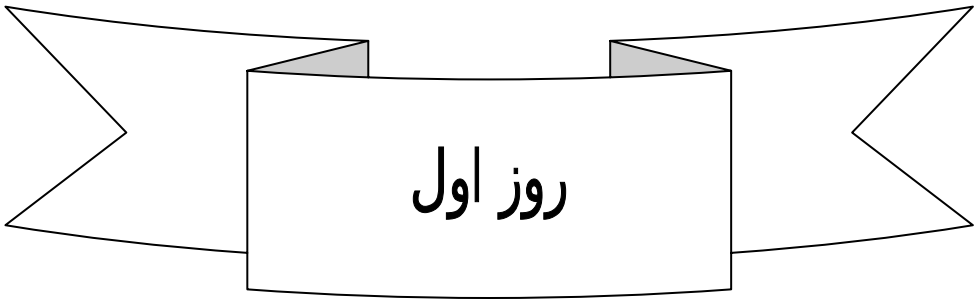
خودآموز UML در شش روز

فهرست مطالب

مقدمه:

در این دوره با دلیل استفاده از UNL در توسعه برنامه هاي نرم افزاری و نمودارهای مورد استفاده در آن و روش ترسیم آنها آشنا خواهید شد. همچنین کتابخانه دیجیتالی بعنوان یک سیستم برای درک بهتر شما از UML مورد بررسی قرار خواهد. همچنین در پایان برخی دروس بخش پرسش و پاسخ و کارگاه وجود دارد تا بتوانید و ضعیت یادگیري خود را بهتر بسنجید.



امروز مطالب زير را ياد خواهيد گرفت:

- ✓ زمان مدلسازي يكپارچه چيست
- ✓ مدلسازي سيستم
- ✓ راهنماي مهندسي سريع نرم افزار(GRAPPLE)
- ✓ اجزاي UML

معرفيUML (زبان مدلسازي يكپارچه)

از آنجاييكه جهان بسيار پيچيده شده است، سيستمهاي كامپيوتري كه در جهان بكار گرفته ميشود بايستي پيچيده باشد. اين سيستمها اغلب درگير قسمتهاي متعدد سخت افزار، نرم افزار، شبكه هاي راه دور، ارتباط به پايگاه داده هايي كه كوهي از اطلاعات را در بر دارد، ميباشند. اگر شما مي خواهيد سيستمي كه چنين عمل ميكند بسازيد، چگونه با اين پيچيدگيها برخورد مي كنيد؟

كليد اين كار اين است كه روند طراحي را به طريقي سازماندهي كنيد كه مشتريان، تحليلگران ، برنامه نويسان و افراد ديگري كه درگير اين سيستم هستند بتوانند بفهمند و با آن موافق باشند. UML اين سازماندهي را ارائه ميدهد.

فرض كنيد: آيا شما ميتوانيد به يک پيمانكار ساختمان بگوييد كه شما يک خانه ٤ خوابه با ٣ حمام، با مساحتي حدود ٢٠٠٠ فوت مربع ميخواهيد، اين را براي من بساز! آيا ما مي توانيم جزييات را طي كار بسازيم؟ همه ما ميدانيم كه اين چيز مضحكي است. متاسفانه اين روش توسعه(development) در صنعت نرم افزار بسيار رايج است. فقط هنگامي كه شما با يک معمار براي طراحي يک blueprint كه نوعي نقشه است و نمودار واقعي چگونگي ساخت خانه را ميدهد، كار ميكنيد، مانند ما روي نمودارهاي UML كه به درستي اين موضوع كه چگونه سيستم نرم افزار سفارش شما ساخته خواهد شد را مستند مي كند، كار خواهيد نمود.

UML در سال ١٩٩٧ بعنوان يک روش براي ترسيم نمودار طراحي نرم افزار گسترش پيدا كرد و با مشاركت بهترين افكار در تحليل و طراحي شئ گرا(object oriented) طراحي گرديد. اين موضوع به مراتب بسيار مهيج ترين رخداد صنعت نرم افزار در سالهاي اخير است. هر نظام مهندسي يک روش استاندارد براي مستندسازي دارد. مهندسين الكترونيک نمودار طرح كلي (schematic diagrams)، مهندسين معماري و مكانيک blueprints و نمودار مكانيكي(mechanical diagrams) دارند. در حال حاضر صنعت نرم افزار UML را در اختيار دارد.

پيش از رفتن به درس بعد برخي مزاياي UML را ملاحظه نماييد.

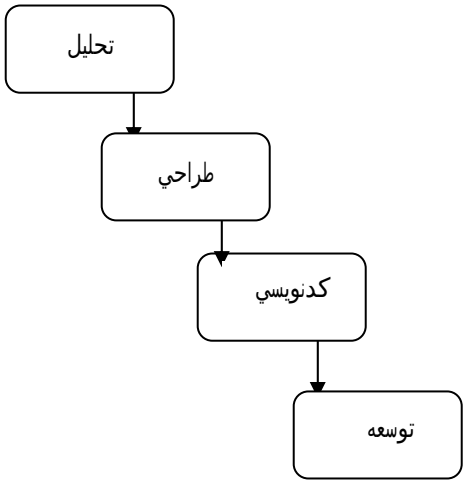
١. سيستم نرم افزار شما بطور حرفه اي طراحي ميشود و قبل از كدنويسي مستند ميگردد. پيشاپيش شما به درستي خواهيد فهميد كه چه چيزي بدست خواهيد آورد.
٢. پس از بدست آوردن اولين طرح سيستم، كد نويسي به راحتي تشخيص داده ميشود و كد نويسي با كارآيي بسيار بالايي انجام ميگيرد. شما هزينه توسعه كمتري خواهيد داشت.
٣. وقفه هاي منطقي در طرحهاي طراحي شده قابل تشخيص خواهد بود. نرم افزار شما همان طور كه انتظار داريد رفتار خواهد نمود. موارد غيره منتظره كمتري وجود خواهد داشت.
٤. تمام طراحي سيستم ميتواند راه توسعه نرم افزار را ديكته كند. تصميمات درست قبل از درگير شدن با كدهاي ضعيف گرفته خواهند شد. مجدداً تمام هزينه ها كمتر خواهد شد.
٥. UML به ما اجازه مي دهد كه يک تصوير كلي ببينيم. ما ميتوانيم سيستمهايي با حافظه بيشتر و كارآمد توسعه دهيم.
٦. هنگامي كه براي انجام اصلاحات در سيستم تان برمي گرديم، اين كار بسيار آسانتر است كه روي سيستمي كار كنيم كه مستندات UML دارد. بازآموزيهاي بسيار كمتري اتفاق مي افتد. هزينه هاي نگهداري سيستم شما پايين تر خواهد بود.
٧. اگر شما نياز پيدا كرديد با توسعه دهندگان نرم افزار ديگري كار كنيد، نمودارهاي UML به آنها اجازه ميدهد تا به سرعت بيشتري در سيستم سفارشي ما دست پيدا كنند. به اين مورد مانند طرح كلي يک راديو فكر كنيد. تكنولوژي چگونه ميتواند بدون داشتن آن طرح دستگاه را درست كند؟
٨. اگر ما نياز داشته باشيم با پيمانكاران خارجي و يا حتي با برنامه نويسان خود ارتباط برقرار كنيم، اين كار بطور موثرتري انجام گيرد.

استفاده از زبان مدلسازي يكپارچه سبب كاهش هزينه كلي، ايجاد نرم افزار مطمئن تر و كارآمدتر و يك ارتباط بهتر بين تمام بخشهاي درگير ميشود. نرم افزارهايي كه با UML مستند شده اند به طور مؤثرتري قابليت اصلاح دارند و به اصطلاح، نرم افزار شما آينده خواهد داشت.

مدلسازي سيستم، روش قديمي در برابر روش جديد

در اين دوره ما درباره سيستمها و يا بخشهايي از سيستمهاي بزرگتر بحث خواهيم نمود.
سيستم، تركيبي از نرم افزار و سخت افزار است كه يک راه حل براي مشكلات كسب و كار ارائه ميدهد.

در روند توسعه سيستم افراد زيادي درگير هستند. اول همه مشتري است، فردي كه مشكلاتي براي حل كردن دارد. يک تحليلگر مشكلات مشتري را مستند ميكند و آن به توسعه دهندگان ميدهد، برنامه نويسان كه نرم افزاري كه مشكل را حل مي كند ميسازند، آن را تست ميكند و روي سخت افزار پياده ميكنند. اين مورد بسيار لازم است زيرا سيستمها امروزه بسيار پيچيده هستند، دانش ويژه شده است بطوريكه يک فرد نميتواند تمام امور يک كسب و كار را بداند، مشكل را بفهمد، راه حل طراحي كند، آن را به برنامه تبديل نمايد، برنامه را روي سخت افزار پياده سازي نمايد و مطمئن باشد كه اجزاي سخت افزار همه به درستي با يكديگر كار ميكنند.



روش آبشاري براي مدلسازي سيستم

- روش قديمي مدلسازي سيستم كه بعنوان مدل آبشاري شناخته شده است، مشخص ميكند كه تحليل، طراحي، كدنويسي و توسعه به دنبال يكديگر مي آيند. تنها هنگامي كه يكي كامل

گردید، مورد بعدي ميتواند شروع شود. اگر يک تحليلگر، تحليل را به طراح، کسي که طراحي را به توسعه دهنده ارائه ميدهد، ارائه دهد، فرصتها اينچنين هستند که اعضاي سه تيم کار را به يکديگر ميدهند و اطلاعات مهم را به اشتراک ميگذارند. معمولاً طرفداران روش آبشاري به کدنويس، بيشترين زمان از پروژه را ميدهند، اين مورد، زمان ارزشمندي از تحليل و طراحي را صرف ميکند.

- در روش جديد، مهندسي نرم افزار معاصر بر دنبال کردن تاثير متقابل بين مراحل توسعه تاکيد دارد. تحليلگران و طراحان، براي مثال، براي ايجاد پايه و اساسي محکم براي برنامه نويسان تلاش ميکنند. برنامه نويسان، به نوبت با تحليلگران و طراحان براي به اشتراک گذاشتن بينش آنها، اصلاح طراحيها، و تقويت کدنويسي تعامل دارند. مزيت آن اين است که نظر به رشد ادراکها، تيم ايده هاي جديد را يکي ميکند و يک سيستم قويتري را ميسازد.

Rapid Application Development = RAD(توسعه سريع نرم افزار)

RAD از ۵ بخش تشكيل شده است:

۱. جمع آوري نيازمنديها
۲. تحليل
۳. طراحي
٤. توسعه
۵. استقرار

- **جمع آوري نيازمنديها**

اين بند شامل فعاليتهاي متعددي است. قبل از اقدام روي يک فعاليت، بسيار مهم است که بدانيد اگر شما نفهميده ايد که مشتري چه ميخواهد، هرگز نخواهيد توانست سيستم درستي بسازيد.

كشف فرآيندهاي كسب و كار: در اينجا تحليلگران با مصاحبه با مشتري و يا يك فرد آگاه كه معرفي ميگردد و به مصاحبه براي رسيدن مرحله به مرحله به فرآيندهاي مربوط پاسخ ميدهد، دركي از فرآيندهاي كسب و كار بدست مي آورند.

انجام تحليل قلمرو: تحليلگران با هدف درك موجوديتهاي اصلي در حوزه مشتري با او مصاحبه ميكنند. در طي مكالمات بين تحليلگر و مشتري، يكي ديگر از اعضاي تيم يادداشت برداري ميكند. مدلساز اشياء به اسامي گوش ميدهد و با ايجاد يك كلاس براي هر اسم، كار خود را آغاز ميكند. در آخر برخي اسامي جزء صفات خواهند شد. همچنين باید به افعال كه عمليات يك كلاس خواهند شد،گوش دهد .
محصول: نمودار كلاس سطح بالا و مجموعه اي از يادداشتهاي ملاقاتها.

شناسايي سيستمهاي همكار: اول همه در فرآيند، تيم توسعه درمي يابد كه چه سيستمهايي به سيستم جديد وابسته هستند و اين سيستم به چه سيستمهايي وابسته است. يك مهندس سيستم به اين عمل توجه ميكند.
محصول: نمودار استقرار(Deployment Diagram)

كشف نيازمنديهاي سيستم: در اين عمل تيم اولين جلسه توسعه نرم افزار مشترک خود را مرور ميكند. (joint application development=JAD) . اين جلسه تصميم گيرندگان سازمان مشتري، كاربران بالقوه و اعضاي تيم توسعه را گرد هم مي آورد. مسوول جلسات جلسه را اداره ميكند. كار مسوول جلسات آنچه كه مد نظر تصميم گيرندگان و كاربران از سيستم است را استنباط ميكند. در پايان اعضاي دو تيم بايد يادداشت بردارند و مدلساز اشياء بايد نمودارهاي كلاسي كه قبلاً بيرون آمده را تصحيح نمايد.
محصول: نمودار بسته(Package Diagram)

ارائه نتايج به مشتري: هنگاميكه تيم تمام فعاليتهاي لازم را به پايان رساند، مدير پروژه نتايج را به مشتريان ارائه ميكند.

- **تحليل**

حال تيم وارد نتايج بخش نيازمنديها گرديده و درك خود از مشكلات را افزايش ميدهد. در حقيقت برخي فعاليتها در طي بخش نيازمنديها در حين اينكه مدلساز اشياء تصحيح نمودار كلاس را آغاز ميكند، شروع ميشود.

درك كاربردهاي سيستم: در يك جلسه JAD با كاربران بالقوه ، تيم توسعه با كاربران جهت پيداكردن actorي كه هر مورد كاربرد را آغاز ميكند و actor ي كه از آن موردهاي كاربرد استفاده مينمايد، كار ميكند.(actor ميتواند يک سيستم و يا يک شخص باشد).
محصول: نمودارهاي مورد كاربرد(Use case Diagram)

استخراج موردهاي كاربرد: بياييد كار با كاربران را ادامه دهيم. هدف، تحليل توالي مراحل در هر مورد كاربرد است.
محصول: توضيح متني از مراحل هر نمودار مورد كاربرد.

تصحيح نمودارهاي كلاس: مدلساز اشياء در طي جلسه JAD به تمام مكالمات گوش ميدهد و نمودارهاي كلاس را تصحيص مينمايد. او بايد نام association، كلاسهاي مجرد، چندتايي(multiplicities)، عموميت دادن و تجمع (aggregation) را تكميل نمايد.
محصول: نمودار كلاس تصحيص شده.(Refine class Diagram)

تحليل تغييرات در حالت در اشياء : همچنين مدلساز اشياء مدل را با نشان دادن تغييرات حالت در هر جا كه لازم باشد تصحيح مينمايد.
محصول: نمودار حالت(State Diagram)

تصحيح تعاملات بين اشياء : اكنون تيم توسعه مجموعه اي از موردهاي كاربرد را دراختيار دارد و نمودارهاي كلاس را تصحيح نمايد. اين زمان تصحيح چگونگي تعامل اشياء است. مدلساز اشياء مجموعه اي از نمودارهايي كه تغييرات حالت را دربردارد را توسعه ميدهد.
محصول: نمودارهاي توالي و همكاري(Sequence and Colaboration Diagram)

تحليل يكپارچگي با سيستمهاي مشترک: مهندس سيستم در تمام مراحل قبلي بطور موازي جلو ميرود، جزئيات دقيق يكپارچگي بين سيستمها را مشخص ميكند. چه نوع ارتباطي را در بر ميگيرد؟ معماري شبكه چيست؟ اگر سيستم بايد به database ي دسترسي داشته باشد، آن چه نوعي است.
محصول: نمودار ريز شده استقرار و در صورت لزوم Data model

- **طراحي**

در اين بخش، تيم با نتايج بخش تحليل براي طراحي راه حل كار ميكند. تحليل و طراحي بايد پيش و پس برود تا طرح تكميل گردد.

توسعه و تصحيح نمودار اشياء : برنامه نويسان نمودارهاي كلاس را ميگيرند و هر نمودار شئ را كه لازم باشد با امتحان هر عمليات و توسعه نمودارهاي فعاليت متناظر، توليد ميكنند. اين نمودارهاي فعاليت بعنوان ماخذ بسياري از كدنويسيها در بخش توسعه بكار ميروند.
محصول: نمودارهاي فعاليت.(Activity Diagram)

توسعه نمودارهاي اجزاء : در اين اقدام برنامه نويسان نقش مهمي ايفا ميكنند. كار مهم در اينجا تجسم اجزايي است كه نتيجه ي بخش بعد هستند و وابستگي بين آنها را نشان ميدهد.
محصول: نمودار اجزاء.(Component Diagram)

برنامه براي استقرار: پس از بدست آوردن نمودارهاي اجزاء مهندس سيستم شروع به برنامه ريزي براي پياده سازي و يكپارچه كردن سيستمهاي مشترک مينمايد. نمودارهاي ايجاد شده نشان ميدهد كه در كجا اجزاء مستقر خواهند شد.
محصول: بخشي از نمودار استقرار توسعه اوليه.

طراحي و نمونه اوليه رابط كاربر(user interface) : اين مورد وارد جلسه JAD ديگري با كاربران درپي جلسات قبلي ميگردد. اين مورد نشان معمولي از ارتباط بين تحليل و طراحي است. تحليلگر رابط گرافيكي كاربران، با كاربران براي توسعه نمونه اوليه كاغذي صفحه اي كه موردهاي كاربرد را به هم مرتبط مينمايد، كار ميكند.
محصول: عكسي زا صفحه نمايش از صفحه نمايش اوليه

تستهاي طراحي: ترجيحاً يک توسعه دهنده يا يک كارشناس تست خارج از تيم توسعه نمودارهاي مورد كاربرد را براي توسعه سند تست براي ابزارهاي تست خودكار استفاده مينمايد.
محصول: سند تست.(Test scripts)

آغاز مستندسازي: متخصص مستندسازي با طراحان جهت شروع نوشتن گزارشات و رسيدن به يک ساختار سطح بالا براي مستندات كار ميكند.
محصول: ساختار مستند.(Documant structure)

- **توسعه**

با تحليل و طراحي كافي، اين بخش بايد به سرعت و بدون ايراد پيش برود. در اين بخش برنامه نويسان درگير ميشوند.

ايجاد كد: با نمودارهاي كلاس، شئ، فعاليت و اجزاي در دست اقدام، برنامه نويسان كدها را براي سيستم ايجاد مينمايند.
محصول: كد.(The code)

كد تست: اين عمل به عمل بعدي و بالعكس تا زمانيكه كه كد تمام مراحل تست را بگذراند بازخورد ميدهد.
محصول: نتايج تست.

ايجاد رابطهاي كاربر، ارتباط دادن به كد و تست: اين كار رابطه اي بين طراحي و توسعه است. در اينجا يک متخصص رابط گرافيكي كاربران(GUI) ، نمونه اوليه رابط آزمايش شده را ايجاد ميكند و آنها را به كد ارتباط ميدهد. تست رابط ين اطمينان را ايجاد مينمايد كه رابط ها به درستي كار ميكنند.

محصول: سيستم در حال كار، تكميل شده با رابطهاي كاربر.

تكميل مستندسازي: در طي بخش توسعه ، كارشناسان مستندسازي بطور موازي با برنامه نويسان براي تكميل همه مستندات كار ميكنند.
محصول: مستندات سيستم.(System documentation)

- **استقرار**

هنگاميكه توسعه تكميل گرديد، سيستم روي سخت افزار مناسب پياده سازي ميشود و با سيستمهاي مشترک يكپارچه ميگردد.

برنامه ريزي براي پشتيباني و بهبود: اين عمل ميتواند بسيار زودتر از بخش توسعه آغاز گردد. مهندس سيستم يک برنامه براي مراحل پيگيري خرابي سيستم ايجاد مينمايد.
محصول: برنامه بهبود خرابي.

نصب سيستم تمام شده بر روي سخت افزار مناسب: اين مرحله توسط مهندس سيستم با هر كمک لازمي از سوي برنامه نويسان انجام ميگيرد.
محصول: سيستم تماماً استقرار يافته.

تست سيستم استقرار يافته: پس از نصب سيستم بر روي كامپيوتر(ها)ي مناسب، تيم توسعه سيستم نصب شده را تست ميكنند. آيا همانطور كه تصور ميكرديد كار ميكند؟ برنامه پشتيباني و بهبود كار ميكند؟ نتايج اين تستها مشخص ميكند كه آيا تصحيحات بيشتري مورد نياز است يا خير.
محصول: نتايج تست.(Test results)

تجليل: هنگاميكه تمام كارها به پايان رسيد، تيم توسعه ميتوانند موفقيت خود را جشن بگيرند.

شالوده روند توسعه، GRAPPLE **(Guidelines for Rapid Application engineering)** است، كه شامل 5 بخش است: جمع آوري نيازمنديها، تحليل، طراحي، توسعه و پياده سازي. هر بخش شامل شماري از فعاليتها است و هر فعاليت يک محصول كاري دارد. نمودارهاي UML محصولات كاري بسياري از فعاليتها هستند.

اجزاي UML

UML تعدا معيني از عناصر گرافيكي را در نمودارها تركيب ميكند. از آنجاييكه اين يک زبان است، UML قوانيني براي تركيب اين عناصر با هم دارد.
هدف از نمودارها ارائه ديد چندگانه از سيستم است و اين مجموعه ديدهاي چندگانه، مدل ناميده ميشود.


مدلUML آنچه كه يک سيستم براي انجام دادن فرض ميكند را شرح ميدهد. چگونگي اجراي سيستم را نمي گويد.

بياييد نگاه خلاصه اي به نمودارهاي UML بيندازيم. UML شامل ۹ نمودار پايه است، اما به خاطر داشته باشيد كه تركيب اين نمودارها نيز امكان پذير است.

نمودار كلاس(Class Diagram)

همه چيز بطور طبيعي در يک طبقه بندي قرار ميگيرد (كامپيوترها، اتومبيلها، درختان و ...). ما به اين طبقه بنديها كلاسها را نسبت ميدهيم.


يک كلاس، طبقه بندي از گروهي از چيزهايي است كه صفات مشابه و يا رفتار مشترک دارند.
نمودار كلاس نمايشهايي كه توسط توسعه دهندگان بكار ميرود را ارائه ميدهند.

نمودار شئ (Object Diagram)


يک شئ مثالي از يک كلاس و چيز خاصي است كه داراي ارزش ويژه اي از صفات و رفتار است.

نمودار مورد كاربرد(Use Case Diagram)

مورد كاربرد توضيحي از رفتار سيستم از جايگاه كاربر است. براي توسعه دهندگان سيستم، اين مورد يک ابزار ارزشمند است: اين مورد يک فناوري درست و ساعي براي جمع آوري نيازمنديهاي سيستم از ديدگاه كاربر است. اگر هدف ساختن يک سيستم است كه افراد واقعي بتوانند از آن استفاده نمايند، اين مورد بسيسار مهم است. در نمايشهاي گرافيكي يک نماد براي actor بكار ميرود.


Actor موجوديتي است كه يک مورد كاربرد را آغاز ميكند و ميتواند يک شخص يا سيستم ديگري باشد.

نمودار حالت (State Diagram)

در هر زمان که در نظر بگیریم، یک شئ در یک حالت بخصوص است. نمودارهاي حالت این حالات و تغییرات آنها را در طي زمان نشان میدهند. هر نمودار حالت با نمادي که حالت شروع را نشان میدهد آغاز میگردد و با نمادي که حالت پایاني را نشان میدهد خاتمه مي یابد. براي مثال هر فردي میتواند یک نوزاد، کودک، نوجوان یا جوان باشد.

نمودار توالي(Sequence Diagram)

نمودارهاي کلاس و نمودارهاي شئ اطلاعات استاتیک را نمایش میدهد. در یک سیستم درحال کار، به هر حال اشیاء با یکدیگر در تعامل هستند و این تعاملات در طي زمان رخ میدهد. نمودار توالیUML تعاملات پویا بر اساس زمان را نشان میدهد.

نمودار فعالیت(Activity Diagram)

فعالیتهایي که در محدوده یک مورد کاربرد یا در رفتار شئ رخ میدهد با یک توالي صورت میگیرد. این توالي با نمودار فعالیت نشان داده میشود.

