۲-۲) Scheduling(زمان‌بندی): نوع زمانبندی فرآیندهای استفاده شده توسط پردازنده جهت اجرا را نشان می‌دهد و شامل انتخاب‌های زیر می‌باشد:

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- **Preemptive:** نشان می‌دهد فرآیندهای با اولویت بالاتر می‌توانند فرآیندهای با اولویت پایین‌تر را متوقف کرده و خود اجرا شوند.
- **Non Preemptive:** نشان می‌دهد فرآیند با اولویت بالاتر نمی‌تواند فرآیند با اولویت پایین‌تر را متوقف کند و باید صبر کند تا دوره آن تمام شود.
- **Cyclic:** کنترل حلقه‌ای را نشان می‌دهد بدین معنی که به هر فرآیند یک Slice زمانی جهت اجرا تعلق می‌گیرد و همینطور فرآیند بعدی الی آخر.
- **Executive:** نشان می‌دهد که یک الگوریتم خاص نحوه زمانبندی را کنترل می‌کند.
- **Manual:** فرآیندها به وسیله کاربر خارج از سیستم زمانبندی می‌شوند.

۱۰-۲-۳) فرآیند: هر پردازنده می‌تواند شامل تعدادی فرآیند جهت اجرا باشد.

- برای اضافه کردن فرآیند جدید به پردازنده باید در برگه Detail و در کادر Processes با استفاده از منوی میانبر Insert پردازش یا فرآیند جدید را به پردازنده اضافه کرد.
- هر فرآیند دارای عنوان و اولویت(Priority) می‌باشد.
- برای حذف فرآیند باید از منوی میانبر(در اثر کلیک راست روی فرآیند) گزینه Delete را انتخاب کرد.
- فرآیندهای تعریف شده در مدل هم ذیل پردازنده اضافه می‌شوند.

۱۰-۳) اضافه کردن ابزارها(Devices): ابزار یک قطعه سخت افزاری بدون قدرت پردازش می‌باشد که شامل عناصری از قبیل پایانه‌ها، چاپگر، اسکنرها و... می‌باشد.

- برای اضافه کردن یک ابزار به نمودار Deployment کافی است آیکون Device را از نوار ابزار انتخاب کرده و در نمودار کلیک کنیم.
- هر ابزار دارای مشخصه Charaeteristics(در برگه Detail از پنجره مشخصات) است که توضیحات فیزیکی ابزار را بیان می‌دارد.

۱۰-۴) اضافه کردن رابطه‌ها: یک رابطه، یک پیوند فیزیکی بین دو پردازنده، دو ابزار یا یک پردازنده و یک ابزار می‌باشد. رابطه نوعا نشانگر اتصالهای فیزیکی شبکه بین گره‌های موجود در شبکه می‌باشد. رابطه هم چنین می‌تواند بیانگر یک پیوند اینترنتی بین دو گره باشد.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- برای اضافه کردن رابطه کافی است آیکون Connection را از نوار ابزار انتخاب کرده و بین دو پردازنده، دو ابزار یا یک پردازنده و یک ابزار بکشیم.
- هر رابطه دارای گزینه Characteristics(در برگه Detail از پنجره مشخصات) است که بیانگر مشخصات فیزیکی رابطه می‌باشد.(از قبیل پهنای باند، سرعت انتقال و...)

۱۰-۵) نشان دادن فرآیندها و نوع زمانبندی آنها در نمودار Deployment

- برای اینکه فرآیندهای یک پردازنده در کنار آن نمایش داده شوند کافی است از منوی میانبر(کلیک راست روی پردازنده) گزینه Show Processes را انتخاب کنیم.
- برای اینکه نوع زمانبندی فرآیندهای قابل اجرا روی پردازنده نشان داده شوند کافی است از منوی میانبر(کلیک راست روی پردازنده) گزینه Show Scheduling را انتخاب کنیم.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

مقدمه‌ای بر تولید کد(Code Generation)

در این فصل به مراحل اساسی که باید قبل از تولید کد برای مدل Rose آنها را انجام دهیم می‌پردازیم.
۶ مرحله برای تولید کد وجود دارد. تمام این مراحل در هر زبانی ضروری نیست اگر چه پیشنهاد می‌شود تمامی آنها در تمام زبانها اجرا شود.