نمودار همکاري(Collaboration Diagram)

عناصر یک سیستم با یکدیگر براي به انجام رساندن مقاصد سیستم کار میکنند و یک زبان مدلسازي باید راهي براي نشان دادن این مورد داشته باشد. نمودار همکاري UML به این دلیل طراحي شده است.

نمودار اجزاء(Component Diagram)

امروزه در مهندسي نرم افزار براي توسعه به صورت تیمي تلاش میکنیم، جایي که هر فرد باید روي جزئي متفاوت کار کند. در روند مدلسازي یک سیستم داشتن نمودار اجزاء بسیار مهم است.

نمودار استقرار(Deployment Diagram)

نمودار استقرارUML معماري فیزیکي یک سیستمِ مبني بر کامپیوتر را نشان میدهد. این نمودار میتواند کامپیوترها و دستگاهها را نمایش میدهد، اتصالات آنها با یکدیگر و نرم افزاري که روي هر دستگاه قرار میگیرد را نشان میدهد.

توسعه یک سیستم یک فعالیت انساني است. بدون یک سیستم نمادسازي که به آساني قابل درک باشد، روند توسعه پتانسیل زیادي براي خطا دارد.
با ترکیب مجموعه اي از نمودارها، UML یک استاندارد ارائه میدهد که تحلیلگران سیستم را قادر به ساخت یک طرح چند جانبه میکند که براي مشتریها، برنامه نویسان و هر کسي که در روند توسعه درگیر است، قابل درک میباشد. داشتن تمامي این نمودارها ضروري است زیرا هر یک با ذینفع متفاوتي در سیستم روبرو است.
مدل UML آنچه که فرض میشود سیستم انجام مي دهد را بیان میکند اما چگونگي آنرا بیان نمیکند.

پرسش و پاسخ

قبل از خواندن پاسخ هر سوال سعي كنيد خودتان به هر سوال پاسخ دهيد.

۱ UML چه زماني گسترش پيدا كرد و مقصودش چيست؟
۲ نظر خود را درباره سيستم مفهومي بگوييد!
۳ RAD چيست و چند بخش دارد؟
٤ نمودارهاي UML را نام ببريد؟
٥ كلاس چيست؟ شئ چيست؟

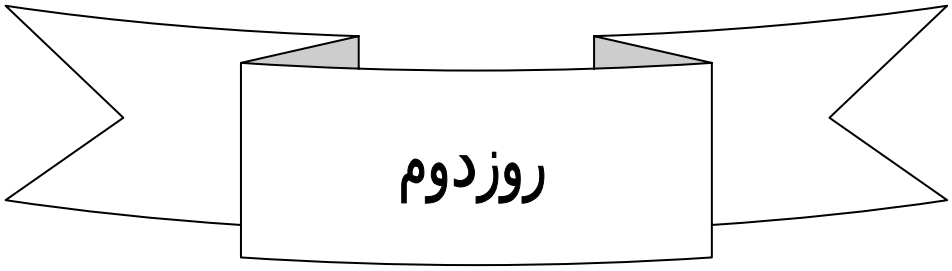
پاسخها:

۱. UML در سال ۱۹۹۷ بعنوان روشي ترسيم نمودار طراحي سيستم گسترش پيدا كرد.
۲. سيستم تركيبي از نرم افزار و سخت افزاركه راه حلي براي مشكلات كسب وكار ارائه ميدهد.
۳. RAD توسعه سريع نرم افزار را نشان ميدهد كه شامل ۵ بخش است: نيازمنديها، گردآوري، تحليل، توسعه و استقرار.
٤. ۹ نمودار UMLوجود دارد: نموداراي كلاس، شئ، فعاليت، مورد كاربرد، حالت، توالي، همكاري، اجزاء و استقرار.
۵. يک كلاس يک طبقه بندي يا گروهي از چيزهايي است كه صفات مشابه و رفتار مشترک دارند. يک شئ يک نمونه از يک كلاس است – يک چيز ويژه كه ارزش ويژه اي از صفات و رفتار را دارد.

کارگاه

خودتان به اين سوالات پاسخ دهيد.

١. چرا روشهاي جديد مدلسازي سيستم از روشهاي قديمي بهتر است؟
٢. فعاليتهاي بخش تحليل GRAPPLE را نام ببريد و محصولات كاري آنها را بيان نماييد.
٣. در كدام بخش آي بخشهايي از RAD ، برنامه نويسان تغيير مي كنند؟
٤. چرا داشتن نمودارهاي متعدد در مدل يک سيستم لازم است؟
٥. كدام نمودارها يک ديد استاتيک از سيستم را ارائه مي دهند؟
٦. كدام نمودارها يک ديد پويا ارائه مي دهند؟(تغييرات در طول زمان را نشان ميدهد)



امروز مطالب زیر را یاد خواهید گرفت:

- ✓ **نمودارهاي كلاس**
- ✓ بصري سازي(visualizing) یک کلاس
- ✓ وابستگي كلاسها(تجمع، وراثت و عمومیت دادن، رابطهاي كاربر و realization)
- ✓ بصري سازي صفات و عملیات
- ✓ **نمودارهاي مورد كاربرد**
- ✓ مدل مورد کاربرد چیست
- ✓ ارتباطات بین مورد کاربردها

نمودارهاي كلاس(Class Diagrams)

در طي اين چند روز شما دانش خود را درباره شئ گرايي(object orientation) به همان اندازه كه درباره UML بيشتر ياد ميگيريد قويتر مينماييد.
در دروس بعد با نمودارهاي كلاس سرو كار خواهيم داشت. چگونه كلاسها، صفات آنها و روشهاي مصاحبه با مشتريان را استخراج كنيم و سپس مقيد كردن كلاسها با ارتباطات نمودارهاي كلاس. شما ميتوانيد مطالب بيشتري در ارتباط با موارد زير بياموزيد:

١. بصري سازي يک كلاس
٢. Association
٣. وراثت و عموميت دادن
٤. تجمع
٥. رابطها و realization
٦. visibility(بصري سازي)

بخش آخر از اين گروه دروس عبارت است از يک مثال، كه تمام مراحل لازم براي ساختن نمودار كلاس را به شما ميدهد.

بصري سازي يک کلاس(visualizing)

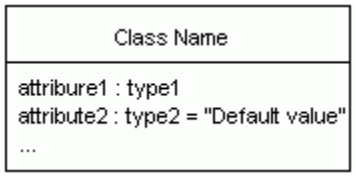
ما در دروس قبلي اشاره کرديم که همه چيز در طبقه بنديها- کلاسها قرار ميگيرند. نمودار کلاس UML شامل کلاسهاي متعددي است که با ارتباطاتي متصل است. اما چگونه UML يک کلاس را نمايش ميدهد؟

مستطيل يک شمايل براي کلاس است. نام کلاس ، طبق قرارداد، يک کلمه با حرف اول بزرگ است . که نزديک به بالاي مستطيل مشاهده مي شود. اگر نام کلاس شما بيش از يک کلمه بود، کلمات را به يکديگر بچسبانيد و حرف اول هر کلمه را بزرگ بنويسيد.

نام کلاس در بالاي مستطيل نوشته ميشود.

صفت(Attribute) يک ويژگي کلاس است. اين مورد دامنه اي از مقاديري را که يک ويژگي بايد در اشياء کلاس نگه دارد را تشريح مي کند. يک کلاس ممکن است صفر يا تعداد بيشتري صفت داشته باشد.

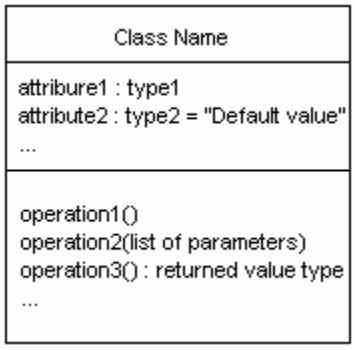
نام صفت يک کلمه اي به حروف کوچک نوشته ميشود. اگر نام بيش از يک کلمه داشته باشد، کلمات به يکديگر متصل ميشوند و هر کلمه به جز کلمه اول با حرف بزرگ شروع ميشود. فهرستي از نام صفات در زير خطي که آنها را از نام کلاس جدا ميکند آورده مي شود.



در شمايل کلاس ، براي هر مقدار صفت ميتوانيد يک نوع مشخص نماييد(رشته اي، اعشاري، عدد، عدد صحيح، منطقي يا نوعي که خود کاربر تعريف نموده باشد) همچنين شما ميتوانيد براي يک صفت يک مقدار پيش فرض نشان دهيد.

يک عمليات چيزي است که يک کلاس ميتواند انجام دهد يا شما (يا کلاسهاي ديگر) ميتوانيد در ارتباط با يک کلاس انجام دهيد.

مانند نام يک صفت، نام يک عمليات با حروف کوچک نوشته ميشود. اگر نام شامل بيش از يک کلمه بود، کلمات به يکديگر مي چسبند و هر کلمه به جز کلمه اول با يک حرف بزرگ شروع ميشود. فهرست عمليات در زير خطي که صفات را از عمليات جدا ميکند شروع ميشوند.



همچنين عمليات ممكن است اطلاعات اضافي داشته باشد. در داخل پرانتزي كه بعد از نام عمليات قرار ميگيرد، شما ميتوانيد پارامتري را كه عمليات روي آن و يا به موازات آن كار ميكند را نشان دهيد. اگر عمليات تابع است ما بايد نوع مقدار برگشتي را مشخص نماييم.

ساير خصيصه هاي اضافي كه ميتوان به صفات كلاس اضافه نمود، قيدها و يادداشتها هستند.
قيدها متنهايي با فرم دلخواه هستند كه در داخل آكولاد { } قرار دارند. متنهايي كه در آكولاد هستند يک يا چند قانون را كه كلاس از آن پيروي ميكند مشخص مينمايند.
يادداشتها همانند عمليات به صفحات پيوست مي شوند. اطلاعات اضافي به كلاس ميدهند. يک يادداشت ميتواند همانند متن شامل يک گرافيک باشد.

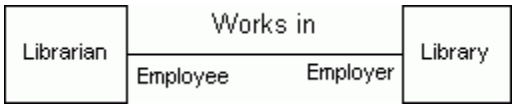
چگونه ميتوانيد كلاسها را از مصاحبه هاي مشتريان استنتاج نماييد؟

در بحث با مشتريان، مراقب اسامي كه آنها براي شرح موجوديتها در كسب و كار خودشان بكار ميبرند باشيد. آن اسامي در مدل شما كلاسها را تشكيل خواهند داد. همچنين مراقب افعالي كه ميشنويد باشيد، زيرا اين افعال عمليات آن كلاسها را تشكيل ميدهند. صفات بعنوان اسامي كه به نام كلاسها مرتبط ميشوند، پديدار خواهند شد.

Associations

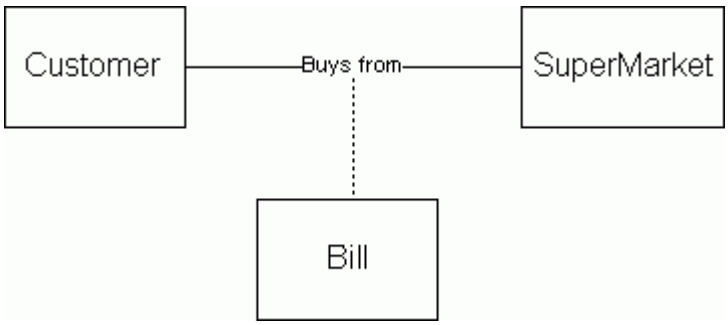
هنگاميكه كلاسها بطور مفهومي به يكديگر مرتبط هستند، اين ارتباط Associations ناميده ميشود.

شما Associations را بعنوان يک خط كه دو كلاس را به يكدگر متصل ميكند، با نام Associations فقط در بالاي خط ميبينيد.



هنگاميكه يک كلاس به كلاس ديگرAssociate ميشود، هر يک معمولاً يک نقش مطابق با آنAssociation ايفا ميكند . شما ميتوانيد اين نقشها را در نمودار با نوشتن آنها نزديک خط در سمت كلاسي كه آن نقش را بازي ميكند، نشان دهيد.

Associations ممكن است پيچيده تر از يک كلاس كه به سايرين مرتبط است باشد. كلاسهاي متعددي ميتوانند به يک كلاس مرتبط شوند. گاهي يک Associations بين دو كلاس بايد از يک قانون پيروي نمايد. شما آن قانون را با قرار دادن يک قيد نزديک خط Associations مشخص ميكنيد.



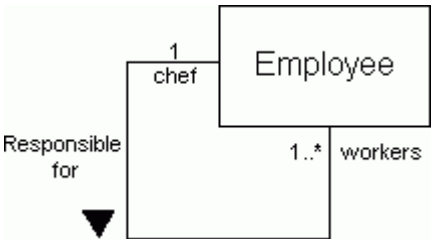
دقيقاً مانند يک كلاس، يکAssociation ميتواند صفات و عمليات داشته باشد. دراين صورت ما يک كلاس Association خواهيم داشت.

شما Associations را به همان طريقي كه يک كلاس عادي را نشان ميدهيد، تجسم ميكنيد و از خط چين براي ارتباط آن به خط Associations استفاده ميكنيد.

چندتايي (Multiplicity) يک نوع ويژه از Associations است که تعداد اشيائي که از يک کلاس به تعداد اشيائي در کلاس وابسته ديگر مرتبط هستند را نشان ميدهد.
يک کلاس ميتواند به صورتهاي زير به کلاسهاي ديگر مرتبط شود:

- يک به يک
- يک به چند
- يک به يک يا بيشتر
- يک به صفر يا يک
- يک به يک فاصله محدود(يک به دو تا بيست)
- يک به دقيقاً n
- يک به مجموعه اي از انتخابها(يک به پنج يا هشت)

UML علامت(*) ستاره را براي نشان دادن بيشتر و بسيار بکار ميرود.
گاهي يک کلاس به خودش وابسته است. اين مورد هنگامي اتفاق مي افتد که يک کلاس اشيائي دارد که ميتوانند نقشهاي متنوعي بازي کنند. اين Associations ها ، Associations هاي بازتابي ناميده ميشوند.

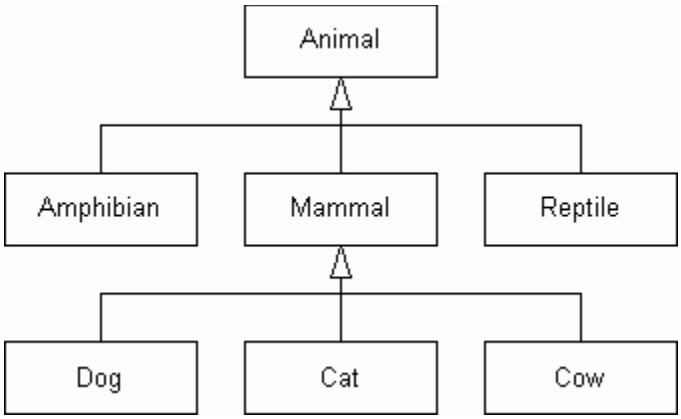


Association
بازتابي

بدون ارتباطات، يک مدل کلاس فهرستي از مستطيلهايي است که مجموعه لغات سيستم را نشان ميدهد. ارتباطات نشان ميدهند که مولفه ها در مجموعه لغات با يکديگر چگونه براي ارائه تصويري از قسمتي از دنيايي که شما در حال مدلسازي آن هستيد مرتبط هستند. Association، يک ارتباط مفهومي بنيادي بين کلاسها است. هر کلاس در يک Association يک نقش ايفا ميکند و چندتايي مشخص ميکند که چند شئ در يک کلاس به يک شئ در کلاس Associated مرتبط است. انواع بسياري از Association ها وجود دارند.

وراثت و عموميت دادن(inheritance & Generalization)

اگر شما چيزهايي درباره يک طبقه از اشياء مي دانيد، بطور خودکار اشيائي را مي شناسيد که مي توانيد به طبقه بنديهايي ديگر انتقال دهيد.
اگر بدانيد که برخي چيزها حيوان هستند، انتظار داريد که بخورد، بخوابد، راهي براي متولد شدن داشته باشد و راهي براي آمدن از يک مکان به ساير مکانها دارد..... اما تصور کنيد که پستانداران، دوزيستان و خزندگان، همه حيوان هستند. همچنين گاوها، سگها، گربه ها ..... در طبقه پستانداران دسته بندي مي شوند. شئ گرايي اين مورد را وراثت نسبت ميدهد.



سلسله مراتب وراثت در قلمرو

يک کلاس (کلاس فرزند يا زير کلاس) ميتواند صفات و عمليات را از ديگري (کلاس والد يا کلاس برتر) ارث ببرد. کلاس والد عمومي تر است بعد از آن کلاس فرزند.
در عموميت دادن، يک فرزند براي والد قابل تعويض است. به بيان ديگر هر کجا که والد ظاهر شود فرزند هم قادر است باشد. با اين وجود عکس آن درست نيست.

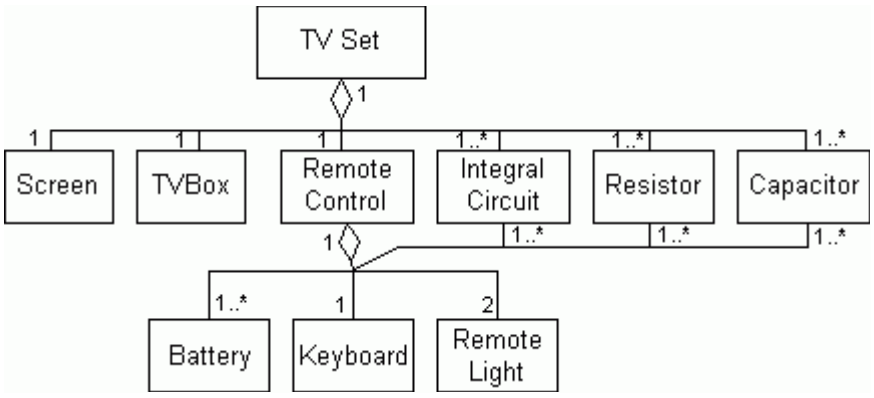
در UML وراثت با يک خط که والد را به کلاس فرزند متصل ميکند، نشان داده ميشود و يک مثلث توخالي در سمت والد قرار ميگيرد.
اگر از جنبه Association نگاه کنيم، وراثت نيز نوعي Association است.

تحليلگران براي پيدا نمودن وراثت چه بايد بکنند؟
تحليلگران بايد دريابند که صفات و عمليات يک کلاس عمومي هستند و شايد براي کلاسهاي متعدد ديگري که صفات و عمليات خود را دارند، نيز بکار روند. امکان ديگر اين است که تحليلگر توجه نمايد که دو يا چند کلاس تعدادي عمليات رايج دارند.

به کلاسهايي که شئ اي(object) ارائه نمي دهند، کلاسهاي مجرد گفته ميشود. يک کلاس مجرد با نوشتن آن با فرم Italic نشان داده ميشود.
يک کلاس ميتواند صفات و عمليات را از ساير کلاسها به ارث ببرد. کلاسهاي ارث بر، فرزند کلاسهاي والدي هستند که از آنها ارث ميبرند. کلاسهاي مجرد تنها به عنوان پايه هايي براي وراثت کانديد هستند و از خود شئ اي ارائه نميدهند.

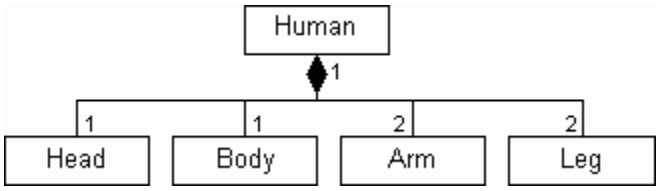
تجمع(Aggregation)

گاهي يک کلاس شامل تعدادي از کلاسهاي ترکيب کننده است. اين مورد نوع ويژه اي از ارتباط است که تجمع نامیده ميشود. اجزاء و کلاسي که آنها تشکيل ميدهند در association جزء- کل هستند.
تجمع بصورت يک سلسله مراتب که کلاس کل"whole" در بالا و اجزا در پايين قرار دارند ، نشان داده ميشود. يک خط کلاس کل را به يک جزء با يک لوزي توخالي نزديک به کلاس کل متصل ميکند.
بيايد نگاهي به بخشهايي که يک دستگاه تلويزيون از آن تشکيل شده، بيندازيم. هر تلويزيون يک جعبه، صفحه نمايش، بلندگو، مقاومتها، خازنها، ترانزيستورها، IC ها... و گاهي يک کنترل از راه دور. کنترل از راه دور ميتواند اين بخشها را داشته باشد: مقاومتها، خازنها، ترانزيستورها، IC ها، باتري، صفحه کليد و چشم الکترونيکي.



گاهي مجموعه اي از اجزاي ممکن در يک تجمع در يک رابطه OR قرار ميگيرند. براي مدلسازي اين مورد، ممکن است از يک قيد استفاده نماييد- کلمه OR در داخل پرانتز روي يک خط چين که دو خط جزء- کل را به هم متصل ميکند، قرار ميگيرد.
يک composite نوع قوي اي از تجمع است. هر جزء در composite ميتواند فقط به يک کل تعلق داشته باشد. نماد composite همانند نماد تجمع است به جز لوزي که توپر است.
اگر شما ظاهر انسان را بررسي نماييد، درميابيد که هر فرد اين اجزا را دارا است: سر، بدن، بازوها، پاها.

این موارد روي شكل نشان داده شده اند.



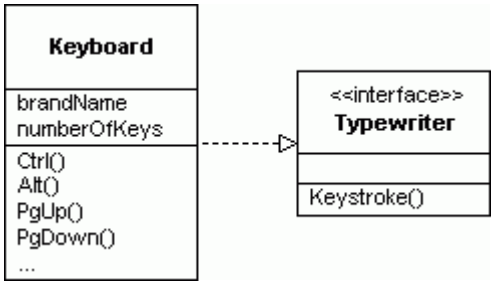
یک composite association. در این association هر جزء کاملاً به یک کل تعلق دارد

یک تجمع، یک association جزء- کل را مشخص میکند. یک کلاس کل"whole" از کلاسهاي اجزء ساخته شده است. یک composite فرم قوي اي از تجمع است و یک جزء در یک composite ميتواند بخشي از تنها یک کل باشد. Composite و تجمع بصورت خطوطي كه كل و اجزا را با یک لوزي توخالي در سمت كل به يكديگر متصل ميشوند، نشان داده ميشود.

رابطها(interfaces) و محقق سازي(Realization)

در دروس قبل آموختيم كه چگونه كلاسها را از مصاحبه با مشتريان استخراج نماييم و آنها را با ارتباطات مختلف بين آنها مرتبط نماييم. اما ممكن است كه برخي كلاسها به والد خاصي مرتبط نباشند، اما رفتار آنها ممكن است شامل برخي عملياتِ همانند با علائم مشابه باشند. شما ميتوانيد عمليات را براي يكي از كلاسها كدگذاري نماييد و از آن براي ساير كلاسها مجدداً استفاده نماييد.

رابط مجموعه اي از عمليات است كه برخي ابعاد رفتار يک كلاس را مشخص مينمايد و مجموعه اي از عملياتي است كه يک كلاس به كلاسهاي ديگر نشان ميدهد.
شما يک رابط را به همان روش مدلسازي يک كلاس، مدلسازي ميكنيد، با شمايل مستطيل، اما رابط هيچ صفتي ندارد و فقط عمليات دارد. روش ديگر، استفاده از يک دايره كوچک است كه با يک خط به كلاس متصل شده است.
صفحه كليد كامپيوتر يک رابط است كه قابليت استفاده مجدد دارد. عمليات استفاده از صفحه كليد از ماشين تحرير مجدداً استفاده شده است. مكان قرار گرفتن كليدها همانند ماشين تحرير است اما نكته اصلي اين است كه عمليات استفاده از صفحه كليد از يک سيستم به سيستم ديگر انتقال داده شده است. همچنين روي صفحه كليد كامپيوتر شما ميتوانيد شماري از عمليات را بيابيد كه در ماشين تحرير نداريد. (Ctrl, Alt, PageUp, PageDown...).



يک رابط مجموعه اي از عمليات است كه يک كلاس انجام ميدهد.

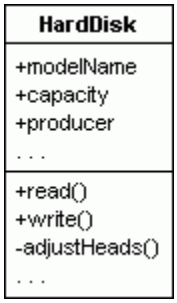
براي تشخيص رابطها از كلاسها، در يک ساختار كليشه اي <<interface>> يا علامت / را در ابتداي نام هر رابط قرار ميدهيم.

ارتباط بين يک كلاس و يک رابط محقق سازي ناميده ميشود. ارتباط بصورت يک خط چين با يک مثلث بزرگ توخالي كه به رابط متصل است و به آن اشاره ميكند، مدلسازي ميشود.

بصري سازي(visibility)

visibility براي صفات و عمليات بكار برده ميشود و وسعت كلاسهايي كه ميتوانند صفات و عمليات يك كلاس داده شده را استفاده نماينده مشخص ميكند.
سه سطح براي visibility وجود دارد (آخرين نمادها كه در UML براي نشان دادن سطوح مختلف بكار ميروند) :

- سطح عمومي : مورد استفاده آن به ساير كلاسها بسط داده ميشود. +
- سطح حفاظت شده : مورد استفاده آن تنها براي كلاسهايي قابل بحث است كه از كلاس اصلي ارث مي برند #
- سطح خصوصي : تنها كلاس اصلي مي تواند از صفت و عمليات استفاده نمايد. -



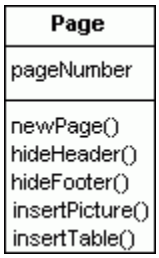
عمليات عمومي و خصوصي در يك ديسك سخت

نمودارهاي كلاس- مثال

در اينجا توضيح مختصري درباره نوشتن يك متن ارائه مي گردد:
فرض كنيد كه شما در حال نوشتن يك مستند در يكي از ابزارهاي معروف پردازش متن مانند نرم افزار word هستيد. شما ميتوانيد كار تايپ يك مستند جديد را شروع نماييد و يا آنچه را از قبل وجود دارد باز كنيد. يك متن را با استفاده از صفحه كليد تايپ ميكنيد.
هر مستند حاوي صفحات متعدد است و هر صفحه شامل سربرگ(Header)، بدنه مستند، و پا برگ(Footer) است. در سربرگ و پا برگ ممكن است روز، زمان، شماره صفحه، مكان فايل و ... را اضافه نماييد.
بدنه مستند داراي جملات است. جملات از كلمات و علائم نقطه گذاري تشكيل شده اند. كلمات حاوي حروف، اعداد و يا كاراكترهاي خاصي هستند. همچنين در يك متن ممكن است تصاوير و جداول را نيز وارد نماييد. جدول از رديفها و ستونها تشكيل شده است. هر خانه اي از جدول مي تواند متون و تصاوير را در خود جاي دهد.
پس از اتمام كار يك مستند، كاربر ميتواند انتخاب نمايد كه مستند را ذخيره نمايد يا آنرا چاپ كند.
اين مورد توضيح ساده اي از ايجاد يك مستند بود. اگر فهرست اسامي كه متن قبلي را شكل داده اند را خلاصه نماييم به فهرست زير ميرسيم:
مستند، ابزار پردازشي متن، نرم افزارword، **متن،** صفحه كليد، **سربرگ، پا برگ،** بدنه مستند، **تاريخ، زمان، شماره صفحه، مكان فايل، صفحه ، جمله، كلمه، علائم نقطه گذاري، حرف، عدد، كاراكتر خاص، تصوير، جدول، رديف، ستون، خانه، كاربر.**
اسامي كه پر رنگ نشان داده شده اند كلاسها و صفات منتخب براي مدل ما هستند.
بياييد با يك مستند شروع كنيم. همانطور كه ميتوانيد ببينيد اين مثال درباره مستندات بحث ميكند، بنابراين كلاس مركزي ما در اين نمودار كلاس يك مستند خواهد بود. يك مستند صفحات متعددي دارد بنابراين تعداد صفحات يكي از صفات كلاس مستند خواهد بود . براي عمليات آن داريم:
Open(), save(), print() و new() . هر مستند حاوي تعداي صفحه ميباشد. صفحات نيز ميتوانند براي كلاس در نظر گرفته شوند

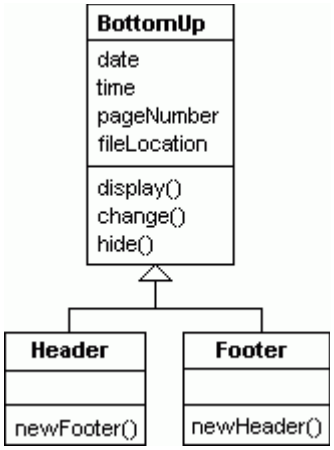
كلاس مستند

کلاس صفحه میتواند شماره صفحه را بعنوان یك صفت داشته باشد و عملیات مجاز در اینجا میتواند اینها باشد: new page(), hide header() و hide footer() . عملیات مربوط به سربرگ و پا برگ بیان میکند که این دو نیز میتوانند یك کلاس باشند.



کلاس صفحه

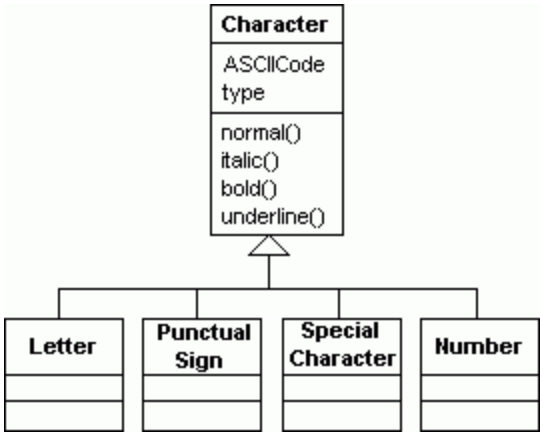
کلاس سربرگ و کلاس پا برگ صفات رایجي دارند: تاریخ، زمان، شماره صفحه و مکان فایل. این صفات براي هر سربرگ و پا برگ اختیاري هستند و ممکن است کاربر آنها را شکل دهد. این موضوع ما را به سوي مطرح کردن یک کلاس متعارف هدایت میکند.
اکنون زمان مناسبي جهت ایجاد یك وراثت است. کلاس والد bottom up خواهد بود (این نام بدین دلیل انتخاب شده است که سربرگ و پا برگ در بالا و پایین هرصفحه دیده میشوند) و صفات رایج سربرگ و پا برگ و همچنین عملیات روبرو را دارا است: display(),hide() و change() . کلاسهاي سربرگ و پا برگ (فرزندان این کلاس)فقط این عملیات را دارا هستند:new Headre() و newFooter().



کلاس bottomup و فرزندان آن

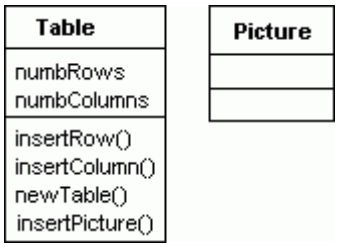
قبل از آنکه بدنه مستند یا متن را امتحان کنیم بیایید نگاهي به فعالیتهاي ایجاد یك متن بیندازیم. متن مستند از جملات ساخته شده است.جملات از کلمات و کلمات از کارکترها ساخته شده اند. اگر کلمات از کاراکترها و جملات از کلمات تشکیل شده اند، در این صورت جملات از کاراکترها تشکیل

شده اند. بنابراين بدنه يك مستند ميتواند از كاراكترها تشكيل شده باشد. بدين جهت براي ايجاد متن مستند ما كلاس كاراكتر با فرزندان آن را بكار خواهيم برد. كلاس كاراكتر ASCIIcode و type را بعنوان صفت خواهد داشت.(type نوع كاراكتر - نرمال، Italic, bold يا under line بودن آنرا بيان ميكند.)و normal(), bold(), italic() و underline() عمليات آن هستند. فرزندان كلاس كاراكتر اينها خواهند بود: حرف، علائم نقطه گذاري وكاركترهاي خاص و نرمال. هم همچنين در بدنه مستند جداول و تصاوير ميتوانند وجود داشته باشند. اينها كلاسهاي جديد نمودار كلاس ما هستند.



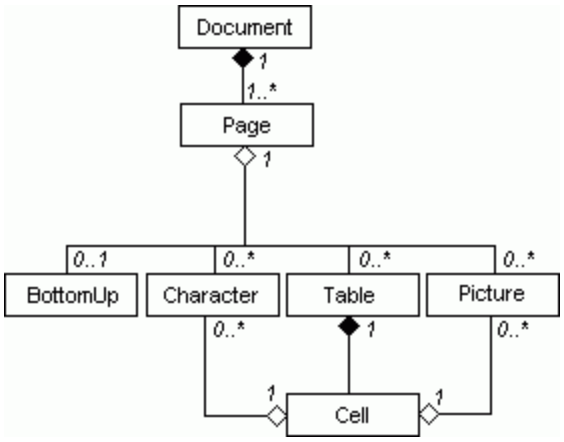
سلسله مراتب كاراكترها

كلاس جدول داراي صفات newRows و newColumns و عمليات newRoww(), newColumn() و new table() ميباشد. هر جدول شامل يك يا چند خانه است.و در هر خانه متن و يا تصوير ميتوان قرار داد.



كلاس جدول و كلاس تصوير

با داشتن تمامي اين كلاسها در پيش رو ميتوانيد association هاي بين آنها را درآورده و مجدداً و براي اين مثال يك نموداركلاس كامل شده خواهيم داشت .



نمودار کلاس جهت ایجاد مستند متني

نمودار کلاس داده شده به صورت جزئیـات تکمیل نشده است. اگر موارد بیشتري در ارتباط با تایپ یک مستند میدانید، اطلاعات خود را به نمودار اضافه نمایید. این مدل میتواند بزرگ و بزرگتر شود و تمام مراحل لازم که براي تحلیل و طراحي با آن مواجه هستید درجایي که ابزار پردازش متني مدلسازي شده است را ایجاد نماید.

نمودارهاي مورد كاربرد(Use case diagram)

نمودارهاي كلاس يک ديد استاتيک از كلاسها در يک سيستم فراهم ميكنند. نمودارهاي بعدي يک ديد پويا فراهم مينمايند و نشان ميدهند كه سيستم و كلاسهاي آن در آينده چگونه تغيير ميكنند.

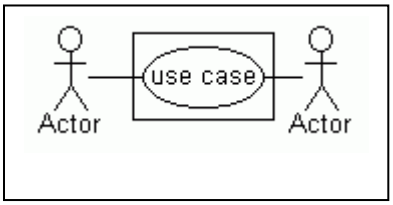
ديد استاتيک به يک تحليلگر كمک ميكند كه با مشتري ارتباط برقرار كند. ديد ديناميک كمک ميكند كه يک تحليلگر با يک تيم از توسعه دهندگان ارتباط برقرار نمايد و به توسعه دهندگان كمک ميكند كه برنامه ها را ايجاد نمايند.

از آنجائيكه نمودار كلاس يک راه مناسب جهت تحريک مشتري براي صحبت كردن درباره يک سيستم از نقطه نظر خود او ميباشد، مورد كاربرد يک ابزار بسيار عالي براي تحريک كاربران بالقوه جهت صحبت درباره سيستم از ديد آنهاست. هميشه براي كاربران آسان نيست كه بطور مفصل درباره اينكه چگونه روش استفاده از يک سيستم را دريافتند صحبت كنند اين يک حقيقت است كه كاربران بيش از آنچه صحبت ميكنند، ميدانند: مدل كاربرد كمک ميكند تا اين حصار شكسته شود.
مصاحبه با كاربران در مجموعه اصطلاحات قلمرو آغاز مي گردد؛ اما بايستي بعد از آن به سمت مجموعه اصطلاحات كاربر منتقل گردد. نتايج اوليه مصاحبه ها بايد actor ها و موردهاي كاربرد سطح بالا كه نيازهاي عملكردي را بطور كلي شرح ميدهند را آشكار نمايد. اين اطلاعات مرزها و قلمرو سيستم را مشخص ميكند.
مصاحبه هاي بعدي با كاربران در اين نيازمنديها به علت مدلهاي مورد كاربرد كه سناريوها و تواليها(sequence) را با جزئيات نشان ميدهند به دقت كاوش ميكند. اين امرممكن است سبب ايجاد مورد هاي كاربرد بيشتري گردد كه روابط Inclusion وextension را تامين نمايد. در اين فاز درک شما از قلمرو بسيار مهم است زيرا اگر آنرا به درستي درک نكنيد ممكن است موردهاي كاربرد بسياري ايجاد نماييد كه ميتواند مانع از روند تحليل شود.
مورد كاربرد مجموعه اي از سناريوها درباره كاربرد سيستم است. هر سناريو يک توالي از رخدادها را شرح ميدهد. هر توالي بوسيله يک شخص ، يک سيستم ديگر، يک قطعه سخت افزار و يا با گذر زمان آغاز شده است. موجوديتهايي كه يک توالي را آغاز ميكنند actor ناميده ميشوند. نتيجه توالي بايد موارد استفاده اي چه براي actorي كه آنرا آغاز كرده و يا actor ديگر باشد.
ميتوان از مورد كاربرد دوباره استفاده نمود. يک روش-Inclusion، عبارت است از بكارگيري مراحل يک مورد كاربرد بعنوان بخشي از مراحل يک توالي در يک مورد كاربرد ديگر. راه ديگر- extension، كه عبارت از ايجاد يک مورد كاربرد جديد با اضافه كردن مراحلي به مورد كاربرد موجود است.

درسهاي بعد با نشان دادن يک مدل مورد کاربرد و تجسم روابط بين موردهاي کاربرد شما را راهنمايي ميکند. در آخر يک مثال مورد کاربرد به شما داده خواهد شد.

معرفي يک مدل مورد کاربرد

يک actor يک مورد کاربرد را آغاز ميکند، و يک actor (که ممکن همان آغازگر باشد اما الزامي نيست) برخي مقادير را از مورد کاربرد دريافت ميکند.



مدل مورد کاربرد

بيضي يک مورد کاربرد را نشان ميدهد و شکل آدمک يک actor را نشان ميدهد actor آغازگر در سمت چپ مورد کاربرد و actor دريافت کننده در سمت راست قرار دارند. نام actor فقط در زير آن ديده ميشود. نام مورد کاربرد در درون بيضي ويا فقط زير آن مشاهده ميشود. خطassociation يک actor را به مورد کاربرد مرتبط ميکند و ارتباط بين actor و مورد کاربرد را نشان ميدهد. خط association همانند خطي که کلاسهاي وابسته را به هم مرتبط مي ساخت، توپر است.

هر مورد کاربرد فهرستي از سناريوها است و هر سناريو توالي از مراحل است. بعلاوه هر سناريو از هر مورد کاربرد صفحه مربوط به خود را دارد که در متن مطابق زير ليست ميشود:

- Actor کسي که يک مورد کاربرد را آغاز ميکند.
- پيش نياز براي مورد کاربرد
- مراحل در سناريو
- شرايط پسين هنگاميکه سناريو کامل است.
- Actor کسي که از مورد کاربرد منفعت ميبرد.

نمودارهاي مورد کاربرد توان بيشتري جهت جمع آوري نيازمنديها ميدهند. آنها موردهاي کاربرد را تجسم ميکنند، ارتباط بين تحليلگران و کاربران و ارتباط بين تحليلگران و مشتري را آسان مينمايند. در يک نمودار مورد کاربرد، بيضي نمادي است که براي موردکاربرد بکار ميرود، براي actor از نماد آدمک استفاده ميشود و خط association يک actor و مورد کاربرد را به يکديگر متصل ميکند. موردهاي کاربرد معمولاً در داخل يک مستطيل که مررز سيستم را نشان ميدهد هستند.

ارتباطات بين موردهاي كاربرد

در درس نمودار مورد كاربرد، ذكر كرديم كه ممكن است از نمودارهاي مورد كاربرد دوبار استفاده شود. يک راه Inclusion و راه ديگر extension بود. اينها ارتباطات بين موردهاي كاربرد هستند . دو نوع ديگر از اين ارتباطات نيز وجود دارد: عموميت دادن و گروهبندي.

Inclusion شما را قادر ميسازد مراحل مورد كاربرد را در مورد كاربرد ديگر دوباره استفاده نماييد.

براي نشان دادن Inclusion، نمادي را كه براي وابستگي(dependency) بين كلاسها استفاده ميكرديد بكار ميبريد- خط چين كه كلاسها را با پيكاني كه به كلاس وابسته اشاره ميكند مرتبط ميكنند. درست بالاي خط، يک كليشه <<include>> را درون گيومه اضافه نماييد.

extension:

extension به شما اجازه ميدهد يک مورد كاربرد جديد به وسيله اضافه كردن مراحل به مورد كاربرد موجود ايجاد نماييد.ensi

همچنين در اينجا يک پيكان نقطه چين به اضافه يک كليشه <<extends>> كه در گيومه قرار داده شده براي نشان دادن extension بكار ميرود. مطابق با مورد كاربرد پايه، نشان exton در زير نام مورد كاربرد مشاهده ميشود.

عموميت دادن(generalization):

كلاسها و همچنين موردهاي كاربرد ميتوانند از سايرين ارث ببرند. در وراثت مورد كاربرد، مورد كاربرد فرزند، رفتار و مفهوم را از والد به ارث ميبرد و به رفتار خود اضافه ميكند. شما ميتوانيد فرزند را در هر كجا كه والد را بكار ميبريد، استفاده نماييد.

عموميت دادن به همان روشي كه كلاسهاي عموميت بخشيدن را مدلسازي كرديد، مدلسازي ميشود- با يک خط توپر كه يک مثلث توخالي به والد اشاره دارد.

ارتباط عموميت دادن ميتواند همانند موردهاي كاربرد بين actor ها وجود داشته باشد.

گروهبندي(grouppping) :

در برخي نمودارها ي مدل كاربرد، ممكن است تعدادي موردهاي كاربرد داشته باشيد و بخواهيد آنها را سازماندهي كنيد. اين زماني ميتواند اتفاق بيفتد كه يک سيستم شامل شماري از زير سيستمها باشد.

سر راست ترين راه تشكيل گروه مرتبط به مورد كاربرد در داخل يک بسته(package) است(يک پوشه برچسب دار).

در درس آینده، مثال نمودارهاي مورد كاربرد ، به تمام این ارتباطات به دقت نگاه میكنیم تا اینكه به شما كمک كنیم موردهاي كاربرد را بعنوان بخشي از نمودارهاي مورد كاربرد درک كنید.

نمودارهاي مدل كاربرد- مثال

اين يک مثال است که ما در آن ميخواهيم نمودارهاي مورد کاربرد را براي يک دستگاه سلف سرويس، مدلسازي کنيم.
کارکردهاي مهم دستگاههاي سلف سرويس اين است که به مشتري اجازه ميدهد محصولي را از دستگاه خريداري نمايند(آبنبات، شکلات، آب ميوه و ....). هر کاربري که شما از او درباره مجموعه سناريوهايي که در طي استفاده از دستگاه رخ ميدهد، سوال کنيد، ميتواند به شما بگويد که مورد کاربرد مهم اين کار "خريد يک محصول" است. بياييد هر سناريوي ممکني را در اين مورد کاربرد آزمايش کنيم. اين سناريوها از طريق بحث با کاربران آشکار ميشود.

مورد کاربرد"خريد يک محصول"

Actor در اين مورد مشتري است. مشتري ميخواهد تعدادي از محصولاتي که توسط دستگاه سلف سرويس به او پيشنهاد ميشود را خريداري نمايد. در ابتدا او پول را به داخل دستگاه مي اندازد، يک يا چند محصول را انتخاب ميکند و ماشين محصول(ها) انتخاب شده را به او ارائه ميدهد. نمودار مدل کاربرد براي اين سناريو را ميتوان به صورت زير نشان داد:

نمودار مدل کاربرد خريد

اما اگر کمي فکر کنيم، ساير سناريوها نيز ناگهان به ذهن ميرسند. ممکن است که دستگاه سلف سرويس يک يا چند محصول را نداشته باشد و يا دستگاه مقدار درستي پول براي برگرداندن به مشتري نداشته باشد.

- آيا ميتوان فرض کرد که با اين سناريوها سرو کار داريم؟
اگر جواب مثبت است، بياييد برگرديم به وقتي که مشتري پول را داخل دستگاه مي اندازد و انتخاب خود را وارد دستگاه ميکند. بعد از اين تصور کنيد که دستگاه اين جنس را ندارد. در اين صورت ترجيحاً بايد پيامي به مشتري نشان داده شود که دستگاه جنس ندارد و به او اجازه دهد که انتخاب ديگري انجام دهد يا پول را پس بگيرد. اگر سناريو مقدار پول نادرست رخ دهد، دستگاه سلف سرويس اصل پول را به مشتري برميگرداند.

در اينجا پيش شرط يک مشتري گرسنه يا تشنه و شرط پسين گرفتن محصول از دستگاه و يا برگرداندن پول است.
اين يک سناريو مورد کاربرد از ديدگاه يک کاربر(مشتري) است. اما کاربران بسيار زياد ديگري وجود دارند. يک کارپرداز بايد دستگاه را دوباره پر کند و يک متصدي جمع آوري بايد پولهاي انباشته شده را جمع آوري نمايد. اين به ما ميگويد که ما حداقل بيش از دو مورد کاربرد بايد ايجاد نماييم:
"پر کردن دوباره " و "جمع آوري پول". بياييد به دقت با مصاحبه با کارپرداز و متصدي جمع آوري نگاهي به هر دو آنها بيندازيم.

مورد کاربرد"پر کردن دوباره"

فعاليتي که کارپرداز بايستي در هر فاصله زماني(يک يا دوهفته) انجام دهد عبارت است از اينکه: کارپرداز قفل امنيتي دستگاه را باز ميکند، در جلوي دستگاه را ميگشايد و هر جنسي را در محل مخصوص به آن به اندازه ظرفيت پر ميکند.(ممکن است که کارپرداز هر جنس را مطابق با ميزاني که مصرف شده پر نمايد.)سپس در جلوي دستگاه را مي بندد و آن را لحاظ امنيتي ايمن ميسازد.
پيش شرط، فاصله زماني و شرط پسين اين است که کارپرداز مجموعه جديدي از ذخاير براي فروش دارد.
اين نمودار مورد کاربرد در شکل زير نشان داده شده است:

مورد کاربرد"پر کردن دوباره "

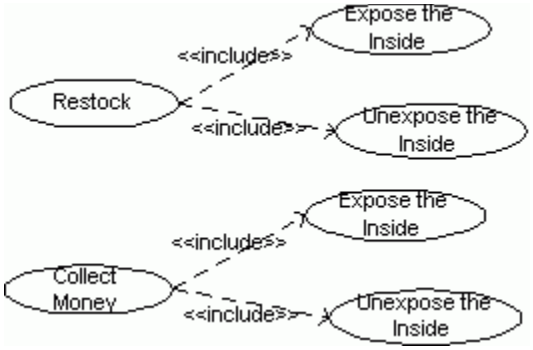
مورد کاربرد "جمع آوري پول"

مسوول اين مورد کاربرد متصدي جمع آوري است. او ميتواند همان شخصي باشد که کارپرداز هم هست. مراحلي که متصدي جمع آوري بايد انجام دهد همانند کارپرداز است، اما متصدي جمع آوري به جاي محصولات با پول سرو کار دارد. وقتي که فاصله زماني مي گذرد اين شخص مقدرا پول لازم را از دستگاه جمع آوري ميکند. شرط پسين در اينجا پولي است که در دست متصدي جمع آوري است.
نمودار مورد کاربرد جمع آوري پول به صورت زير است:

نمودار مورد کاربرد جمع آوري

مراحل باز كردن قفل امنيتي دستگاه، باز كردن در جلو، بستن و ايمن سازي دستگاه در مورد كاري كه كارپرداز و متصدي جمع آوري بايد انجام دهند، يكسان است. اين جاي خوبي براي دربرداشتن يک مورد كاربرد است. بياييد مراحل "باز كردن قفل ايمني" و "باز كردن در" را در يک مورد كاربر به نام "نشان دادن قسمت داخلي " و "بستن در دستگاه" و "ايمن سازي " را در مورد كاربردي به نام "مخفي كردن قسمت داخلي" تركيب نماييم.