۱۱-۱) چک کردن مدل: این مرحله ما را در پیدا کردن ناسازگاریها و مشکلات موجود در مدل که ممکن است تأثیر منفی بر کد بگذارد، یاری می‌کند.

- برای چک کردن مدل کافی است گزینه Check Model را از منوی Tools انتخاب کنیم، هر خطایی که پیدا شود در پنجره Log ظاهر میشود. خطاهایی که معمولا رخ میدهند عبارتند از:
  - پیغام‌های موجود در نمودارهای تعامل که به یک عملیات نگاشت نشده‌اند
  - آبجکت‌های موجود در نمودارهای تعامل که به یک کلاس نگاشت نشده‌اند.
- با استفاده از گزینه Show Access Violation در منوی Report(هنگام باز بودن نمودار کلاس) می‌توان مواردی که یک رابطه بین دو کلاس در بسته‌های مختلف وجود دارد اما هیچ رابطه‌ای بین خود بسته‌ها وجود ندارد و خطاهایی که اتفاق می‌افتد را پیدا کرد.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- برای بعضی از زبانهای برنامه‌نویسی می‌توان گزینه چک مخصوص آن زبان را نیز انتخاب کرد مثلا برای زبان جاوا گزینه Syntax Check این کار را انجام میدهد.

۱۱-۲) ایجاد کامپوننت‌ها: در این مرحله باید کامپوننت جهت نگهداری کلاسها ایجاد شوند.(یا از قبل در نمودار کامپوننت ایجاد شده باشند) در C++ می‌توان بدون این مرحله هم کار را پیش برد ولی Rose رأس کامپوننت‌ها را ایجاد نخواهد کرد.(در حالیکه در زبان جاوا اینکار را می‌کند)

۱۱-۳) نگاشت کلاسها به کامپوننت‌ها: این مرحله نیز در C++ الزامی نیست ولی بهتر است انجام شود.

۱۱-۴) تعیین خصوصیات تولید کد: تعدادی انتخاب در تولید کد وجود دارد که می‌توان آنها را برای کلاسها، صفات کامپوننت‌ها و سایر عناصر مدل تنظیم کرد. این خصوصیات چگونگی تولید شدن کد را کنترل می‌کند. تنظیمات پیش فرض معمول در Rose تعبیه شده است ولی میتوان آنها را تغییر داد. بطور مثال در C++ می‌توان کنترل کرد که آیا تابع Get برای هر صفت از کلاس تولید شود یا خیر.

- هر زبان در Rose تعدادی خصوصیات تولید کد دارد که در فصل بعد خصوصیات مورد نظر برای زبان C++ را بررسی می‌کنیم.
- برای دیدن خصوصیات تولید کد باید پنجره امکانات را باز کرده و سپس برگه زبان مورد نظر(مثلا C++) را انتخاب کنیم.
- در کادر Type می‌توان لیست کشویی را باز کرده و Attribute، Class، Operation و سایر عناصر مدل را انتخاب کرد. هر زبان در کادر لیست کشویی برای خود عناصر مدل متفاوتی دارد. هر عنصری که انتخاب می‌شود نیز دارای تنظیمات متفاوتی است که در لیست پایین ظاهر می‌شود.
- هر تغییری که در تنظیمات فوق الذکر داده شود بر روی تمام عناصر مدل در هنگام تولید کد تأثیر می‌گذارد. اگر بخواهیم محدوده تأثیرگذاری به یک کلاس خاص یا کامپوننت خاص محدود شود باید پنجره مشخصات مربوط به آن کلاس یا کامپوننت خاص را باز کرده و تغییرات لازمه را در برگه C++ اعمال کنیم.