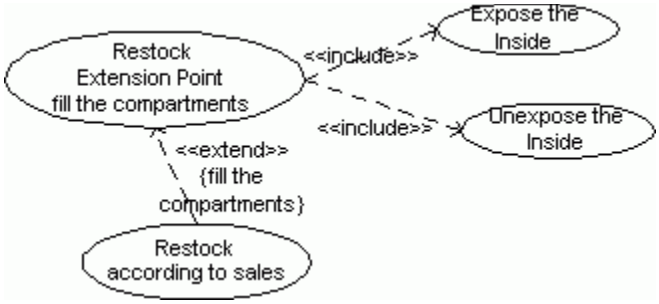
با احتساب يک مورد كاربرد، موردهاي كاربرد پر كردن دوباره و جمع آوري همانند شكل زير ديده ميشوند:



فرايند Inclusion در موردهاي كاربرد پر كردن دوباره و جمع آوري پول

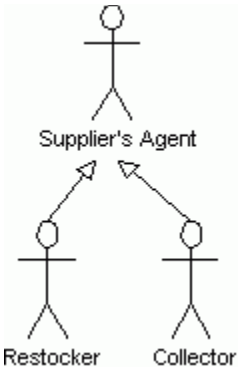
مورد كاربرد پر كردن دوباره ميتواند پايه موردهاي كاربرد ديگر باشد: "پر كردن دوباره مطابق با فروش". در اينجا كارپرداز ممكن است اجناس را با محصولات جديد مطابق با فروش آنها پر نمايد. اين extension يک مورد كاربرد است.

بعد از Inclusion و extension مورد كاربرد پر كردن دوباره ميتواند به شكل زير باشد:



فرايند بسط در مورد كاربرد پر كردن دوباره

همچنين عموميت دادن ممكن است براي actor هاي كارپرداز و متصدي جمع آوري رخ دهد. اگر هر دوي آنها يک شخص باشند، بياييد بگوييم مامور پر كردن دوباره ، سپس كارپرداز و متصدي جمع آوري هر دو فرزند مامور پر كردن دوباره هستند. در شكل بعد اين موضوع نشان داده شده است:



عموميت دادن بين كارپرداز و متصدي جمع آوري پول

ما به پايان تحليل خود درباره دستگاه سلف سرويس رسيديم. نمودار مورد كاربرد تكميل شده نيازمنديهاي كاركردي سيستم را نشان مي دهد و ما قادر خواهيم بود حركت بعدي طراحي و توسعه را انجام دهيم.

پرسش و پاسخ

اگر شما تمام دروس امروز را یاد گرفته باشید، اکنون زمان مناسبي براي امتحان و تقویت دانش شما درباره کلاس و نمودارهاي مورد کاربرد بعنوان بخشي از UML است.

۱. در UML چگونه یک کلاس را نمایش میدهید؟
۲. کلاس مجرد چیست؟
۳. موجودیتي که یک مورد کاربرد را آغاز میکند را چه مي نامید؟
٤. شباهتها و تفاوتهاي بین کلاسها و موردهاي کاربرد چیست؟
٥. دو مزیت بصري سازي یک مورد کاربرد را نام ببرید.

پاسخها:

۱. مستطيل نماد یک کلاس است. نام کلاس را در بالاي آن قرار میدهید و صفات را در زیر آن لیست مي کنید و در زیر آن لیست عملیات را قرار میدهید.

۲. یک کلاس مجرد بعنوان یک پایه براي وراثت بکار میرود اما هیچ شئ اي ارائه نمیدهد.

۳. موجودیتي که یک مورد کاربرد را آغاز میکند actor نامیده میشود 

٤. شباهتها: هر دو عناصر ساختاري هستند. هر دو میتوانند ارث ببرند.
تفاوتها: کلاس شامل صفات و عملیات است. مورد کاربرد شامل سناریوها و هر سناریو شامل توالي مراحل است. نمودار کلاس یک دید استاتیک از بخشهاي سیستم ارائه میدهد، مورد کاربرد یک دید دینامیک از رفتار را نشان میدهد. کلاس دورن یک سیستم را نشان میدهد. مورد کاربرد نشان میدهد که سیستم چگونه به بیرون نگاه میکند.

۵. با بصري سازي شما میتوانید:
۱- موردهاي کاربرد را به کاربران نشان دهید و اطلاعات اضافي را از آنها دریافت نمایید.
۲- آنها را با سایر نمودارهاي دیگر ترکیب نمایید.

روز سوم

امروز مطالب زير را ياد خواهيد گرفت:

- ✓ نمودارهاي حالت
- ✓ **انتقال بين حالتها**
- ✓ نمودارهاي توالي
- ✓ **روشهاي ايجاد توالي ها**
- ✓ نمودارهاي همكاري
- ✓ **نوشتن نمودارهاي همكاري**

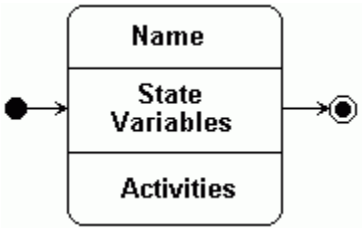
نمودارهاي حالت

در دروس بعدي شما با عناصري كار خواهيد نمود كه قبلاً كار نكرده ايد. اين عناصر، عناصر رفتاري هستند كه نشان ميدهند كه چگونه بخشهاي يک مدل UML در طي زمان تغيير ميكنند. همانطور كه سيستم با كاربران و احتمالاً با ساير سيستمها در تعامل است، اشيائي كه سيستم را ميسازند از تغييرات لازمي گذر ميكنند كه تعاملات را همساز ميكنند. اگر شما ميخواهيد سيستمي را مدلاسازي نماييد بايد مكانيسيمي براي تغيير مدل داشته باشيد. اين مكانيسم در UML، نمودارهاي حالت هستند.

- هنگامي كه شما كليدي را ميزنيد، لامپ حالت خود را از خاموش به روشن تغيير ميدهد.
- هنگامي كه screen saver بر روي كامپيوتر شما در حال نمايش است و شما به كليدهاي صفحه كليد ضربه ميزنيد يا موس را حركت ميدهيد، كامپيوتر شما از حالت screen saver خارج مي شود و به حالت كار _حالت فعال برمي گردد.

نمودار حالت UML، حالاتي را كه يک شي ميتواند همراه انتقال بين حالات داشته باشد و نقطه شروع و نقطه پايان توالي تغييرات حالت را نشان دهد.

حالتها در نمودار حالت با مستطيلهايي كه گوشه هاي آن گرد است نمايش داده ميشوند و نماد انتقال يک فلش توپر است. يک دايره توپر در نقطه شروع توالي قرار دارد و يک نماد هدف مانند در نقطه پايان نمايش داده ميشود.



نمادهاي UML در نمودارهاي حالت. ناحيه بالا براي نام حالت، ناحيه مياني براي متغيرهاي حالت و ناحيه پايينيِ براي فعاليتها ميباشد. انتقالها با فلش توپر نمايش داده ميشوند و دايره ها براي نقاط شروع و پايان هستند.

همچنين شما مي توانيد جزئيات را به شمايل حالت با تقسيم آن به سه ناحيه اضافه نماييد. ناحيه بالا نام حالت را در بردارد(اين نام را شما بايد هنگامي بكار ببريد كه بدانيد كلاس تقسيم شده است يا نه) ناحيه مياني متغيرهاي حالت را دربردارد (زمان سنجها، شمارشگرها، تاريخها و ....)و ناحيه پايييني فعاليتها را دربر دارد(ورود- هنگامي كه سيستم وارد يک حالت ميشود چه اتفاقي مي افتد، خروج- هنگامي كه سيستم از يک حالت خارج ميشود چه اتفاقي مي افتد، انجام دادن- هنگامي كه سيستم در يک حالت است چه اتفاقي مي افتد.)

جزئيات حالت و انتقال

شما ميتوانيد رويدادي كه باعث رخ دادن يک انتقال ميشود (يک رويداد آني) و محاسباتي (فعاليت) كه وقوع تغيير حالت را ايجاد و اجرا مينمايد را نشان دهيد.

براي افزودن رويدادها و فعاليتها آنها را نزديک خط انتقال مينويسيد، از يک نشان مميز(/) براي جدا كردن يک رويداد آغازگر از يک فعاليت استفاده كنيد.

گاهي يک رويداد سبب يک انتقال بدون فعاليت مرتبط ميگردد و گاهي يک انتقال به دليل اينكه يک رويداد يک فعاليت را كامل نموده رخ مي دهد.(به جاي اينكه به علت يک حادثه باشد) اين نوع انتقال، انتقال متداوم(**triggerless transition**) ناميده ميشود.

همچنين ممكن است شرط محافظ باشد. هنگاميكه اين مورد مواجه شد، انتقال رخ ميدهد. (براي مثال به حالت screen saver رفتن هنگاميكه فاصله زماني n دقيقه از بيكاري موس گذشته باشد، فعال شده است)

برخي حالتها ممكن است بسيار پيچيده تر از آن باشند كه با يک مستطيل گرد نشان داده شوند. ممكن است درون يک حالت حالتهايي داشته باشيم. اين حالتها ، زير حالت ناميده ميشوند(ترتيبي و همزمان)

زيرحالتهاي ترتيبي در حالتهاي متوالي كه يكي پس از ديگري رخ ميدهند مي آيند.

زير حالتهاي همزمان بايد حاوي دو يا چند زير حالت ترتيبي كه در يک زمان رخ ميدهند باشند. اين مورد با يک خط چين بين حالتهاي همزمان نشان داده مي شود.

اگر هر مشكلي در ارتباط با درک اين مفاهيم داريد، لطفاً به درس مثال نمودار حالت مراجعه نماييد.

UML نمادي را فراهم ميكند كه نشان ميدهد كه يک حالت مركب، زيرحالت فعال خود را هنگاميكه شي به خارج از حالت مركب انتقال پيدا ميكند با به ياد مي آورد. اين نماد Ⓗ با يک خط توپر به زيرحالت يادآوري شده مرتبط ميشود، بايک فلش كه انتهاي آن يک دايره توپر است و به زير حالت اشاره ميكند.

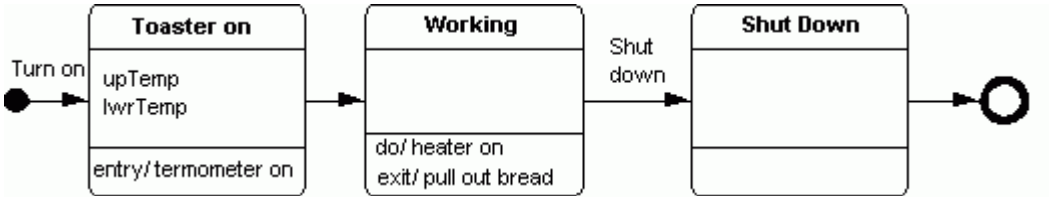
يک پيام كه يک انتقال را در نمودار حالت شيء گيرنده آغاز ميكند، يک علامت ناميده ميشود. در جهان شئ گرا، ارسال يک علامت همانند ايجاد يک شئ راهنما و انتقال آن به شئ گيرنده است. اين شئ راهنما ويژگيهايي دارد كه بعنوان صفات نشان داده ميشوند. چون يک علامت يک شئ است، امكان ايجاد سلسله مراتب وراثت دسته علائم وجود دارد.

نمودار حالت- مثال

فرض کنید در حال طراحي یک دستگاه درست کردن نان برشته(toaster) هستید. ممکن است که نمودارهاي UML فراواني درست کنید، اما آنجا تنها نمودارهاي حالت مورد علاقه ماست.

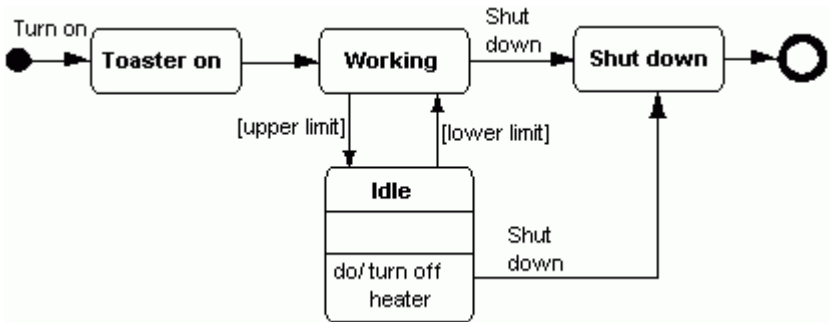
- مراحل درست کردن نان برشته چیست؟

اول همه باید toaster را روشن کنیم، نان را داخل آن گذاشته و چند دقیقه صبر کنیم تا طبخ شود. اولین نمودار حالت در زیر نشان داده شده است:



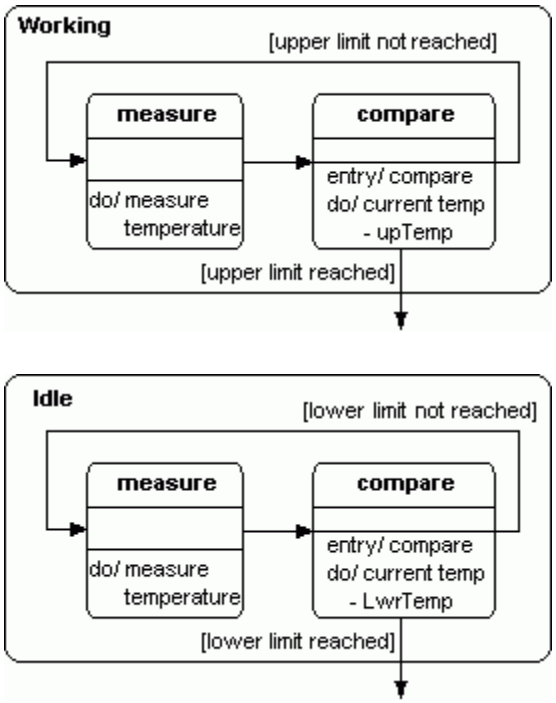
نمودار حالت آغازي براي درست کردن نان برشته

اما این نمودار حالت نهایي نیست. براي جلوگیري از سوختن نان گرمکن دستگاه باید گرما را در رنج خاصي از دما تولید نماید(حد بالا و پایین آن محدود شده است). بدین منظور گرماسنج دماي گرمکن را اندازه میگیرد و هنگامي که به حد بالاي دما رسید گرمکن به حالت بیکاري میرود . این حالت تا هنگامیکه دماي گرمکن به حد پایین کاهش بیابد باقي خواهد ماند و بعد از آن دوباره به حالت کاري میرسد. با این حالت جدید، نمودار حالت توسعه یافته به صورت زیر خواهد بود.



نمودار حالت توسعه یافته براي درست کردن نان برشته

انتقال بين حالت بيكاري و حالت كاري با جزئيات نشان داده نميشود. براي انجام اين كار بايد زير حالتها اضافه شوند.



زيرحالتها درحالتهاي بيكاري و حالت كاري

زير حالتها در حالت كاري و حالت بيكاري خيلي به هم شبيه هستند. هر دو اندازه گيري دارند و حالتها را مقايسه ميكنند، اما در روند مقايسه دما تفاوت قائل مي شوند. حالت كاري بايد دماي جاري را با حد بالاي دما مقايسه نمايد(اگر به اين دما رسيد حالت كاري به حالت بيكاري ميرود) و حالت بيكاري دما را با حد پايين مقايسه ميكند(حالت بيكاري هنگامي كه به دماي حد پايين ميرسد با حالت كاري جايگزين ميگردد).

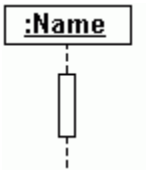
داشتن نمودار حالت بسيار مهم است زيرا به تحليلگران، طراحان و توسعه دهندگان در فهم رفتار اشياء در سيستم كمک ميكنند. توسعه دهندگان، بطور اخص، بايد بدانند كه رفتار اشياء چگونه فرض شده است زيرا آنها بايد اين رفتارها را در نرم افزار اجرا نمايند. اجرا نمودن يک شيء كافي نيست: توسعه دهندگان بايد آن شيء را جهت انجام برخي كارها بسازند.

نمودارهاي توالي

در درس گذشته نمودارهاي حالت را آموختيم كه بر روي حالتهاي يک شئ تمركز ميكند.
اما لازم است بين اشياء ارتباط داشته باشيم. نمودار توالي UML به مرحله بعد ميرود و نشان ميدهدكه اشياء با يكديگر در طول زمان چگونه ارتباط دارند. در اين حوزه extension يافته، شما يک بعد مهم را به حساب ميآوريد: زمان. ايده كليدي در اينجا اين است كه تعاملات بين اشياء در يک توالي مشخص رخ ميدهد و توالي زماني را از شروع تا انتها صرف ميكند.

نمودار توالي شامل اشيائي است كه به روش معمول- در يک مستطيل نامگذاري شده – نشان داده ميشود(زير نام نوشته شده خط كشيده شده است). **پيامها** به صورت فلش توپر و **زمان** بصورت يک پيشرفت عمودي نشان داده مي شود.
بياييد نگاه دقيقتري به بخشهاي نمودارهاي توالي بيندازيم.

اشياء :
اشياء نزديک بالاي نمودار از چپ به راست قرار گرفته اند. به هر ترتيبي كه نمودار را ساده تر ميكند، مرتب ميشوند. extension پايين به بالا از هر شئ يک خط چين است كه خط حيات شئ ناميده ميشود. همراه خط حيات يک مستطيل باريک وجود دارد كه فعاليت ناميده ميشود. فعاليت، اجراي يک عمليات را كه شئ انجام ميدهد را نشان ميدهد. طول اين مستطيل طول مدت فعاليت را نشان ميدهد.



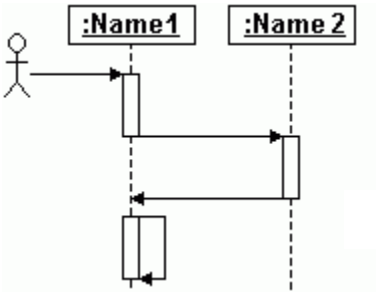
نمايش يک شئ در يک نمودار توالي

پيامها:
يک پيام كه از يک شئ به شئ ديگر ميرود از خط حيات يک شئ به خط حيات شئ ديگر ميرود. يک شئ ميتواند يک پيام به خودش پيام ارسال نمايد به عبارت ديگر از خط حيات خود به خودش برگشت دهد. اين مورد بازگشت ناميده ميشود. مثالهاي بازگشت را ببينيد.
جدول زير همه پيامهاي ممكن كه در نمودارهاي توالي UML وجود دارد را با نمايشهاي گرافيكي آنها نشان مي دهد.

ساده	این یک انتقال کنترل از یک شئ به شئ دیگر است.	
همزمان	اگر یک شئ یک پیام همزمان ارسال نماید، شئ براي پاسخ به پیام ، قبل ازپرداختن به کسب و کار آن صبر میکند	
غیر همزمان	اگر یک شئ یک پیام غیر همزمان ارسال نماید، شد براي پاسخ به آن قبل از اقداماتش صبر نمیکند.	

زمان

نمودار، زمان را در یک مسیر عمودي نشان میدهد. زمان در بالا آغاز میشود و به سمت پایین پیشروي مي کند. یک پیام که به بالا نزدیکتر است در زمان زودتري نسبت به پیامي که به پایین نزدیکتر است واقع شده است.



مجموعه نمادهاي ضروري براي یک نمودار توالي به همراه نمادهایي که با همدیگر کار میکنند.

نماد آدمک توالي را آغاز میکند، اما این نماد آدمک بخشي از مجموعه نمادهاي نمودار توالي نیست.

در نمودار توالي اشیاء از چپ به راست در بالا قرار گرفته اند. هر خط حیات شئ، یک خط چین است که از شئ به طرف پایین امتداد پیدا میکند. یک فلش، یک خط حیات را به خط حیات دیگر مرتبط مینماید و پیامي از یک شئ به شئ دیگر رانشان میدهد. زمان در بالا آغاز مي شود و به طرف پایین پیشروي میکند. با وجود این یک actor بطور نوعي توالي را آغاز مینماید، نماد actor بخشي از مجموعه نمادهاي نمودار توالي نیست.

راههاي ايجاد نمودار توالي

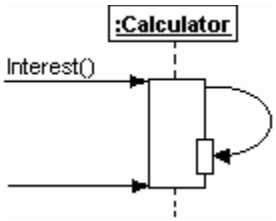
همانطور كه نمودار مورد كاربرد ميتواند يک نمونه از مورد كاربرد نشان دهد (يک سناريو) يا ميتواند عمومي باشد و تمام سناريوهاي مورد كاربرد را به يكديگر متصل نمايد ، نمودارهاي توالي نيز ميتوانند نمونه و يا عمومي باشند. نمودارهاي توالي عمومي اغلب فرصتهايي جهت نمايش عبارات if و حلقه while ارائه ميدهند. هر شرطي براي عبارت"if" را در يک براكت قرار دهيد.

ممكن است در يک نمودار توالي يک شئ ايجاد شود. هنگاميكه چنين چيزي رخ داد، شما يک شي ايجاد شده را مانند روش معمول با يک مستطيل نامگذاري شده نشان ميدهيد.

تفاوت در اينجا است كه شما آنرا در بالاي نمودار توالي همانند ساير اشياء قرار نمي دهيد. در عوض شما آن را در امتداد يک بعد عمودي قرار ميدهيد بنابراين مكان آن مشابه زمان وقتي است كه ايجاد ميشود. پيام اشياء را ايجاد ميكند با Creat() برچسب گذاري ميگردد.

گاهي يک شي عملي دارد كه خودش را طلب ميكند. اين امر بازگشت ناميده ميشود.

فرض كنيد كه يک شي در سيستم شما يک ماشين حساب است و فرض كنيد يكي از عمليات آن انجام محاسبات دلخواه است. به منظور انجام محاسبات مركب براي يک چارچوب زماني خاص كه شامل چندين دوره زماني ميباشد، عمليات محاسبات مربوطه بايد خود تعدادي از زمانهاي خاص را بيفزايد. نمودار توالي براي آن به صورت زير است:



نمايش بازگشت در نمودار توالي

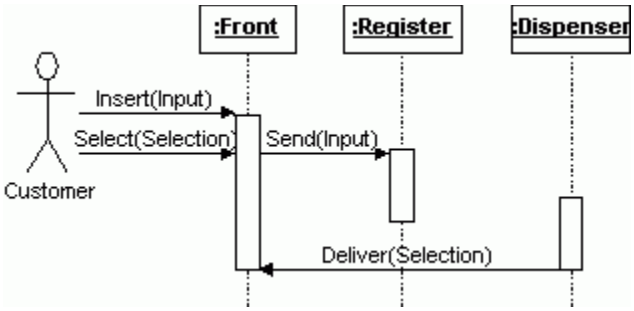
نمودار توالي- مثال

در اين مثال نمودار توالي براي يکي از مثالهاي قبلي-دستگاه سلف سرويس ارائه خواهد شد. مثال نمودار مورد کاربرد را ببينيد.
بياييد فرض کنيم در دستگاه سلف سرويس سه شئ کاري را که در ارتباط با آن هستيد انجام ميدهند:

- **قسمت جلو دستگاه:** رابط کاربري که دستگاه سلف سرويس به مشتري ارائه ميدهد.
- **ثبت کننده پول:** بخشي از دستگاه که پولها در آنجا جمع آوري مي گردد.
- **توزيع کننده**: که محصول انتخاب شده را به مشتري تحويل ميدهد.

نمودار توالي نمونه ممکن است با اين توالیها طراحي گردد:

١. مشتري پول را در شيار مخصوص پول مي اندازد.
٢. مشتري يک انتخاب انجام ميدهد.
٣. پول به محل ثبت پول منتقل ميگردد.
٤. ثبت کننده پول چک ميکند که آيا محصول انتخاب شده در توزيع کننده وجود دارد .
٥. ثبت کننده پول، ذخيره خود را به روز ميکند.
٦. ثبت کننده پول توزيع کننده اي دارد که محصول را از قسمت جلوي دستگاه تحويل ميدهد.