۱۱-۴-۱) نسخه‌برداری از خصوصیات تولید کد: به جای انجام تغییرات به طور مستقیم بر روی مجموعه‌های پیش فرض خصوصیت

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

می‌توان از آنها نسخه‌برداری کرده و سپس تغییرات را بر روی کپی انجام داد.

- برای نسخه‌برداری کافی است دکمه Clone را در پایین کادر کشویی فشار داد. پنجره مدالی ظاهر میشود که نام مجموعه جدید را می‌پرسد و بعد از وارد کردن نام می‌توان مجموعه را ایجاد کرد.
- برای حذف نسخه‌های ایجاد شده از خصوصیات تولید کد کافی است مجموعه مورد نظر را از کادر کشویی Set انتخاب کرده و کلید Remove را فشار دهیم. مجموعه پیش‌فرض را نمی‌توان حذف کرد.

۱۱-۵) انتخاب کلاس، کامپوننت یا بسته جهت تولید کد: به هنگام تولید کد می‌توان یک کلاس، کامپوننت یا بسته خاصی را از پنجره مرورگر یا از طریق نمودار انتخاب کرد تا تولید کد درباره آن صورت گیرد. بسته مورد نظر می‌تواند ذیل نمای منطقی یا نمای کامپوننت باشد. تمامی کلاسها و کامپوننت‌های موجود در بسته تولید کد خواهد شد. هم چنین با استفاده از کلید Ctrl می‌توان همزمان به انتخاب چند کلاس، کامپوننت یا بسته جهت تولید کد مبادرت کرد.

۱۱-۶) تولید کد: در این مرحله برای هر زبان باید منوی مناسب را از منوی Tools انتخاب کرد مثلا برای زبان C++ استاندارد باید از منوی C++ گزینه Code Generationرا انتخاب کرد یا برای ساخت Project در Visual C++ به همراه کد تولید شده برای کلاسها کافی است از منوی Visual C++ گزینه Update Code را انتخاب کنیم.

برای نشان دادن یا مخفی کردن گزینه مربوط به زبانها در منوی Tools می‌توان آنها را در پنجره مدال Add-In Manager... از منوی Add-In تنظیم کرد.

۱۱-۷) چه چیزی تولید می‌شود: وقتی فرمان تولید کد صادر میشود Rose با استفاده از نمای منطقی و کامپوننت اطلاعات لازمه را جمع‌آوری می‌کند. آنچه که تولید می‌شود عبارتست از:

- کلاسها: تمامی کلاسهای موجود در مدل در کد تولید می‌شوند.
- صفات: شامل صفتهای هر یک از کلاسها می‌باشد به اضافه Type، Visibilityو مقدار پیش فرض آنها.
- Operation Signatures: عملیات‌ها به همراه تمامی پارامترها، نوع داده‌ای پارامتر و نوع بازگشتی عملیات.

Info@IRANMEET.COM
www.pupuol.com

WWW.IRANMEET.COM

- Relationships(رابطه‌ها): بعضی رابطه باعث ایجاد صفاتی در برخی کلاسها می‌شوند.
- کامپوننت‌ها: منجر به ایجاد فایل کد منبع می‌شود.
- بعد از تولید کد دو مرحله باقی می‌ماند اول تکمیل کد توسط برنامه‌نویسان و دوم طراحی واسط گرافیکی. Rose برای ما طراحی گرافیکی انجام نمی‌دهد.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

تولید کد C++ و Visual C++ و مهندسی معکوس در Visual C++

۱۲-۱) نحوه تولید کد در C++

۱۲-۱-۱) تولید کد بصورت C++ استاندارد: اگر بخواهیم تولید کد را بصورت C++ استاندارد داشته باشیم باید از قبل زبان برنامه‌نویسی کامپونت‌های مورد نظر را به C++ تغییر داده باشیم.(در غیر این صورت گزینه C++ در منوی میانبر کلاس یا کامپونت ظاهر نمی‌شود)

- بعد از تعیین C++ به عنوان زبان پیش فرض کامپونت کافی است روی کلاس(ها) یا کامپونت(های) مورد نظر منوی میانبر را باز کرده و گزینه Generate Code را از گزینه C++ انتخاب کنیم.
- برای دیدن فایل تولید شده میتوان از منوی میانبر، گزینه C++، گزینه‌های Browse Header یا Browse Body فایل‌های تولید شده CPP. و H. را مشاهده کرد.