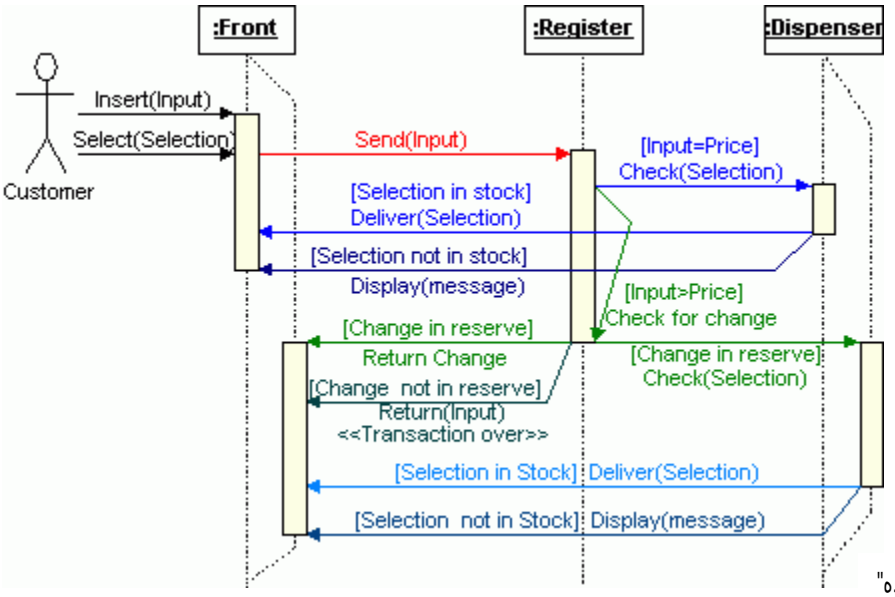


سناريوي"خريد محصول". چون اين مورد بهترين مورد سناريو است يک نمودار توالي نمونه است،

اما هنگاميکه نمودارهاي مورد کاربرد توسعه داده شدند، دو سناريو اضافي جهت نشان دادن مورد تمام شدن محصول و مورد پول نادرست معرفي ميشوند. اگر هنگاميکه يک نمودار توالي رسم ميکنيد، تمام سناريوهاي يک مورد کاربرد را در نظر بگيريد،ي ک نمودار توالي کلي ايجاد نموده ايد.
سناريوي محصول تمام شده اين توالیها را ايجاد مينمايد:

١. پس از انتخاب يک جنس که ديگر در دستگاه وجود ندارد پيام "اين جنس تمام شده" نشان داده شود.

۲. بدون معطلي بايد يک انتخاب ديگر نمايش داده شود.
۳. بايد دکمه اي وجود داشته باشد که مشتري جهت برگرداندن پول خود بتواند آنرا انتخاب نمايد.
٤. اگر مشتري جنسي را که موجود است انتخاب نمود، همه چيز همانند سناريوي مورد خوب که مقدار پول وارد شده به دستگاه درست است پيش ميرود. در غير اين صورت دستگاه از سناريوي مقدار پول نادرست پيروي ميکند.
٥. مشتري يک جنس موجود را انتخاب ميکند يا دکمه اي را که پول را به او برميگرداند فشار ميدهد.



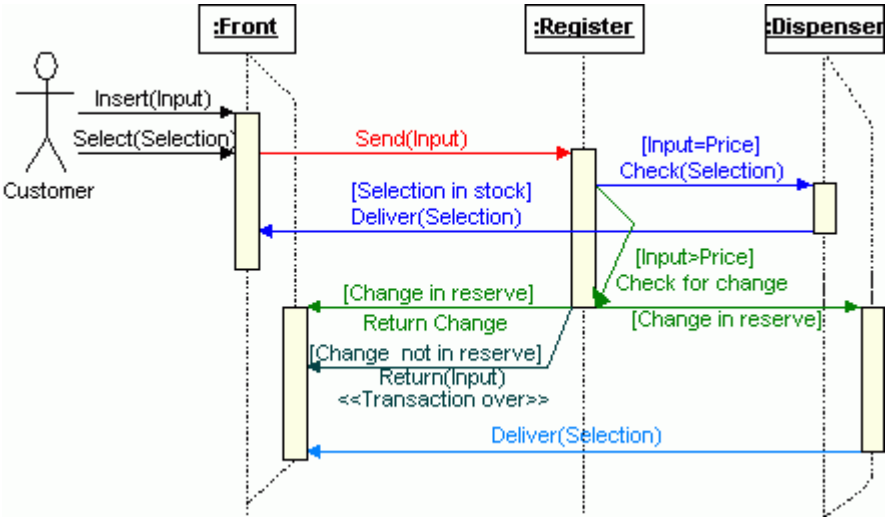
سناريو"محصول تمام شده"

مراقب يک انشعاب از کنترل در پيام به سبب شرايط if باشيد. چون هر مسير به شئ يکساني ميرود، انشعاب باعث ايجاد يک "شاخه" از کنترل در خط حيات شئ دريافت کننده ميشود، خط حيات را به مسيرهاي جدا تفکيک ميکند. در برخي موارد در توالي، شاخه ها در پيام يکي ميشوند ، همانطوريکه انشعاب در خط حيات انجام ميدهد.

براي سناريو مقدار پول نادرست فهرستي از توالیها در زير آورده شده است:

۱. ثبت کننده پول، پولي که مشتري وارد دستگاه کرده را با قيمت محصول چک ميکند.
۲. اگر مبلغ پرداخت شده بيشتر از قيمت است، اختلاف محاسبه ميگردد و ثبت کننده پول خردي که بايد پس داده شود را چک ميکند.
۳. اگر اين اختلاف محاسبه شده به صورت پول خرد وجود داشته باشد، ثبت کننده باقي پول را به مشتري برميگرداند و همه چيز مانند قبل پيش ميرود.

٤. در غير اينصورت، ثبت كننده پول وارد شده به دستگاه را برميگرداند و پيامي نمايش ميدهد كه مشتري مقدار پول درست را وارد كند.
٥. اگر پول وارد شده كمتر از قيمت محصول بود، ثبت كننده كاري انجام نميدهد و منتظر پول بيشتر مي ماند.



سناريو "مقدار پول نادرست"

نمودارهاي همكاري

نمودارهاي همكاري همانند نمودارهاي توالي نشان ميدهند كه اشياء چگونه در تعامل هستند. اين نمودار اشياء را به همراه پيامهايي كه از يكي به ديگري منتقل ميشود را نشان ميدهد. اما اگر نمودار توالي اين كار را انجام ميدهد چرا UML نياز به نمودار ديگري دارد؟ آيا آنها كار يكساني انجام نمي دهند؟

نمودار توالي و نمودار همكاري مانند هم هستند. آنها بطور معنايي هم ارزند؛ يعني اطلاعات يكسان ارائه ميدهند و شما ميتوانيد نمودار نمودار توالي را به نمودار همكاري تبديل نماييد و برعكس: مهمترين تفاوت بين آنها اين است كه نمودار توالي با توجه به زمان مرتب شده و نمودار همكاري با توجه به فضا مرتب گرديده است.

يک نمودار همكاري بسطي از نمودار شي است. علاوه بر ارتباطات بين اشياء، نمودار همكاري، پيامهايي كه اشياء به يكديگر مي فرستند را نشان مي دهد.

پيامهاي بين شي با يک فلش كه به گيرنده پيام اشاره مي كند، نزديک به خط ارتباط بين دو شي نشان داده مي شود. برچسبي نزديک به فلش نشان مي دهد كه پيام چيست. پيام بطور نمونه به شي گيرنده آن ميگويد كه يكي از عمليات خود را اجرا نمايد. يک جفت پرانتز پيام را خاتمه مي دهد. در داخل پرانتز، پارامتري را كه عمليات روي آن كار انجام مي دهد را قرار ميدهيد.

نمودارهاي همكاري مي توانند به نمودارهاي توالي تبديل شوند و برعكس. بنابراين شما مي توانيد اطلاعات توالي را در نمودار همكاري نشان دهيد.براي انجام اين كار بايد بدانيد كه:

يک عدد مطابق با عدد دستور پيام د رتوالي به برچسب پيام اضافه نماييد. دو نقطه(:) عدد و پيام را از يكديگر جدا ميكند.

اينها اساس نمودار همكاري است. بعلاوه دو درس آينده روند نوشتن نمودار همكاري را شرح ميدهد.

نوشتن نمودارهاي همكاري

شما ميتوا نيد تغييرحا لت شئ را در يک نمودار همكاري نشان دهيد. همچنين ميتوانيد به همان طريقي كه شرايط را در نمودار توالي نشان مي دهيد، نشان دهيد .

در مستطيل شئ ، حالت شئ مشخص است. به نمودار يک مستطيل ديگر اضافه كنيد كه نماينده شئ است و حالت تغيير پيدا كرده را نشان مي دهد. اين دو را با خط چين به هم مرتبط نماييد و خط چين را با يک كلمه كليشه اي <<become>> بر چسب گذاري نماييد.

شرط را داخل يک براكت[] قرار مي دهيد و مهمترين چيز اين است كه شرايط با عدد گذاري هماهنگي داشته باشند .

براي بدست آوردن مفاهيم دقيق تر به مثال نمودار همكاري مراجعه نماييد.

همچنين شما ميتوانيد ايجاد شئ را ارائه دهيد.

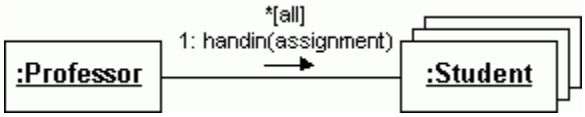
در روند ايجاد شئ شما يک كليشه <<creat>> به پيامي كه شئ را ايجاد مي كند اضافه ميكنيد.

در نمودار همكاري مي توانيد شئ شما يک كليشه <<creat>> به پيامي كه شئ را ايجاد مي كند اضافه ميكنيد .

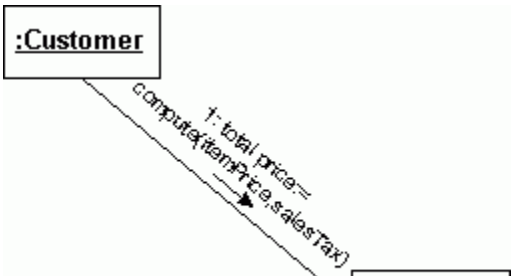
در نمودار همكاري مي توانيد اشياء چند گانه و نتايج برگشتي را نشان دهيد.

نمايش اشياء چند گانه چند رديف مستطيل است كه به سمت عقب زياد مي شوند. شما يک شرط كه داخل براكت است و قبل از ان يک ستاره قرار داده شده براي مشخص نمودن اينكه اين پيام به همه اشياء مي رود اضافه مي كنيد .

نتايج بر گشتي بصورت يک توضيح كه يک نام از مقدار برگشتي را دارند در سمت چپ نوشته ميشوند و در ادامه آن علامت = : و بعد نام عمليات و ليستي از صفحات ممكني كه عمليات ميتواند جهت توليد نتيجه بپذيرد آورده ميشود. سمت راست پيام، امضاي پيام ناميده مي شود .



استادي از گروهي از دانش آموزان درخواست ميكند كه تكاليف خود را تحويل دهند.



يک پيام ميتواند يک درخواست براي يک شئ جهت انجام محاسبه و برگرداندن بها باشد.

در برخي تعاملات، يک شئ ويژه جريان را کنترل ميکند. اين شئ فعال ميتواند به اشياء غير فعال پيام ارسال نمايد و با ساير اشياء فعال در تعامل باشد. همچنين ممکن است شما با شئ اي برخورد کنيد که يک پيام را فقط بعد از يکسري پيام که ارسال شد(احتمالاً غير متوالي)، ارسال نمايد. به بيان ديگر شئ بايد پيام خود را با مجموعه پيامهاي ديگران همزمان نمايد.

نمودار همکاري، شئ فعال را مانند سايرين نمايش ميدهد ولي کادر آن کلفت و ضخيم است. پيامهاي همزمان کننده با ليستي از پيامها که بايد قبل از اينکه پيامها رخ دهند کامل شده باشند، در اولويت هستند. يک ويرگول موارد ليست شده را از هم جدا ميکند و ليست با يک علامت مميز تمام ميشود(/).

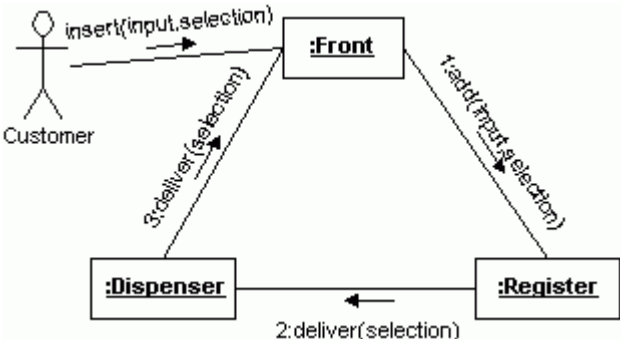
اکنون که شما تمام اين مفاهيم را دريافت کرده ايد، آماده ايد که به درس بعد، مثال نمودار همکاري برويد.

نمودار همكاري – مثال

دستگاه سلف سرويس دوباره در اين مثال هدف ما خواهد بود. در مثال قبل (مثال نمودار توالي را ببينيد) نمودار توالي براي اين سيستم رسم شد. اكنون زمان خوبي براي تبديل توالیها به نمودارهاي همكاري است.

سناريو مورد خوب شامل مراحل متعددي است:

۱. مشتري پول را در دستگاه مي اندازد و يک يا چند محصول ارائه شده در دستگاه را انتخاب مينمايد.
۲. هنگاميكه ثبت كننده پول، پول را دريافت ميكند (در اين مورد مشتري مقدرا پول درست را قرار ميدهد و هميشه يک محصول از جنس انتخاب شده وجود دارد)، محصول انتخاب شده به توزيع كننده تحويل ميشود.
۳. توزيع كننده محصول را به قسمت جلوي دستگاه تحويل ميدهد و مشتري آن محصول را ميگيرد.

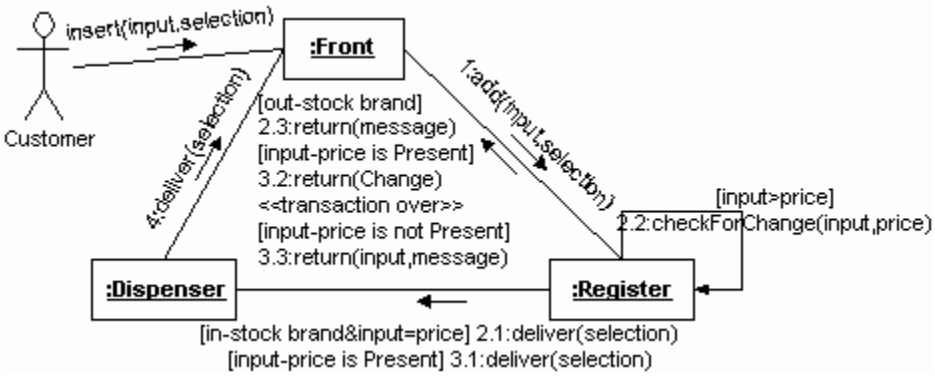


اما هنگاميكه اين دستگاه و كار آنرا به جزئيات بررسي كرديم، سناريوهاي جديدي مطرح شدند: سناريوهاي مقدار پول نادرست و تمام شدن محصول. در حين ايجاد نمودارهاي توالي، شرايط براي نمايش كار دستگاه مطرح شدند. در اينجا نيز شرايط بكار خواهند رفت.

شرايط را در يک جفت براكت [ ] قرار مي دهيد و شرط قبل از برچسب پيام مي آيد. نكته مهم اين است كه شرطها با عددگذاري هماهنگ باشند. هنگاميكه عبارتif را مدلسازي ميكنيد، دو شرط ممكن وجود دارد.

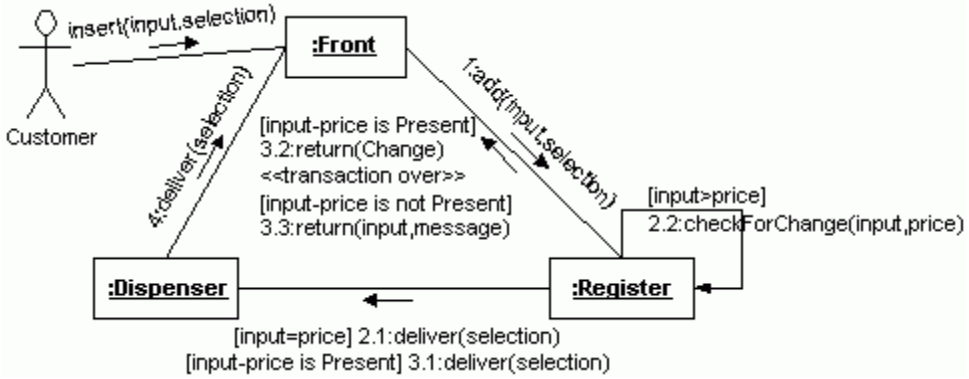
پيامها ي اين شرايط يک عدد يكسان دارند ويک نقطه اعشار و عدد ديگر را اضافه ميكنيد. اين عمل تودرتويي(nesting) نام دارد.

در سناريوي محصول تمام شده، دستگاه بايد تمام شدن محصول را نشان دهد و به مشتريان اجازه دهد که يک محصول ديگر را انتخاب نمايد و يا پول خود را پس بگيرد. اگر مشتري انتخاب ديگري انجام دهد اين روند دوباره تکرار خواهد شد. همچنين ممکن است سناريو مقدار پول درست مطرح گردد. نمودار همکاري براي اين مورد درشکل زير آمده است:



همانطور که ميتوانيد ببينيد، از اين تصوير دو سناريو ممکن است رخ دهد. محصول تمام شده و محصول موجود، دو امکان موجود هستند. اگر فرض کنيم که اين کار بعد از پيام ۱:add(input,selection) رخ دهد، عدد اين پيام ۲ خواهد بود و براي سناريوي محصول موجود عدد ۱ وبراي سناريوي محصول تمام شده عدد ۲ بعد از نقطه ممیز قرار ميگيرد.

هنگاميکه دستگاه پول خرد درست ندارد چه اتفاقي مي افتد؟ دستگاه بايد پيام پول خرد تمام شده را نشان دهد، پول را برگرداند و به مشتري براي وارد کردن مقدار پول مورد نياز پيغام دهد. در نتيجه معامله به پايان رسيده است. هنگاميکه دستگاه مقدار پول خرد درستي دارد، بقيه پول را به مشتري برميگرداند و و محصول انتخاب شده را تحويل ميدهد. دو شاخه اي که از روند تودرتويي توليد ميشوند با ۲,۱ و ۲,۲ عدد گذاري ميشوند و سه احتمال در پيام شرطي ۳ مي تواند اين باشد:۳,۱ و ۳,۲ و ۳,۳ .
نمودار همکاري در شکل زير نشان داد شده است:



اين مطالب همه چيزهاي لازم براي اين مثال و درس امروز بود. حال شما بايد به سوالاتي كه در بخش پرسشها و پاسخها و كارگاه امروز آمده پاسخ دهيد. در روز بعد سه نمودار آخر UML به شما معرفي خواهد شد.

پرسش و پاسخ

در اينجا تعداي سوال براي درک سطح دانش شما درباره آنچه امروز آموخته ايد وجود دارد.

١. از چه طريق قابل توجهي يک نمودار حالت با يک نمودار کلاس، يک نمودار شئ يا يک نمودار مورد کاربرد تفاوت دارد؟
٢. انتقال آني چيست؟
٣. چگونه ميتوانيد روند کنترل انجام شده با حالت while را در نمودار توالي نشان دهيد؟
٤. چگونه ميتوانيد يک پيام را در نمودار همکاري نشان دهيد؟
٥. چگونه ميتوانيد تغييرات حالت را در يک نمودار همکاري نشان دهيد؟

پاسخها

۱. نمودار حالت حالات یک شئ جدا را مدل سازي میکند.
یک نمودار کلاس ، نمودار شئ یا یک نمودار مورد کاربردي ک سیستم یا حداقل بخشي از سیستم را مدلسازي میکنند

۲. یک انتقال آني انتقالي است که به سبب فعالیتهایي منطبق با حالت به جاي واکنش یک رویداد رخ میدهد.

۳. براي نشان دادن جریان کنترلي که با عبارت while به آن اشاره شده است، شرط را درون براکت قرار دهید و قبل از براکت سمت چپ یک علامت ستاره قرار دهید.

٤. براي نشان دادن پیامها در نمودارهاي همکاري، فلش نزدیک خط وابستگي قرار داده میشودکه به شئ دریافت کننده اشاره دارد.

۵. در داخل مستطیل یک شئ حالت آنرا نشان دهید. مستطیل دیگر را براي آن شئ اضافه نمایید و حالت تغییر یافته را نشان دهید. آن دو را به یک خط به هم مرتبط نمایید و خط را با یک کلیشه <> برچسب گذاري نمایید.

روز چهارم

امروز مطالب زير را ياد خواهيد گرفت:

- ✓ **نمودار هاي فعاليت**
- ✓ چگونه يك نمودار فعاليت بسازيد .
- ✓ **نمودار هاي اجزء**
- ✓ گروه بندي اجزاء سيستم و ايجاد نمودار اجزء
- ✓ **نمودار هاي استقرار**
- ✓ تشكيلات فيزيكي سيستم

نمودار هاي فعاليت(activity diagram)

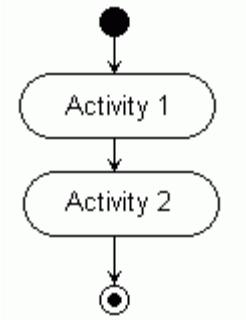
اگر تا به حال يك دوره مقدماتي از برنامه نويسي گذرانده باشيد ، احتمالاً با **فلوچارت** مواجه شده ايد. فلوچارت دنباله اي از مراحل ، فرايندها ، نقاط تصميم گيري و شاخه ها مي باشد . برنامه هاي مبتدي ، براي تصور مشكلات و يافتن راه حلها ، به استفاده از فلوچارت ترغيب شده اند .

نمودار فعاليت UML بسيار شبيه فلوچارت هاي قديمي است كه مراحل (كه فعاليت ها ناميده مي شوند) ، همانند نقاط تصميم گيري و شاخه ها را نشان مي دهد. نمودار فعاليت براي نشان دادن آنچه در فرايند هاي كسب و كار يا در يك عمل رخ مي دهد، بسيار مفيد است . شما آن را جزء لازم تحليلهاي سيستم در نظرخواهيد گرفت .

در ابتدا و پيش از هر چيز، يك نمودار فعاليت براي آنكه نگاه ساده اي به آنچه در طول يک عمل يا يک فرايند رخ مي دهد داشته باشد، طراحي شده است و توسعه نمودار حالت مي باشد. نمودار حالت، حالت هاي يک شئ را نشان مي دهد و فعاليت ها را آنطور كه فلشها به حالت ها متصل شده اند، بيان مي كند. نمودار فعاليت، فعاليت ها را بارز مي كند.

هر فعاليت با يک مستطيل گرد شده، باريكتر و بيشتر به شكل بيضي در آمده، نسبت به شمايل حالت، نشان داده مي شود. فرايند درون يک فعاليت كامل مي شود و سپس انتقال خودكار به فعاليت بعدي صورت مي گيرد. يک فلش، انتقال يک فعاليت به فعاليت بعدي را نشان مي دهد.

همچنين هر نمودار فعاليت يک نقطه شروع دارد كه با يک دايره توپر نشان داده مي شود و يک نقطه پاياني دارد كه با يک دايره هدف، نشان داده مي شود.



انتقال از يک فعاليت به فعاليت ديگر در نمودار فعاليت

در درس بعدي راه هاي ايجاد يک نمودار فعاليت و قوانين بين فعاليت ها توضيح داده مي شود و در انتها مثالي براي اين نوع از نمودارها به شما ارائه مي شود .