۱۲-۱-۲) تولید کد بصورت Visual C++: اگر بخواهیم تولید کد را در Visual C++ داشته باشیم می‌بایستی ابتدا زبان پیش فرض تمامی کامپونت‌های مدل را به VC++ تغییر دهیم.

- در مرحله بعد از منوی Tools، منوی Visual C++، گزینه Update Code را انتخاب می‌کنیم. این انتخاب منجر به شروع یک Wizard می‌شود.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- بعد از گذر کردن از صفحه اولیه خوشامدگویی لیست کامپونتهایی که دارای زبان VC++ باشند ظاهر می‌شوند. هر یک از آنهایی را که بخواهیم در پروژه نهایی تولید شده دارای کدی باشد چک‌مارک می‌زنیم.
- در صورتیکه پروژه‌ای معرفی نشده باشد پنجره‌ای ظاهر می‌شود که از ما درخواست تعیین پروژه را می‌کند که می‌توان یکی از پروژه‌های از قبل ایجاد شده را انتخاب کرد یا همانجا یک پروژه جدید را در محیط نرم افزار Visual Studio ایجاد کرد و آنرا به پروسه تولید کد معرفی کرد.
- در انتهای پروسه تولید کد ممکن است پنجره‌ای ظاهر شود و سئوالی در مورد فایل‌های Dlg و Application موجود در پروژه که عنصر معادلی در مدل ندارند بپرسد که آیا حذف شوند یا خیر.

۱۲-۲) خصوصیات(Properties) تولید کد C++: تولید کد C++ با استفاده از Rose بسیار انعطاف پذیر است و بر چیزهایی که تولید می‌شوند و خیلی جزئیات کد تولید شده کنترل کامل داریم. تمام اینها از طریق خصوصیات تولید کد کنترل می‌شوند و ما می‌توانیم به دلخواه آنها را تغییر دهیم.

برای تغییر خصوصیات تولید کد باید در پنجره امکانات به برگه C++ مراجعه کنیم.(یا برای یک کلاس خاص از مسیر پنجره مشخصات کلاس، برگه C++ اقدام کنیم) هر عضو از لیست کشویی کادر Type مبین دسته‌ای از خصوصیات کد است که ذیلا به معرفی آنها می‌پردازیم.

۱۲-۲-۱) خصوصیات Project: به کل پروژه اشاره می‌کند و شامل چیزهایی از قبیل دایرکتوری پیش فرض که به هنگام تولید استفاده می‌شود و پسوند فایل‌هایی که انتخاب می‌شوند و... می‌شود.

جهت مشاهده لیست کاملی از این مشخصات به جدول ۱-۱۲، ص ۴۷۳ مراجعه شود.

۱۲-۲-۲) خصوصیات Class: خصوصیاتی هستند که بر کلاسها اعمال می‌شوند. بطور مثال به ما امکان می‌دهند نام کلاسها را عوض کنیم، معین کنیم که آیا توابع سازنده و مخرب ایجاد شوند یا خیر و...

جهت مشاهده لیست کاملی از این مشخصات به جدول ۲-۱۲، ص ۴۷۸ مراجعه شود.

۱۲-۲-۳) خصوصیات Attribute: خصوصیاتی هستند که به صفتها در هنگام تولید کد اشاره می‌کنند و شامل مسائلی از قبیل اینکه آیا تابع Get و Set داشته باشند یا خیر و... می‌باشد.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

جهت مشاهده لیست کاملی از این مشخصات به جدول ۳-۱۲، ص ۴۸۴ مراجعه شود.

۱۲-۲-۴) خصوصیات Operation: خصوصیات تولید کد مخصوص عملیات بوده و مسائلی از قبیل نام عملیات، مجازی بودن(Virtual) و... را پوشش می‌دهد.

جهت مشاهده لیست کاملی از این مشخصات به جدول ۴-۱۲، ص ۴۸۶ مراجعه شود.

۱۲-۲-۵) خصوصیات Module Specification: خصوصیاتی هستند که به فایل‌های H. که تولید می‌شوند مرتبط است. این خصوصیات می‌تواند شامل اینکه آیا فایل H. تولید شود یا خیر و اینکه آیا اخطار Copyright در فایل H. درج شود یا خیر و... می‌شود.

جهت مشاهده لیست کاملی از این مشخصات به جدول ۵-۱۲، ص ۴۸۸ مراجعه شود.

۱۲-۲-۶) خصوصیات Module Body: خصوصیاتی است که به فایل‌های CPP. تولید شده وابسته هستند. مثلا اینکه فایل Cpp. تولید شود یا خیر و آیا اخطار Copyright در فایل Cpp. درج شود یا خیر را شامل می‌شود.

جهت مشاهده لیست کاملی از این مشخصات به جدول ۶-۱۲، ص ۴۹۱ مراجعه شود.

۱۲-۲-۷) خصوصیات Association: خصوصیاتی از تولید کد است که با رابطه‌های Association سروکار دارد. در C++ فقط یک خصوصیت برای Association وجود دارد.

۱۲-۲-۸) خصوصیات Role: خصوصیاتی است که بر روی کد تولید شده ناشی از رابطه‌ها تأثیر می‌گذارد. نام صفتی که می‌خواهد ایجاد شود و کلاس Container صفت مثالهایی از این دسته هستند.

جهت مشاهده لیست کاملی از این مشخصات به جدول ۷-۱۲، ص ۴۹۴ مراجعه شود.

۱۲-۲-۹) خصوصیات Aggregation(با نام Has در لیست کشویی می‌آید): خصوصیاتی است که با رابطه‌های Aggregation در تولید کد نهایی مرتبط است.

جهت مشاهده لیست کاملی از این مشخصات به جدول ۸-۱۲، ص ۵۰۲ مراجعه شود.

۱۲-۲-۱۰) خصوصیات Dependency: کنترل می‌کند که چگونه رابطه‌های Dependency در C++ به کار گرفته شود. لیست خصوصیات Dependency به دو عدد محدود می‌شود.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

جهت مشاهده توضیحات لیست خصوصیات به جدول ۹-۱۲، ص ۵۰۶ مراجعه شود.

۱۲-۲-۱۱) خصوصیات Subsystem: آن دسته از خصوصیات است که بر بسته‌های نمای کامپوننت موجود در مدل اعمال می‌شود.

جهت مشاهده لیست کاملی از این خصوصیات به جدول ۱۰-۱۲، ص ۵۰۷ مراجعه شود.

۱۲-۲-۱۲) خصوصیات Class Category: آن دسته از خصوصیات است که بر بسته‌های نمای منطقی موجود در مدل اعمال می‌شود.

جهت مشاهده لیست کامل از این خصوصیات به جدول ۱۱-۱۲، ص ۵۰۷ مراجعه شود.

۱۲-۲-۱۳) خصوصیات Generalitation: اشاره به تولید شده ناشی از رابطه توارث است و فقط یک عدد می‌باشد.

۱۲-۳) نحوه تولید کد در Visual C++: نحوه تولید کد در C++ و Visual C++ تا حدود زیادی با هم متفاوت است. ابتدا اشاره مختصری به این تفاوتها کرده و سپس به جزئیات بحث تولید کد در VisualC++ می‌پردازیم.