نمودار فعاليت، توسعه نمودار حالت مي باشد. نمودار هاي حالت، حالتها را بارز ميكند و فعاليت ها را بصورت فلشهايي بين حالت ها نشان مي دهد. نمودار هاي فعاليت، بر فعاليتها تمركز مي كنند. هر فعاليت به صورت يک مستطيل گرد شده نشان داده مي شود كه در ظاهر خيلي بيشتر ازشمايل حالت،

بيضي شده است. نمودار فعاليت براي نقطه شروع و نقطه پايـاني از نـشان هـاي مـشابه نمـودار حالـت استفاده مي كند.

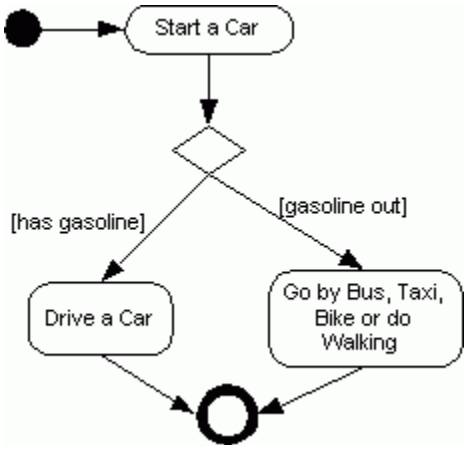
ساختن یک نمودار فعالیت

در این درس روش هاي ارائـه تـصمیمات، مـسیرهاي همزمـان، علائـم و swimlanes در نمودارهـاي فعالیت توضیح داده خواهد شد .

تصمیمات

نقاط تصمیم گیري مي توانند به دو طریق نشان داده شوند و بستگي به شما دارد که چگونـه از آنهـا در نمودار هاي فعالیت خود استفاده کنید.

روش اول نشان دادن مسیرهاي ممکن که مستقیماً از یک فعالیت بیرون مي آیند، مي باشد.
روش دیگر، انتقال فعالیت به یک لوزي کوچک مي باشد که مسیر هاي ممکن از این لوزي خارج شـوند.
در هر کدام از این روشها، شرط را با یک عبارت شرطي درون علامت [] نزدیـک مـسیر مناسـب، نـشان دهید.
تصور کنید که باید به محل کارتان بروید. به سمت ماشین تان مي روید و سوئیچ را درون اسـتارت قـرار میدهید. حال دو حالت وجود دارد : ماشین استارت میزند، یا ماشین استارت نمي زند. این حالـت هـاي ممکن سبب به وجود آمدن دو فعالیت دیگر مي شوند : راندن ماشین و یا رفتن به محل کار با اتوبـوس، تاکسي، دوچرخه یا پیاده رفتن. آنچه در بالا گفته شد در شکل زیر نشان داده شده اسـت (بـه دو روش نشان داده شده براي تصمیمات توجه کنید) :

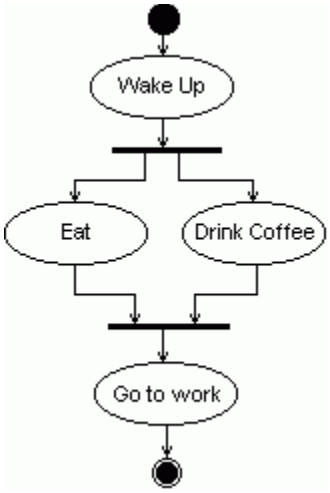


نمودار فعالیت براي نشان دادن دو روش تصمیم گیري

مسير هاي همزمان

وقتي فعاليت ها را مدلسازي ميكنيد، مكرراً موقعيت هاي مناسبي داريد كه يک انتقال را به دو مسير مجزا كه در يک زمان واحد (همزماني) اجرا مي شوند و سپس به هم ميرسند، تقسيم كنيد.

جدا كردن مسيرها، توسط يک خط توپر پررنگ عمودي نشان داده مي شود كه براي انتقال و نشان دادن مسيرهايي كه ازخط خارج مي شوند، مي باشند. براي يكي كردن مسيرها نقطه انتهايي پيكان مسيرها را در يک خط توپر پررنگ ديگر نشان مي دهيم.



نمودار فعاليت يک انتقال را كه به دو مسير كه همزمان اجرا مي شوند و سپس بهم ميرسند، جدا ميكند.

علائم(Signals)

در طي توالي اي از فعاليت ها، ميتوان يک علامت ارسال نمود. وقتي علامت رسيد، سبب ميشود كه يک فعاليت رخ دهد.

نمادي كه براي فرستادن يک علامت به كار مي رود، يک چند ضلعي محدب و نمادي كه براي دريافت يک علامت به كار مي رود، چند ضلعي مقعر مي باشد.

ارسال و دريافت يک علامت

Swimlanes

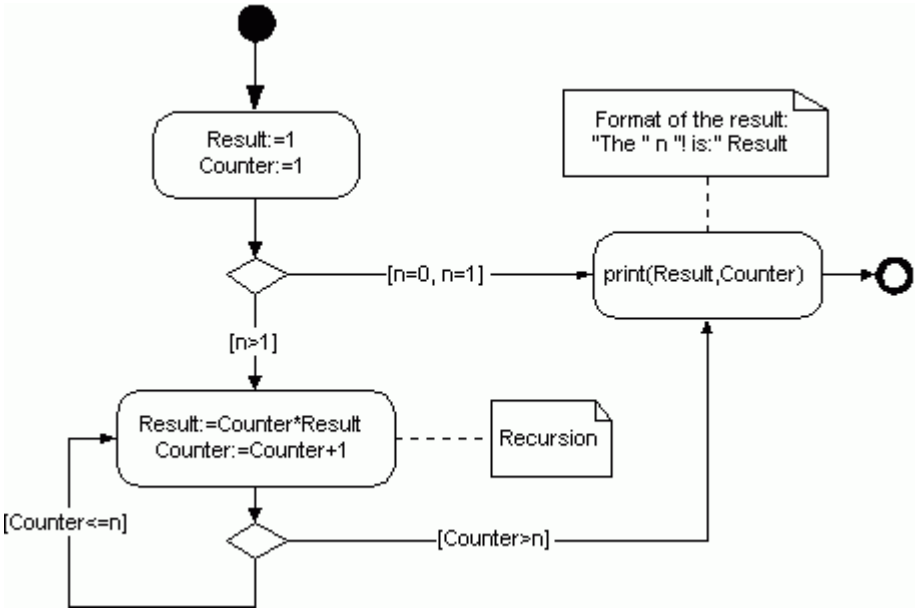
نمودار فعاليت، بعد بصري سازي نقشها را اضافه مي كند. براي انجام اين كار، نمودار را به قسمت هاي موازي كه Swimlanes ناميده مي شوند، تقسيم كنيد. هر Swimlanes نام يک نقش را در بالا نشان مي دهد و فعاليت هاي هر نقش را ارائه مي كند.

ممكن است كه نمودار فعاليت را با نماد هاي نمودارهاي ديگر تركيب كنيد و يک نمودار پيوندي hybrid diagram توليد كنيد.

در درس بعدي مثالي كه براي يک نمودار فعاليت نوشته شده است، به شما داده خواهدشد.

نمودار فعاليت - مثال

اين مثال با رياضيات حل شده است. بعضي اوقات نياز داريد كه فاكتوريل n يعني !n را محاسبه كنيد كه فرمول آن به صورت n!=n*(n-١)*(n-٢)*...*٢*١ مي باشد.
بياييد به تفحص درباره مساله بپردازيم. از تعريف فاكتوريل داريم ٠!=١ و ١!=١ و براي بقيه اعداد بايد از فرمول داده شده استفاده كنيم. در برنامه نويسي كامپيوتر، اين مساله با روش بازگشتي نوشته مي شود. در اينجا همزمان با ساختن يک نمودار فعاليت، روش بازگشتي معرفي خواهد شد.
مي توانيد نام عمليات را computeFact(n) بگذاريد. براي انجام عمليات نياز به يک شمارنده داريد كه مشخص كند آيا عمليات به فاكتوريل n ام رسيده يا نه. متغيري كه توالي شمارش ها را نگهدارد و دو متغير ديگر كه مقادير ٠! و ١! را نگهدارند.
يک نمودار فعاليت تكميل شده، در شكل زير نشان داده شده است.



اگر با برنامه نويسي كامپيوتر سروكار داريد، به اين نتيجه خواهيد رسيد كه اين نمودار فعالي ، بسيار شبيه يک روش اجرايي(عملكرد) كه فاكتوريل ها را محاسبه مي كند، مي باشد.

اطلاعات بيشتري درباره رابط ها و اجزاء

در درس هاي پيشين نمودارهايي را آموختيد كه با موجوديتهاي مفهومي سروكار داشتند. در درس هاي آتي، نمودار UML اي را خواهيد آموخت كه يک موجوديتهاي حقيقي را نشان مي دهـد: اجـزاي نـرم افزار.

اما اجزاي نرم افزار(software component)چيست ؟ اجزاي نرم افزار، بخش فيزيكي سيـستم مـي باشد و در يک كامپيوتر مستقر است، نه در ذهن يک تحليلگر.
چه رابطه اي بين يک جزء و يک كلاس وجود دارد ؟
يک جزء را همانند پياده سازي نرم افزار يک كـلاس در نظـر بگيريـد. كـلاس چكيـده اي از صـفات و عمليات را بيان مي كند و يک جزء مي تواندپياده سازي اي بيش از يک كلاس باشد.
اجزا و روابط بين آنها به اين منظور مدلسازي ميكنيد كه:

١. Clients ها بتوانند ساختار تمام شده سيستم را مشاهده كنند.
٢. توسعه دهندگان، ساختاري براي كارهاي آينده داشته باشند.
٣. نويسندگان فني اي كه بايد مستند سازي را ميس سازند و به فايل هـا كمـک كننـد، ميتواننـد بفهمند كه راجع به چه مي نويسند.
٤. براي دوباره استفاده كردن آماده باشيد.

وقتي با اجزا سروكار داريد، با رابط هاي آنها هم سروكار پيدا خواهيد كرد.
يک رابط(interface) مجموعه اي از عمليات مي باشد كه برخي مسائل مربوط به رفتار كلاس را مشخص مي كند همچنين مجموعه اي از عمليات است كه يک كلاس، براي كلاس هاي ديگر نشان مي دهد. براي شما به عنوان يک مدلساز، اين امر به آن معناست كه روشي كه شما يک رابط را براي يک كلاس ارائه مي دهيد، مشابه روشي است كه يک رابط را براي يک جزء ارائه مي دهيد. در اين حالت براي يک كلاس و رابط هاي آن، رابطه بين يک جزء و رابط هاي آن، محقق سازي(realization) ناميده مي شود .
مي توان يک جزء را با جزء ديگري جايگزين كرد به شرط آنكه جزء جديد، با رابـط هـايي مـشابه جـزء قبلي مطابقت داشته باشد. مي توان از يک جزء مجدداً در سيستم ديگري استفاده كـرد بـه شـرط آنكـه سيستم جديد بتواند به جزئي كه استفاده مجدد از آن شده از طريق آن رابط جزء، دسترسي پيدا كند.
هر چه در مسير مدلسازي پيش مي رويد، با سه نوع جزء سروكار مي يابيد:

۱. **اجزاي استقرار**، كه اساس سيستم هاي اجرا پذير را مي سازند (DLL’s، اجراپذيرها، كنترلهاي ActiveX ، javaBean ها)
۲. **اجزاي محصول كار**، كه از اجزاي پياده سازي بوجود آمده اند. (data file ها و source code file ها)
۳. **اجزاي اجرا** ، به عنوان نتيجه يک سيستم در حال اجرا به ايجاد ميشوند.

به جاي ارائه يک موجوديت مفهومي، مانند يک كلاس يا يک حالت، نمودار اجزا كه يک موارد واقعي را نشان مي دهد- يک جزء نرم افزار- ارائه مي دهد. اجزاي نرم افزار، دركامپيوترها مستقر مي باشند، نه در ذهن تحليلگران. يک جزء از طريق رابط هايش قابل دسترسي است. رابطه بين يک جزء و رابط اش محقق سازي ناميده مي شود. وقتي يک جزء به خدمات جزء ديگري را دسترسي دارد، از يک رابط دوروني استفاده مي كند.

نمودار هاي اجزا(component diagrams)

يک نمودار اجزا شامل اجزا، رابط ها و رابطه بين آنها مي باشد. همچنين انواع نشان هاي ديگري کـه تـا کنون خوانده ايد، در يک نمودار اجزا نيز ديده ميشوند.

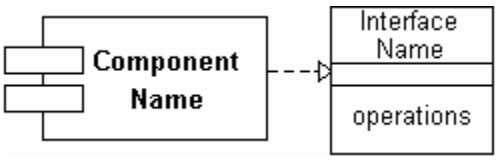
شمايل اصلي نمودار اجزا يک مستطيل است که دو مستطيل در سمت چپ قرار گرفته اند. نـام جـزء درون شمايل نوشته مي شود. ايـن نـام بـه صـورت رشـته (string) مـي باشـد. اگـر جـزء، عـضوي از يکpackage باشد، مي توانيد نام جزء را با نام package به صورت پيشوند بياوريد.

شمايل يک جزء

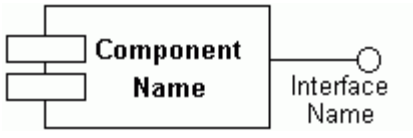
جزء و رابط هايي که در نمودار اجزا وجود دارند به دو روش قابل نمايش هستند:

يک روش آن است که رابط به صورت مستطيلي که شامل اطلاعات وابسته به رابـط مـي باشـد، نـشان داده شود. رابط با يک خط چين و يک مثلث توخالي که realization را به تصوير مي کـشد، بـه جـزء متصل است.

روش ديگر براي نشان دادن رابط يک دايره کوچک است که با يک خط به جزء متصل است کـه ارتبـاط realization را نشان مي دهد.



رابط را ميتوان با يک مستطيل که حاوي اطلاعات است و با فلش realization به جزء مرتبط است نشان داد.



رابط يک دايره کوچک است که با يک خط به جزئي که نمايانگر realization است متصل ميشود

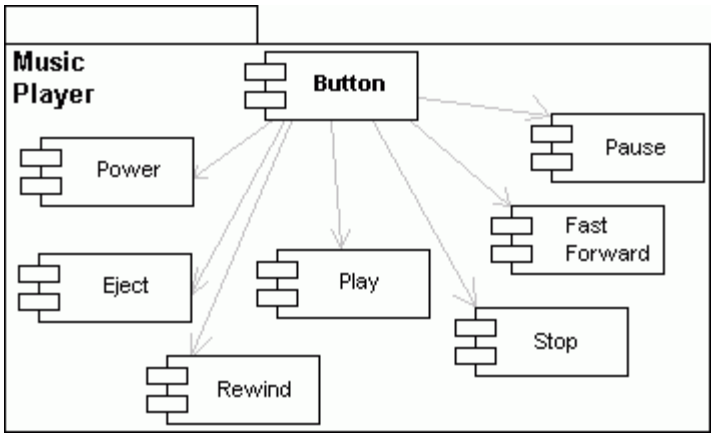
نمودار اجزا – مثال

فرض کنید که احتیاج داریم براي اجراي آهنگي از درايو CD-ROM، يک نرم افزار بسازيم. براي ايـن منظور، يک زبان برنامه نويسي بصري مانند ويژوال بيسيک يا دلفي، مي تواند مورد استفاده قـرار گيـرد. اگر اين زبان، ازکنترل هاي مولتي مديا پشتيباني کند، مي توانيم از اجـزاي آن هـر جـا کـه لازم باشـد استفاده کنيم، يا مي توانيم از اجزاي جديد استفاده کنيم. يک طرح گرافيکي ممکن، براي اجـرا کننـده آهنگ ما مي تواند به صورت زير باشد:

چنانچه مي بينيد اين اجراکننده آهنگ UML به کنترل هاي زير نياز دارد:

- play
- stop
- eject
- pause
- fast forward
- rewind
- power

اين کنترل ها توسط دکمه ها (buttons) تحقق مي يابند. بنابراين براي نمايش ايـن کنتـرل هـا يـک دکمه خواهيم داشت. اگر به دکمه ها به عنوان يک اجزاي جدا از هم نگاه کنيم، ميتـوانيم يـک نمـودار UML اجزاء استخراج نماييم. آنچه گفته شد در شکل زير نشان داده شده است:



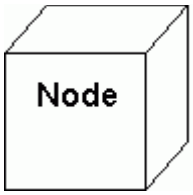
نمودار جزء براي اجراكننده آهنگ

همه اجزايي كه در نمودار بالا نشان داده شده اند به يک جزءكلي- دكمه تعلق دارند، اما فعاليتي كه آنها نشان مي دهند متفاوت است و بايد با برنامه نويسي آنها، اين فعاليتها را بدست آوريم.

نمودار هاي استقرار

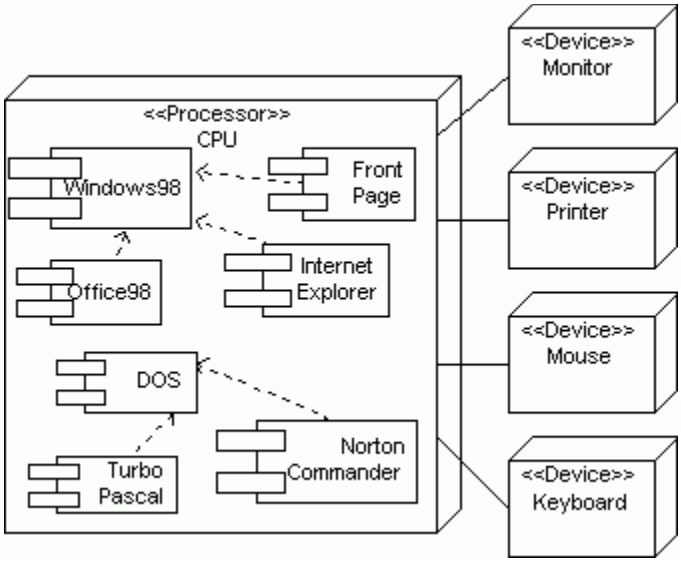
اكنون كه به انتهاي نمودار هاي UML رسيده ايم. فرصت مناسبي است كه نگاهي به سخت افزار داشته باشيم. چنانچه مي توانيد ببينيد، ابتدا از مواردي كه در تحليلها قرار داشتند به اجزايي كه در كامپيوتر ها قرار دارند رفتيم و از آنجا به سخت افزاري كه در دنياي حقيقي قرار دارد مي رويم.
سخت افزار، يک موضوع اوليه و ابتدايي در يک سيستم چند جزئي است. يک برنامه كاري محكم براي پياده سازي سخت افزار در يک طراحي سيستم بسيار اساسي است. UML، با استفاده از نمادها شما را براي ايجاد يک تصوير واضح از چگونگي نصب سخت افزار نهايي، آماده مي سازد
مورد اصلي سخت افزار يک node مي باشد، كه يک نام كلي براي انواع منابع محاسباتي مي باشد. دو نوع از انواع node ميسر است. يک **پردازشگر** يک node است كه مي تواند يک جزء را به اجرا درآورد، و يک **دستگاه** نيز يک node مي باشد كه نمي تواند يک جزء را به اجرا درآورد. يک دستگاه، نوعاً از راههاي مختلفي با دنياي بيرون ارتباط برقرار مي كند.

در UML ، يک مكعب نشان دهنده يک node است. node نام خود را دارد و مي توانيد از يک كليشه براي نشان دادن اينكه چه نوع منبعي است، استفاده كنيد. اگر node بخشي از يک package باشد، آنگاه نام package جلوتر از نام node خواهد بود. يک خط كه دو مكعب را به هم وصل ميكند، اتصال بين دو node را نشان مي دهد. مي توانيد از يک كليشه براي فراهم نمودن اطلاعات مربوط به اين اتصال، استفاده كنيد.



مكعب ، يک node را در UML نشان مي دهد

همچنين هر node ، برخي اجزاي نرم افزار را در سيستم استقرار ميدهد. براي نشان دادن اجزاي مستقر شده، آنها را در رابطة وابسته با node نشان مي دهيم.
حال بياييد يک نمودار استقرار براي يک سيستم كامپيوتر خانگي بسازيم. كامپيوتر از اجزاء سخت افزاري: CPU، مانيتور، ماوس، صفحه كليد و اجزاي نرم افزاري نصب شده: ويندوز ۹۸ ، آفيس ۹۸ ، جستجوگر اينترنت نسخه ٤ ، front page، DOS ، سفارش دهنده نورتون و توربو پاسكال تشكيل شده است.



node اصلي CPU مي باشد كه همه اجزاي نرم افزار همراه با روابط آنها را در بر مي گيرد. بقيه اجزاي سخت افزاري دستگاه هايي هستند كه به واحد پردازشگر اصلي متصل شده اند .

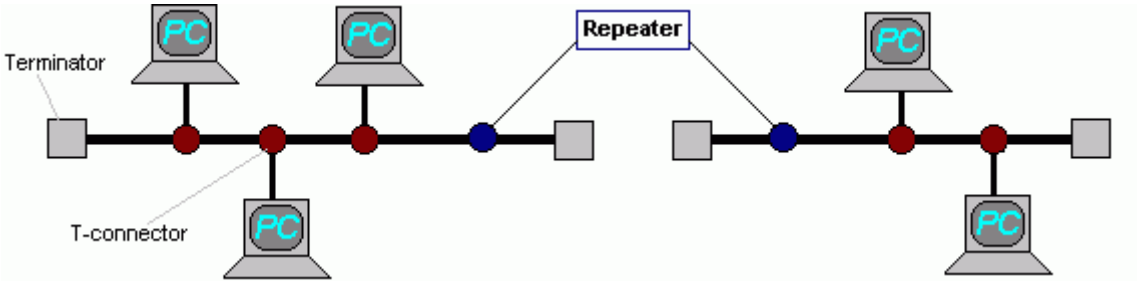
مي توانيد اين نمودار را با افزودن مودم و ارتباط با اينترنت، توسعه دهيد. سعي كنيد ايـن نمـودار را بـا تعميم مواردي كه در زير آمده، دوباره رسم كنيد.

نمودار استقرار UML تصويري فراهم سازد كه نشان دهد سيستم فيزيكي وقتي همه با هـم قـرار مـي گيرند، چگونه رفتار مي كند. سيستم شامل node هايي (كه با مكعب نشان داده مي شوند) است كه به وسيله خط هايي كه اتصالات ناميده مي شوند، به هم متصل هستند. دو نوع node وجود داشـته باشـد كه يكي پردازشگري باشد كه بتواند يک جزء را اجرا كند و ديگري يک دسـتگاه كـه بـا دنيـاي حقيقـي ارتباط برقرار مي كند. نمودار هاي استقرار براي مدل سازي شبكه ها بسيار مفيد هستند.