- خروجی تولید کد Visual C++ در یک پروژه ظاهر میشود در حالیکه در C++ به صورت فایل‌های منفرد دیده می‌شود.
- خصوصیات تولید کد در Visual C++ بسیار محدودتر از C++ است. بسیاری از این خصوصیات و امکانات در Model Assistant تعبیه شده که به آن خواهیم پرداخت.
- کد تولید شده نیز تفاوتهای زیادی با هم دارند بطور مثال در C++ قطعات کد باید ما بین دو خط Comment ایجاد شده توسط Rose قرار گیرد در حالیکه در Visual C++ فقط یک خط Comment بالای آن می‌آید. هم چنین صفحات کد C++ بسیار شلوغتر و تعداد مشخصات از مدل که در کد بصورت توضیح آمده‌اند بسیار بیشتر است.
- در C++ برای هر کلاسی که ایجاد می‌کنیم(یا برای دسته‌ای از کلاسها) باید یک کامپوننت ایجاد کرده و به آن نگاشت کنیم که کامپوننت متناظر باید از نوع Package Specification یا Package Body باشد در حالیکه در Visual C++ فقط کافی است یک کامپوننت از هر نوع تعریف شده و همه کلاسها به آن نگاشت شوند.
- در C++ کد تولید در فایل همنام کامپوننت قرار می‌گرفت در حالیکه در Visual C++ در فایل همنام کلاس قرار می‌گیرد.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- شاخه‌بندی کلاسها و کامپوننت در نمای منطقی یا کامپوننت مستقیما در کد تولید شده بصورت شاخه‌بندی متناظر منعکس می‌شد در حالیکه در Visual C++ این گونه نیست.

۱۲-۳-۱) تولید کد: قبل از اینکه بتوان تولید کد را انجام داد ابتدا باید یک کامپوننت از نوع اجرایی ایجاد کرد و زبان برنامه‌نویسی آنرا VC++ قرار دارد. مرحله بعد نگاشت کردن کلیه کلاسها به این کامپوننت است.

- برای شروع تولید کد می‌توان از منوی Tools، منوی Visual C++، گزینه Upadte Code را انتخاب کرد. در صورتیکه اولین باری باشد که این عمل را انجام می‌دهیم یک پنجره Wizard ظاهر می‌شود.
- در پنجره Wizard فوق الذکر این امکان وجود دارد که ارتباط بین کامپوننت را با یک پروژه Visual C++ برقرار کنیم.(در صورت لزوم از همانجا بتوانیم آنرا ایجاد کنیم)
- در مرحله بعد میتوانیم کلاسهای مورد نظرمان را جهت تولید کد انتخاب کنیم.
- اگر فقط بخواهیم کد یک کلاس خاص را تولید کنیم کافی است از منوی میانبر گزینه Update Code را انتخاب کنیم.
- هر گاه کلاس جدیدی به مدل اضافه شده یا از آن حذف شود هنگام تولید کد پنجره‌ای مبنی بر اضافه یا حذف کردن کلاس مورد نظر به پروژه نهایی ظاهر می‌شود و از کاربر سئوال می‌کند.
- در صورتیکه کلاسهایی وجود داشته باشند که به هیچ کامپوننتی نگاشت نشده باشند می‌توان این عمل را در پنجره Wizard تولید کد نیز انجام داد.

۱۲-۳-۱-۱) دستیار مدل(Model Assistant): در صورتیکه از طریق منوی میانبر روی یک کلاس گزینه Model Assistant را انتخاب کنیم پنجره‌ای ظاهر می‌شود که دارای امکانات زیادی برای تولید کد مطلوب است که ذیلا به مهمترین آنها اشاره می‌شود.

- امکان دادن فرمان تولید کد از طریق کلید Update Code.
- امکان تغییر نوع کلاس به Union، Typedef، Enum، Struct، Class که در این صورت ساختار دیگر قسمتهای پنجره متناسب با نوع کلاس تغییر می‌کند. این عمل از طریق تغییر در لیست کشویی Class Type امکان پذیر است.
- امکان اضافه کردن Enumهای جدید، Typedefهای جدید و کلاس‌های داخلی جدید به این کلاس.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- امکان انتخاب توابع اپراتوری =، == و...جهت تولید کد.
- امکان انتخاب توابع سازنده و مخرب جهت تولید کد.
- امکان اضافه کردن صفات و عملیات و هم چنین اضافه کردن Accessorهای صفات.
- در مورد کلاسهایی که از کلاسهای موجود در MFC ارث می‌برند امکان اضافه کردن توابع جهت Override، پیاده‌سازی هندلرهای پیغام(Message Handler) در مورد کلاسهای پنجره و هم چنین هندلرهای دستورات(Command Handler) وجود دارد.
- می‌توان یک جستجوی متنی هم در محتویات کلاس یا کلاسهای MFC انجام داد.
- می‌توان الگوهای از قبل آماده شده با نام Code Template را به کلاس اعمال کرد که در این صورت مشخصاتی که در الگوی مزبور تعریف شده یکجا به این کلاس منتقل می‌شود. کاربرد این الگوها زمانی است که بخواهیم یکسری مشخصات تکراری را به کلاسهای مختلف اعمال کنیم.
- تعیین Visibility مربوط به کلاس، صفات و عملیات آن نیز از دیگر امکانات این پنجره است.
- می‌توان کد بدنه توابع(عملیات) را نیز در این پنجره وارد کرد.

۱۲-۳-۲) مهندسی معکوس(Reverse Engineering): در مهندسی معکوس تغییرات صورت گرفته در کد به مدل برگردانده می‌شود و باعث می‌شود مدل با کد به هنگام شود.

- اعمال مهندسی معکوس در Visual C++ با استفاده از منوی Tools، منوی Visual C++، گزینه Update Model From Code... برای کل پروژه یا استفاده از منوی میانبر، گزینه Update Model برای یک کلاس خاص انجام می‌شود.
- در صورتیکه کلاسها یا صفات یا عملیات جدیدی در پروژه تعریف شده باشند که قبلا در مدل وجود نداشته‌اند یا بر عکس کلاسها، صفات و عملیاتی که قبلا در مدل وجود داشته‌اند و از پروژه حذف شده‌اند، پیغامهایی ظاهر می‌شود و در این موارد از کاربر سئوال می‌کند.

۱۲-۳-۳) برخی پیکربندی‌ها: می‌توان تا حدود زیادی در روال تولید کد و مهندسی معکوس اعمال مدیریت کرد و پیکربندی آن را به نحو مطلوب تغییر داد.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

- اگر از منوی منوی Tools، منوی Visual C++، گزینه Property را انتخاب کنیم پنجره‌ای ظاهر میشود که دارای ۵ برگه اصلی است:
  - برگه اول با نام Code Update پیکربندی گزینه‌های مربوط به تولید کد را بعهده دارد.
  - برگه دوم با نام Model Update پیکربندی گزینه‌های مربوط به مهندسی معکوس را بعهده دارد.
  - برگه سوم با نام Containers است که در این برگه می‌توان کلاسهای ظرف جدید را معرفی کرد.
  - برگه چهارم با نام Class Operations که با استفاده از آن می‌توان توابع اپراتوری استاندارد را تعریف کرد. این توابع از این به بعد قابل انتخاب به عنوان اپراتورهای کلاس(با استفاده از پنجره Model Assistant) می‌باشند.
  - برگه پنجم با نام Accessors که نحوه عملکرد توابع Get و Set صفات کلاسها را برای ما قابل برنامه‌ریزی می‌کنند.
- می‌توان آخرین تغییرات ایجاد شده در مسیر تولید کد را با استفاده از منوی منوی Tools، منوی Visual C++، گزینه Undo Last Code Update ملغی کرد و کد را به آخرین کد ما قبل تغییر برگرداند.
- بعضی امکانات در منوها و جایگاههای مختلف برنامه وجود دارد که منحصر به آبجکت‌های Com و Atl می‌باشد که در این نوشتار به آنها اشاره‌ای نشده است.

Info@IRANMEET.COM
www.pupuol.com

پوپول مرجع دانشگاه ومدرسه
WWW.IRANMEET.COM

ماخذ

۱) کتاب مرجع کامل UML With Rational Rose ترجمه شده توسط مهندس مهرداد توانا و مهندس عاطفه شیجونی، انتشارات ساحر
۲) Rational Rose 2002 Help

Info@IRANMEET.COM
www.pupuol.com