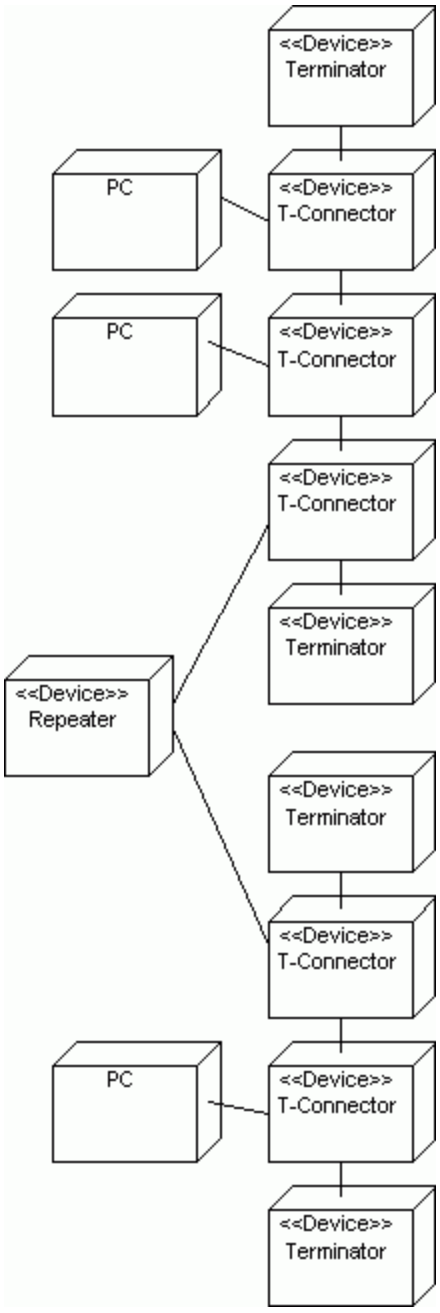
نمودار استقرار – مثال

در اين مثال شبكه Ethernet را براي شما شرح مي دهيم. اگر با شبكه هاي كامپيوتري آشنا هستيد، اين مثال برايتان بسيار آسان خواهد بود، در غير اين صورت سعي كنيد توضيح زير را بفهميد:
Etherne كم حجم، يک شبكه محبوب به حساب مي آيد و در مكان هاي محلي مانند اتاق ها يا ساختمان هايي كه كابل اتصال مي تواند قرار بگيرد، استفاده مي شود. محاسبات رياضي وجود دارد كه طول كابل كل شبكه و طول كابل از يک كامپيوتر به كامپيوتر ديگر را محاسبه مي كند. اين محاسبات براي نمودار استقرار چندان اهميتي ندارند.
كامپيوترها از طريق وسيله هايي كه T-connectors ناميده مي شوند به كابل شبكه متصل مي شوند. گاهي اوقات اين اتصال از طريق وسيله اي به نام vampire-tap صورت مي گيرد .
T-connectors سه نقطه اتصال دارد و شكل آن بسيار شبيه حرف T مي باشد و به همين دليل است كه اين نام را به آن داده اند. كابل شبكه از يک نقطه انتهايي T-connectors به نقطه انتهايي ديگر مي رود و سومين نقطه انتهايي كابلي دارد كه به كامپيوتر وصل مي شود.

به علت آن كه طول شبكه محدود است نقطه انتهايي كابل بايد دستگاهي به نام ترميناتور داشته باشد كه پايداري نامحدودي به انتهاي كابل بدهد. همچنين يک بخش شبكه مي تواند از طريق يک تكرار كننده به بخش هاي ديگر بپيوندد. تكراركننده وسيله اي است كه علامت را براي كاهش از دست دادن آن علامت، تقويت مي كند. اترنت كم حجم در شكل زير ارائه شده است:



حال كه با مفهوم شبكه اترنت كم حجم ، بياييد نمودار پياده سازي براي آن رسم كنيم. اين نمودار به صورت زير خواهد بود:



نمودار پياده سازي براي شبكه اترنت

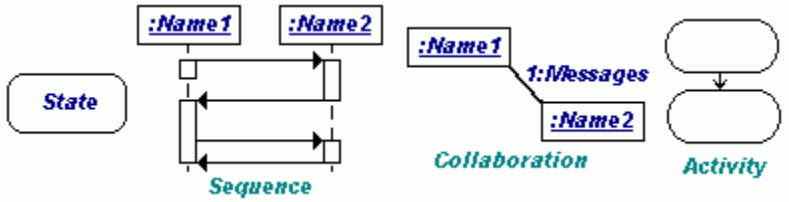
با اين مثال، همه نمودار هاي UML به شما نشان داده مي شود. در اين جلسه تنها كاري كه بايد انجام دهيد اين است كه تمرين ها را انجام دهيد اما پيش از آن، يک تصوير كلي از تمام نماد هاي UML به شما ارائه مي شود:.

مجموعه اي از نماد هاي نمودار UML

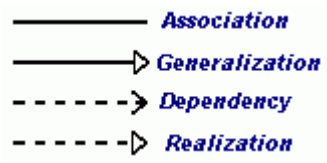
در اينجا مجموعه كاملي از نماد هايي كه UML در نمودار خود معرفي مي كند وجود دارد :

عناصر ساختاري

عناصر رفتاري

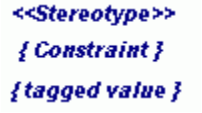


روابط

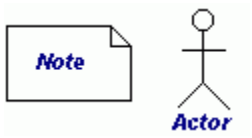


گرو هبندي

extension



حاشيه نويسي و actorها



پرسش و پاسخ

در اين جلسه مجموعه نمودار هاي UML را به پايان رسانده ايد. حال دانش خود را نسبت به نمودار هاي فعاليت، اجزا و نمودار هاي استقرار بسنجيد.

١. دو راه مختلف نمايش نقطه تصميم گيري در نمودار هاي فعاليت چيست؟
٢. انتقال علامت و دريافت آنرا چگونه نشان مي دهيم؟
٣. سه نوع مختلف اجزاء كدامند؟
٤. رابطه بين يک جزء و رابط آن چه ناميده مي شود؟
٥. دو نوع از انواع node ها چيستند؟

پاسخها

۱. یک راه از طریق نشان دادن یک لوزي که شاخه هايي از آن بيرون آمده اند، مي باشد. راه ديگر نشان دادن شاخه هايي مي باشد که مستقيماً از فعاليت بيرون آمده اند. در هـر يـک از ايـن دو راه بايد شرط را درون دو علامت [] ، روي هر شاخه قرار دهيم.
۲. علامت انتقال، توسط پنج ضلعي محدب و علامت دريافت شده توسط چندضـلعي مقعـر نـشان داده مي شود.
۳. سه نوع اجزاي موجود، اجزاي استقرار، اجزاي محصول کار و اجزاي extension هستند.
٤. رابطه بين يک جزء و رابط آن ، realization يا محقق سازي ناميده مي شود.
٥. دو نوع node ها در نمودار پياده سازي ، يکي پردازشگرها مي باشند کـه مـي تواننـد اجـزا را اجرا کنند و ديگري دستگاه ها که به دنياي بيرون متصل مي شوند.

کارگاه

لطفاً به سوالات زير پاسخ دهيد و نمودار هاي لازم را رسم كنيد ، اين امر دانش شما را در زمينه هاي نمودار هاي فعاليت ، نمودار هاي اجزا و نمودار هاي پياده سازي قدرت مي بخشد.

١. آيا واقعاً با هر چيزي كه نمودار مورد كاربرد نشان مي دهد ، به نمودار هاي فعاليت احتياج داريم ؟
٢. Swimlanesچيست ؟ مثال بياوريد .
٣. رابط خارجي چيست ؟رابط داخلي چيست ؟
٤. راههايي را كه مي توانيد از آن طريق يک جزء را جايگزين و يا استفاده كنيد نام ببريد .چه چيزي براي داشتن يک جزء قابل استفاده مهم است ؟
٥. چگونه مي توانيد يک node را در يک نمودار پياده سازي نشان دهيد ؟ چه نوع اطلاعاتي در يک node مي تواند ظاهر شود ؟
٦. يک نمودار فعاليت براي اتصال يک كامپيوتر به اينترنت رسم كنيد و پس از آن يک نمودار توسعه براي يک سيستم كامپيوتر خانگي كه به اينترنت وصل مي شود ، رسم كنيد .

روزپنجم

امروز مطالب زیر را یاد خواهید گرفت:

- ✓ **مدلسازي كتابخانه ديجيتالي**
- ✓ يافتن فرايند هاي كسب و كار
- ✓ تحليل حوزه
- ✓ جلسه JAD

كتابخانه ديجيتالي – مثال

تئوري نمودار UML را در جلسه پيش به پايان رسانديم. اكنون موقعيت مناسبي است كه به تمرينات مناسب بپردازيم. اين جلسه و جلسه بعد مربوط به كتابخانه هاي ديجيتالي هستند و اگر با آنها آشنايي نداريد، نگران نباشيد، پس از اين دو جلسه همه چيز برايتان بسيار واضح خواهد شد.

كتابخانه هاي كلاسيک به گونه اي است كه كتابدار با حجم انبوهي از كارهاي كاغذي سروكار دارد. هر كاربر بايد براي گرفتن كتاب به طور حضوري آنجا داشته باشد و بداند كه چه كتابي را مي خواهد و اگر كتاب مورد نظرش را نشناسد فردي كه در كتابخانه كار مي كند، كتابهاي متناسب با آن كتاب به وي پيشنهاد مي كند.

موج جديد در اين جهت، بر پايه توسعه نرم افزار است و اينترنت، در رأس موجوديت كتابخانه جديد كه **كتابخانه ديجيتالي** ناميده مي شود قرار گرفته است. اين نوع كتابخانه يک دسترسي بين المللي همراه با امكان هاي مرور سريع، كتاب هاي ديجيتال و تعداد زيادي فايل هاي مولتي مديا، ميباشد. اين كتابخانه ها يک كتاب يا يک مجله را به شما معرفي مي كنند و صورت آماري كتابهاي اخير و تعداد بسيار زيادي كتاب كه در آينده نزديک وجود خواهند داشت را ارائه مي دهند و همه اين كارها را در حالي مي توانيد انجام دهيد كه در مبل راحت خود لم داده ايد. حتي با وجود دور بودن كتابخانه، به دست آوردن يک كپي از كتاب يا مجله مورد نظر با استفاده از سيستم پستي كارآمدي كه به كامپيوتر متصل است، امكان پذير مي باشد.

جلسه پنجم و ششم، شما را در فرايند تحليل كردن و اجراي يک كتابخانه ديجيتالي، راهنمايي خواهد كرد. تمام مراحل معرفي شده در فرآيند توسعه سريع نرم افزار (Rapid Application Development) و نمودار هاي UML محصولات كار را خواهيد فهميد. دو درس نخست در طول مصاحبه، جزييات كار يک كتابخانه ديجيتال را براي شما تشريح مي كنند. حال بياييد كارمان را شروع كنيم.

يافتن فرآيند هاي كسب و كار ۱

در اين درس خواهيد آموخت كه چگونه با يک client مصاحبه كنيد. اين مورد مكالمه اي بين يک تحليلگر و يک clien است. محصول نهايي اين مكالمه يک نمودار فعاليت خواهد بود.

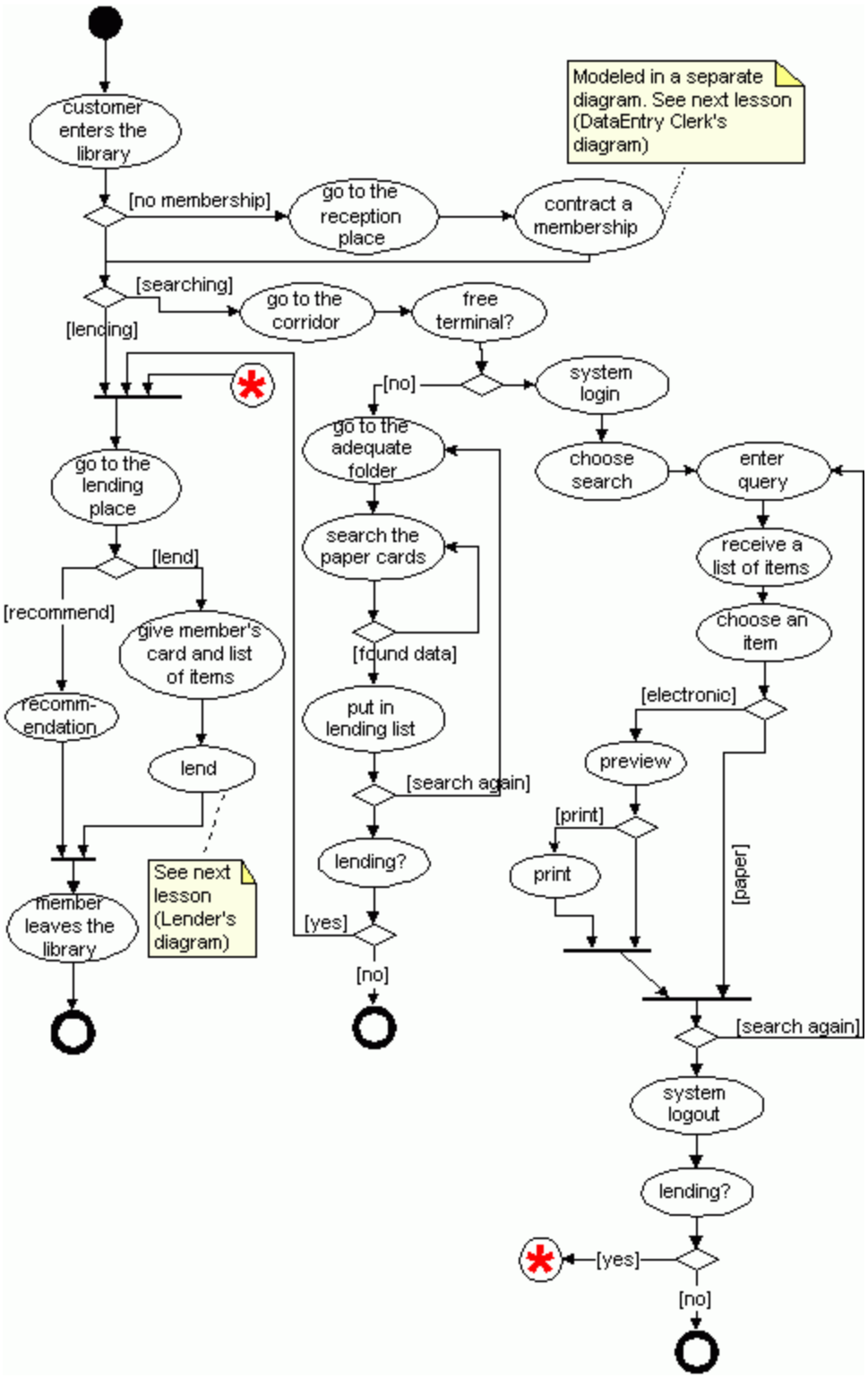
در متن زير ، حرف A براي تحليلگر و حرف C براي clien در نظر گرفته شده است.

A	بياييد كارمان را آغاز كنيم. بگوييد كه وقتي يک مشتري به كتابخانه شما وارد مي شود چه رخ مي دهد؟
C	به راهرويي كه جستجوي داده ها را در آنجا مي تواند انجام دهد مي رود. پس از آنكه كتاب يا مجله مورد نظر را براي امانت گرفتن انتخاب كرد، به محلي كه مي تواند آن كتاب يا مجله را به امانت بگيرد مي رود.
A	گفتيد جستجوي داده ها، چه نوع داده اي؟
C	براي هر كتاب، نويسنده يا مجله يک data record اصلي بر اساس حروف الفبا وجود دارد. هر مدرک بر اساس نوع محصول (كتاب يا مجله) شامل داده هاي مختلفي مي باشد. اين مدرک ها بر روي كارت هاي كاغذي نوشته شده اند و در پوشه هاي مختلفي قرار مي گيرند. دسترسي به اين كارت ها، از طريق كامپيوترهايي كه در راهرو يافت مي شوند، نيز امكان پذير مي باشند.
A	اين به آن معناست كه كتابخانه شما، كامپيوتري شده است. چرا از كامپيوتر و مدرک هاي كاغذي استفاده مي كنيد ؟
C	به دليل اينكه در دنياي تكنولوژي برتر زندگي مي كنيم، بايد از مراحل اين زمان پيروي كنيم.كامپيوتر ها تعداد زمان كمي است كه شناخته شده اند اما مدرک هاي كاغذي از زماني كه كتابخانه ها ساخته شده اند وجود دارند. همچنين برخي از مشتري ها كه صبور نيستند، در صورتي كه تمام كامپيوترها اشغال شده باشند، مي توانند از مدرک هاي كاغذي استفاده كنند.
A	آيا داده هاي يكساني در يک مدرک كاغذي و كامپيوتر وجود دار ؟
C	بله. تمام داده هاي قديمي از يک مدرک كاغذي به مدرک كامپيوتري منتقل شده اند و داده هاي جديدي كه در كارت هاي كاغذي نوشته ايم در فرمت الكترونيكي هم نوشته شده اند.
A	پايگاه داده هاي شما دركجا قرار دارند ؟
C	پايگاه داده هاي ما در كامپيوتر سرور قرار دارند كه اين سرور قلب شبكه محلي ما به حساب مي آيد- تمام پايانه ها در كتابخانه ما قرار دارند.
A	بياييد به داده ها بازگرديم. گفتيد كه در فرمت الكترونيكي و فرمت كاغذي، داده يكساني وجود دارد. در مكالمات بعدي از آنها مي توانيم با نام يكسانindex نام ببريم .

C	موافقم.
A	وقتي از index صحبت مي كنيم، چه نوعي از اطلاعات براي انجام يک جستجو ضروري است؟
C	داده هاي يک كتاب و يک مجله بسيار شبيه هم هستند. هر دوي آنها ISBN (شماره منحصر به فرد هر كتاب و مجله)، عنوان، سال انتشار، محل انتشار دارند، اما كتاب ها نويسنده و طبقه دارند و مجله ها جلد و ناحيه دارند. نتايج بدست آمده، نوع كتاب ها يا مجلات را مشخص ميكنند .
A	نوع كتاب و مجله يعني چه؟
C	به جز كتاب ها و مجلات كاغذي، كتابها و مجلات الكترونيكي هم در كتابخانه ما پشتيباني ميشوند.
A	كتابها و مجلات الكترونيكي. چگونه اين گونه موارد (كتابها/ مجلات) را اداره مي كنيد؟
C	سروري كه داريم براي ذخيره سازي اين گونه موارد استفاده مي شوند. ما برخي از موارد الكترونيكي را توليد مي كنيم، اما در پايگاه داده هايمان(Database) ، آدرس هاي اينترنتي كتابها و مجلات الكترونيكي كتابخانه هاي ديگري كه با هم كار مي كنيم نيز، وجود دارند.
A	شما حجم بالايي از داده ها داريد. آيا كسي را داريد كه وارد كردن داده و نگهداري آنها به او مربوط باشد؟
C	بله، كارمنداني براي انجام اين كار داريم.
A	گفتيد كارمندان. فكر مي كنم صحبت كردن راجع به آنها در بحث كنوني نمي گنجد، راجع به آنها بعداً صحبت خواهيم كرد. بحث مشتري ها هنوز به اتمام نرسيده است. پيش از آنكه به فرآيند امانت دادن بپردازيم، اجازه دهيد ببينيم تا اين نقطه از مصاحبه به چه نتايجي رسيده ايم: هر مشتري يک جستجوي داده براي كتاب ها و مجلات انجام مي دهد. فرمت الكترونيكي كتابها كه از جستجو بدست مي آيد، فوراً قابل دسترسي است. فرمت هاي كاغذي كتاب ها، امانت داده مي شوند البته در صورتي كه آن كتاب قبلاً امانت داده نشده باشند. حال ميتوانيم به سراغ فرآيند امانت گرفتن برويم.
C	در ابتدا بركاربرهايي كه در كتابخانه جستجوي داده را انجام داده اند، تاكيد مي كنم. امانت دادن براي كتاب هايي صورت مي گيرد كه فرمت كاغذي آن وجود داشته باشد و اين امر دربخش امانت دادن كتابخانه انجام مي گيرد. كتابها و مجلات جستجو شده در صورتي به مشتريان امانت داده مي شود كه در آن زمان، يک نسخه از آن در كتابخانه موجود باشد.
A	بياييد به هر دو حالت ممكن بپردازيم، اگر نمونه اي در كتابخانه وجود داشته باشد يا وجود نداشته باشد، يعني نزد فرد ديگري امانت باشد يا نباشد، در هر دو حالت چه رخ خواهد داد؟
C	وقتي كه آن نمونه موجود نباشد، مشتري مي تواند كتاب يا مجله مورد درخواست خود را رزرو

	کند، آنگاه پس از آنکه یک نسخه از آن نمونه به کتابخانه برگردانده شد، کتابخانه باید مشتري را مطلع سازد. آنگاه مشتري باید به کتابخانه برود و نمونه مورد نظر را امانت بگیرد، پس از آنکه کتابخانه به آن فرد در مورد آن نمونه اطلاع داد براي چند هفته کوتاه، نمونه را براي مشتري آزاد نگاه مي دارد.
A	اگر یک نمونه آزاد باشد چه رخ مي دهد ؟
C	در این حالت، کارمندي که کارش به امانت دادن مربوط میشود و به وي امانت دهنده مي گوییم باید داده هاي آن مشتري را بررسي کند، که آیا وي عضو کتابخانه مي باشد یا خیر. اگر مشتري عضو کتابخانه باشد، امانت دهنده، شماره ISBN کتاب یا مجلات امانت داده شده و داده هاي شخصي مشتري را در مدرک امانت گرفتن ثبت مي کند. همچنین یک تاریخ در این مدرک ثبت مي شود که در پایگاه داده ها ذخیره خواهد شد و هر مشتري تعداد محدودي کتاب و مجله را در زمان محدودي مي تواند به امانت بگیرد.
A	آیا به کارمندان هم اجازه امانت کتاب و مجلات را مي دهید؟
C	بله، هر کارمند در کتابخانه ما مدرک اي مشابه هر یک از اعضاء دارد، که به کارمندان هم از آن طریق کتاب امانت مي دهیم.
A	گفتید که مدرک هاي امانت گرفتن در پایگاه داده ها امانت گرفتن ذخیره مي شود، آیا این کار بر عهده امانت دهنده است یا شخص دیگري آن را انجام مي دهد؟
C	شبکه محلي ما تنها شامل پایانه هاي جستجو نیست، بلکه هر امانت دهنده نیز یک پایانه دارد که مي تواند از طریق آن داده هاي مشتري ها و آزاد بودن نمونه ها را بررسي کند. همچنین از این طریق به آساني مي توان فهمید که آیا مشتري کتابها را برگردانده یا خیر.
A	یک لحظه صبر کنید، بحث مان به مسیر دیگري کشیده شد، ما بحث راجع به وظایف کارمندان راشروع کرده ایم در حالیکه بحث راجع به مشتري ها را به اتمام نرسانده ایم. حال بگویید که یک نمونه آزاد از کتاب و مجله که در کتابخانه وجود دارد چگونه امانت گرفته مي شود و مدرک امانت گرفتن پس از آن چگونه ایجاد مي گردد ؟
C	دوباره تاکید مي کنم که تنها کتابهاي با فرمت کاغذي مي توانند امانت داده شوند زیرا نمونه اي از آنها وجود دارد. فرمت هاي الکترونیکي کتابها و مجلات ، نمونه ندارند.به عبارت دیگر کتاب هاي با فرمت الکترونیکي تعداد نامتناهي نمونه دارند و یا هر زماني، در کتابخانه موجود مي باشند .
A	آیا امکان دارد که هیچ نمونه اي از کتابها و مجلات در لیستي که مشتري جستجو مي کند موجود نباشد؟ آیا مشتري ها مي توانند از فرد امانت دهنده بخواهند که کتاب یا مجله اي را به آنها پیشنهاد کند ؟

C	البته. امانت دهنده مي تواند به تاريخچه امانت گرفته شدن كتابها و مجلاتي كه مشتري به امانت گرفته، نگاه مختصري بياندازد و اگر كتابها يا مجلاتي را در حيطه مورد علاقه مشتري خوانده است، برخي چيزها را به مشتري پيشنهاد كند و مشتري مي تواند آن پيشنهاد را بپذيرد يا رد كند.
A	اين تمام آن چيزي است كه تا اكنون بايد ميدانستيم. فكر مي كنم رسم نمودار كاربر انجام گرفته است و مي توانيم نگاهي به آن بياندازيم:



نمودار فعالیت اعضا مطابق با مصاحبه قبلي

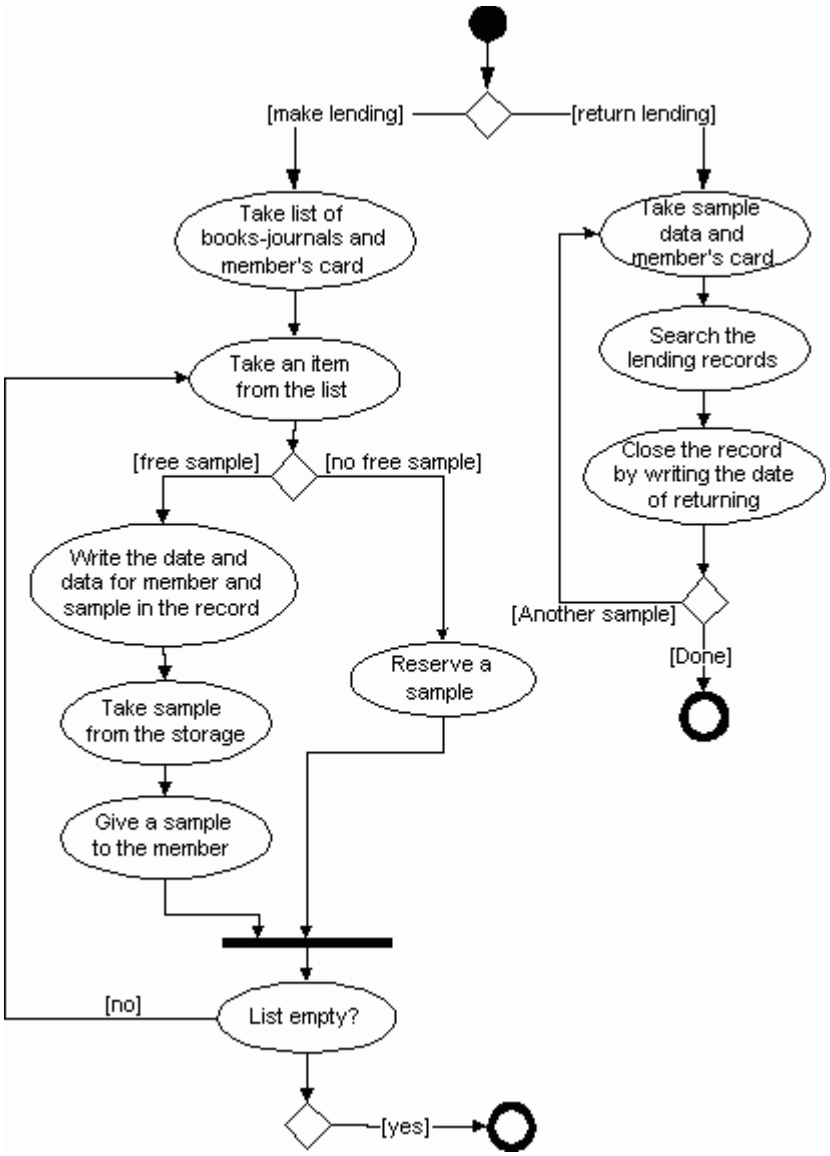
يافتن فرآيندهاي كسب و كار ٢

اين درس تعميمي از درس گذشته مي باشد. حال مصاحبه اي با client و توضيحات مربوط به كار كارمندان در كتابخانه ارائه مي دهيم. بياييد به تمام جزئيات داده شده و تحليل توليد نمودار فعاليت، نگاهي بيندازيم:

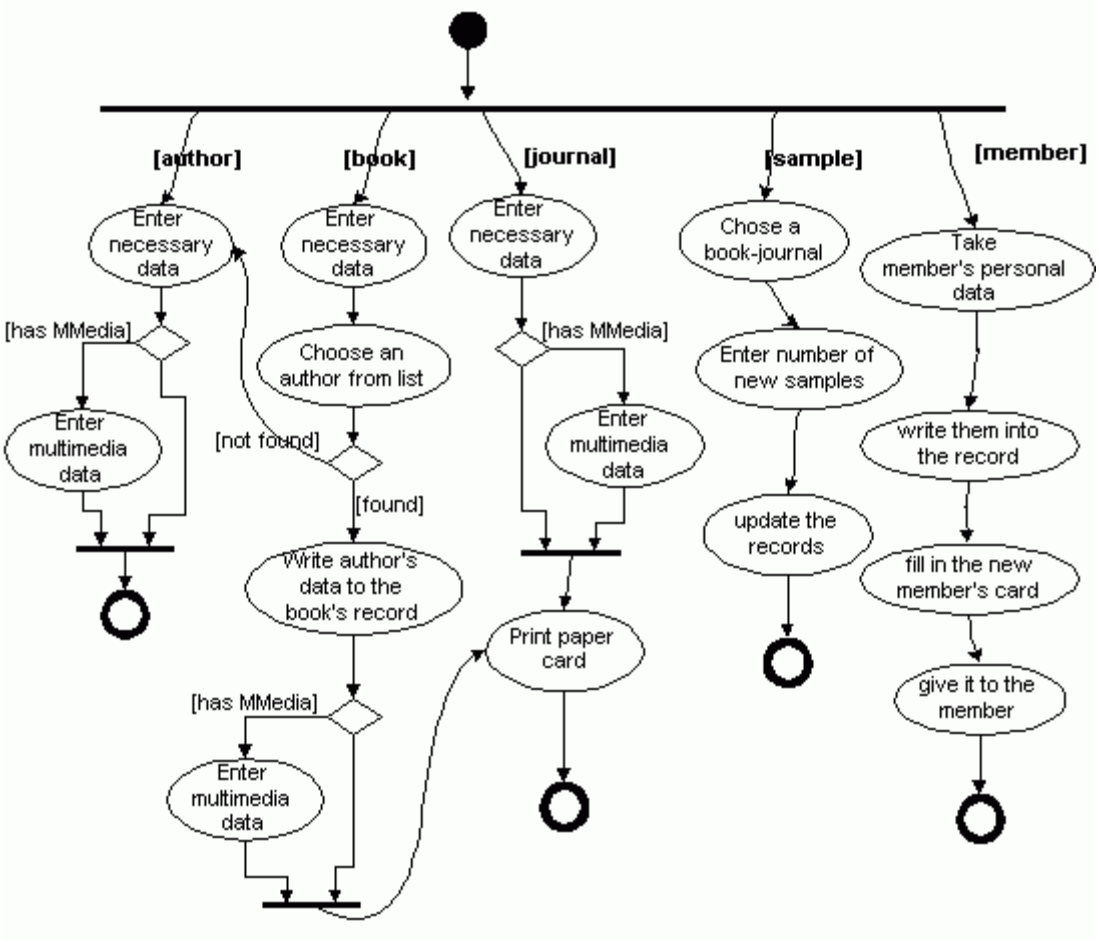
A	در مصاحبه قبلي راجع به مشتري ها و همه فعاليت هاي مربوط به آنها مانند امانت گرفتن، رزرو كردن، پيشنهادات و جستجو صحبت كرديم. در فرآيند امانت گرفتن و پيشنهادات، روابط متقابل كارمندان معرفي شد. حال بياييد دوباره به بحث راجع به كارمندان برگرديم و درباره كارمنداني كه با client ها، كتابها و مجلات در ارتباط هستند صحبت كنيم.
C	بسيار خوب. كارمندان در مصاحبه قبلي معرفي شدند و ديديم كه كارشان در ارتباط با امانت دادن و پس گرفتن كتابها و مجلات و فرستادن گزارش به اعضايي كه بايد كتابها را برگردانند و اعضايي كه كتابها را رزرو كرده اند و نمونه آنها دركتابخانه موجود است، مي باشد.
A	آيا اينها تمام وظايف فرد امانت دهنده مي باشد؟
C	بله.
A	وقتي راجع به مشتري صحبت مي كنيم، به حجم بسيار زيادي از داده ها اشاره مي شود، برخي كارمندان براي اين امر استخدام شده اند، مي توانيد بيشتر توضيح دهيد؟
C	پس از خريد كامپيوترها، تعدادي از كارمندان قديمي كه به فرآيند نوشتن كارت هاي كاغذي مشغول بودند، به كار كردن با كامپيوتر مي پردازند اين كارمندان كه مسئول وارد كردن داده هستند، حال با برخي از پايانه ها براي وارد كردن داده و نگهداري آنها كار مي كنند. همچنين براي نگهداري database (پايگاه داده) متخصصين database، استخدام شده اند .
A	اگر من بخواهم عضو كتابخانه شما بشوم، چه كارهايي بايد انجام بدهم؟
C	در ابتدا هر عضو جديد بايد به محل پذيرش برود تا به عضويت درآيد. كارمندان كه عضويت جديد را وارد مي كنند، مدرک جديدي مشتمل بر نام و نام خانوادگي و آدرس و محل سكونت عضو جديد، براي وي دايركارت ميكنند و يک شماره ثبتِ نامِ شخصي به آن اضافه مي كنند. همچنين شماره تلفن و آدرس email و تاريخ تولد نيز در اين مدرک ثبت خواهد شد. پس از ثبت نام، يک كارت عضويت به فرد داده مي شود و يک مدرک كاغذي براي مستند سازي كتابخانه از آن چاپ ميشود.
A	چگونه داده كتابها و مجلات را وارد مي كنيد ؟
C	كتابها و مجلات معمولاً صفات زيادي دارند كه شامل ISBN، عنوان، سال انتشار و محل انتشار

	ميشود. همچنين هر مورد تعدادي نمونه دارد. برخي مواقع يک داده مولتي مديا نيز توسط يک مورد پشتيباني مي شود. اگر اين مورد يک تبليغ در فرم کاغذي باشد ما آنرا به شکل الکترونيکي تبديل ميکنيم و در database قرار ميدهيم.
A	آيا داده مولتي مديا را فقط براي موارد الکترونيکي پشتيباني مي کنيد يا داده هاي کاغذي هم مولتي مديا دارند؟
C	شکل کتابها و مجلات اهميت ندارد، براي هر دو فرمت کاغذي و الکترونيکي، مولتي مديا را پشتيباني مي کنيم. کتاب هاي محبوب دنيا، در صورتي که مولتي مديا از آنها پشتيباني کند، محبوب تر خواهند شد. هر کاربري کنجکاو است که با اين نوع از اطلاعات برخورد داشته باشد.
A	اگر من در داده ها جستجو انجام دهم، مي توانم پشتيباني داده مولتي مديا را براي آن مورد مرور کنم؟ و چه نوعي از مولتي مديا پشتيباني ميشود؟
C	بله. در طول جستجوي کامپيوتري در نتايج بازگشتي اگر يک مورد يک داده مولتي مديا داشته باشد، کاربرها مي توانند آن داده را مرور کنند. نوع مولتي مديايي که پشتيباني مي شوند شامل داده، صوت، تصوير و عکس مي باشد.
A	چگونه يک داده را براي نويسنده کتاب وارد مي کنيد؟
C	متاسفم، فراموش کرده بودم که نويسنده هايي که به طور مجزا داده هايشان وارد مي شود را توضيح بدهم. براي نويسنده ها، نام و نام خانوادگي، سال و محل تولد وارد مي شود. در طول فرآيند وارد کردن داده کتاب، اگر نويسنده کتاب در ليست نويسنده ها وارد شده باشد، آنگاه از آن ليست انتخاب مي شود. در جاي ديگري هم فرآيند ايجاد داده نويسنده را استفاده مي کنيم براي کتابهايي که طبقه آنها وارد شده است.
A	در مورد مجلات چطور ؟
C	براي مجلات جلد و ناحيه وارد مي شود.
A	گفتيد که مدارک موازي هم توليد مي شوند (کاغذي و الکترونيکي) در کارت هاي کاغذي مولتي مديا را چگونه نشان مي دهيد ؟
C	داده مولتي مديا تنها در فرمت الکترونيکي وارد مي شود. من گفتم که کارت هاي کاغذي از کارت هاي الکترونيکيprint گرفته مي شوند ، اما فقط اطلاعات ضروري print گرفته مي شوند.
A	ما کار افراد امانت دهنده و کارمندهايي که داده هاي کتابها، نويسنده ها، مجلات و اعضا را وارد مي کنند را بررسي نموديم. اما هيچ داده نمونه اي نمي بينم. در فرآيند امانت دادن، گفتيد که يک نمونه آزاد، مي تواند امانت داده شود. داده هاي نمونه را چگونه نمايش مي دهيد ؟
C	زماني را که راجع به وارد کردن کتاب صحبت مي کرديم يادآوري مي کنم، فرآيند وارد کردن

	کتاب، شامل وارد کردن شماره نمونه جديد هم مي شود اما اين امر هنگامي امكان پذير مي باشد كه كتاب به كتابخانه برسد و يک مدرک براي آن در database وجود داشته باشد. در آن حالت تنها داده نمونه وارد مي شود و اين داده شامل امضاي نمونه مي شود كه تنها به امانت گيرنده مربوط مي شود (اطلاعات مربوط به محل ذخيره نمونه در اتاق ذخيره)
A	آيا فكر مي كنيد كه تجزيه و تحليل مربوط به كارمندان را به پايان رسانده ايم ؟
C	البته، به انتهاي بحث رسيده ايم و اميدوارم كه توضيحات ام در مورد تمام مراحل كار كتابخانه، تصوير روشني از كتابخانه هاي مشابه كتابخانه ما، به شما داده باشد .



نمودار فعاليت براي فرد امانت دهنده كه از مصاحبه بالا بدست آمده است



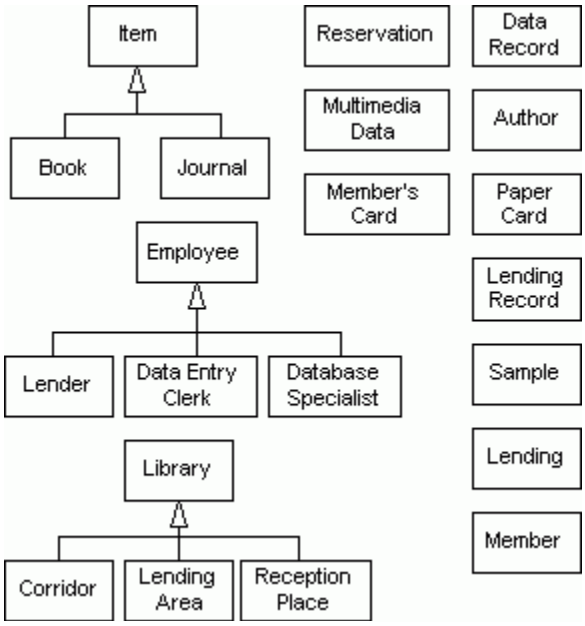
نمودار فعالیت کارمند وارد کننده داده ، که از مصاحبه بالا استخراج شده است

تحليل قلمرو Domain

قبل از مصاحبه هاي اضافي، ابتدا تيم توسعه دهنده بايد بر روي مفهوم مصاحبه فرآيند كسب و كار، كار كنند و اين كار يک مدلساز اشياء ميباشد و هدف توليد نمودار كلاس ابتدايي است.
مدلساز اشياء، اسم ها، فعل ها و عبارات فعلي را جستجو مي كند، برخي از اسم ها در مدل، كلاسها و برخي صفات خواهند شد. فعل و عبارات فعلي مي توانند عمليات يا برچسب وابستگيها باشند.
بياييد خود با قوانين مدلساز هماهنگ كنيم و توسعه نمودار كلاس را شروع كنيم. از مصاحبه client، ميتوانيم اسامي زير را استخراج كنيم :
راهرو، data record، مدرک ذخيره سازي، شبكه محلي، كامپيوتر سرور، پايانه كامپيوتر، كتاب كاغذي، مجله كاغذي،كتاب الكترونيكي، مجله الكترونيكي، مورد، داده، محدود زمان، تعداد نسخ، كتاب، مجله، نمونه، نويسنده، كارت كاغذي،كتابخانه، database، ISBN، عنوان، سال انتشار، طبقه، محل انتشار، جلد، ناحيه، رزرو، كارمند، امانت گرفتن، محل امانت گرفتن، امانت گيرنده، كاربر، عضو، مدرک امانت گرفتن، گزارش، تعداد نمونه ها، امانت دهنده، متخصص database، محل پذيرش، نام، نام خانوادگي، آدرس، محل زندگي ، شماره ثبت نام شخصي، تلفن،email، تاريخ تولد ، كارت عضويت، مستندسازي كتابخانه، داده هاي مولتي مديا، اطلاعات، متن، صوت، تصوير، عكس، امضا .

و فعل ها عبارتند از :
رفتن، جستجو كردن، امانت گرفتن، بردن، مرتب كردن، چاپ كردن، قرار دادن، انجام دادن، داشتن، تبديل كردن، نوشتن، اداره كردن، دسترسي داشتن، خواندن، ازقبل ديدن، استفاده كردن از نمونه، رزرو كردن، دادن، آگاهي دادن، درست كردن، آزاد كردن يک نمونه ، بررسي كردن داده، پيشنهاد كردن، پست كردن، خريدن، عمل كردن، يادآوري كردن، درگير بودن، مرور كردن، وارد كردن .

حال سعي مي كنيم برخي از اسم هاي معني دار را گروه بندي كنيم:
يک گروه شامل افراد مي شود كه عبارتند از: اعضا، نويسنده،كارمند، امانت دهنده، متخصص database،كارمند امانت دهنده. اين گروه مي تواند شامل يک بخش فرعي نيز باشد چون تمام افراد به غير از اعضا و نويسنده، كارمند هستند .
گروه ديگر شامل اقلام كتابخانه است مانند: كتابها ، مجلات، نمونه ها، رزرو، و امانت گرفتن.
گروه سوم شامل محوطه كتابخانه مي باشد: راهرو ، محل امانت گرفتن، محل پذيرش.
در گروه چهارم اين اسم ها قرار مي گيرد: كارت اعضا، data record، كارت كاغذي ، داده مولتي مديا، مدرک امانت گرفتن.



كلاسهاي مجرد نمودار كلاس را به گروههاي معنادارافراز كرده اند

نمودار كلاس اوليه به صورت فوق مي باشد، در بخش بعدي وابستگي بين كلاس ها را پيدا خواهيم كرد و نمودار كلاس توصيفي تري نسبت به نمودار بالا توليد مي كنيم.

شكل دهي Association ها

حال ما association هاي بين برخي كلاسها را ايجاد و برچسب گذاري مي كنيم. فعل ها و عبارات فعلي ميتوانند در برچسب گذاري ما را ياري كنند اما نبايد خود را به اين فعل ها و عبارات فعلي مصاحبه محدود كنيم.

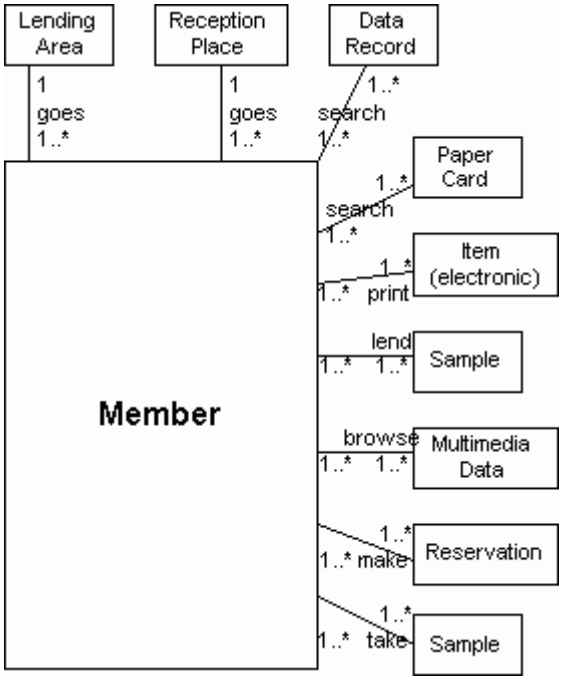
برچسب هايي كه تا حدي توصيفي تر هستند مي توانند خودشان را پيشنهاد كنند .

بياييد اول با اعضا شروع كنيم و **association ها** را با توليد عباراتي **association** را توصيف ميكنند، برچسب گذاري كنيم. در اينجا عباراتي وجود دارند كه فوراً به ذهن هر كسي خطور مي كنند:

- فردي كه عضو است كارتهاي كاغذي يا data record را جستجو مي كند.
- فردي كه عضو است به محل امانت گرفتن مي رود.
- فردي كه عضو است يک نمونه را امانت مي گيرد.
- فردي كه عضو است يک رزرو انجام ميدهد.
- فردي كه عضو است نمونه را مي برد.
- فردي كه عضو است داده مولتي مديا را مرور مي كند.
- فردي كه عضو است به محل پذيرش مي رود.
- فردي كه عضو است موارد الكترونيكي را print ميگيرد.

وقتي association ها را برچسب مي زنيم، مي توانيم چندتايي(multiplicities) آنها را در خطوط association ها قرار دهيم.

بنابراين كلاس مشتري مشابه زير خواهد بود:

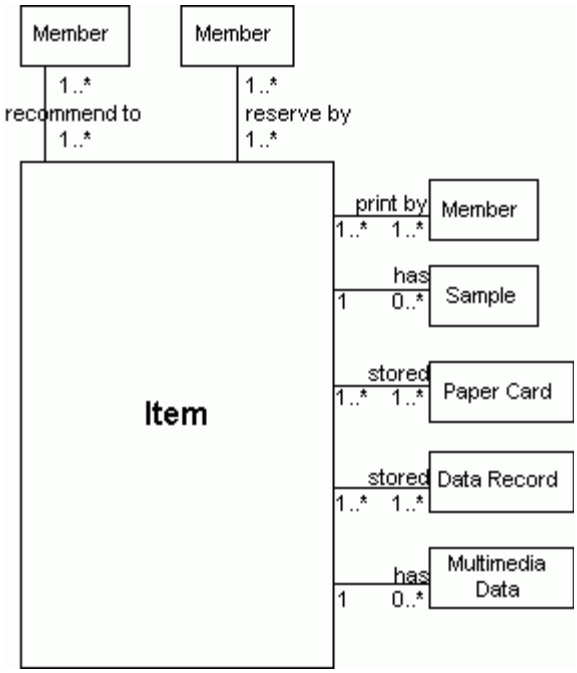


كلاس اعضا با چند تايي در association ها

Association ها براي موارد يک كلاس(در اينجا موارد به جاي كتاب و مجله است)، مي توانند از ميان عبارات ليست زير باشد:

- موردي كه مي تواند به فرد عضو پيشنهاد شود.
- موردي كه توسط يک عضو رزرو مي شود.
- موردي كه توسط يک عضوprint مي شود.
- موردي كه بر حسب حروف الفبا در كارتهاي كاغذي و data record مرتب شده است.
- موردي كه نمونهاي دارد.
- موردي كه مولتي مديا دارد.

پس از تكميل نام ها و تعدد وابستگيها، كلاس مورد به صورت زير خواهد بود:

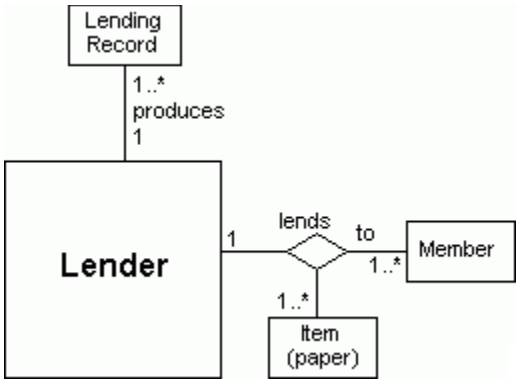


Association ها با موارد(کتابها و مجلات)

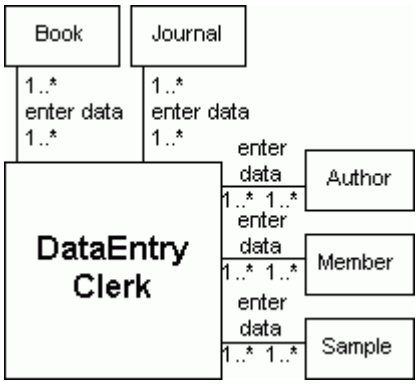
روند شکل دهيAssociation ها و چند تايي براي کلاس کارمندان جالب توجه است. به ياد آورديد که کارمندان هم از اعضاي کتابخانه هستند، بنابراين تمام Association ها کلاس اعضا، در کلاس کارمندان هم قرار ميگيرد. اما برخي از کارمندان کارهاي ديگري نيز انجام مي دهند که بقيه اعضا مجاز به انجام آنها نيستند، مانند:

- امانت دهنده، مدرک امانت گرفتن ايجاد مي کند.
- امانت دهنده موردي را به اعضا، امانت مي دهد.
- متصدي وارد کننده داده، داده هاي کتابها را وارد مي کند.
- متصدي وارد کننده داده، داده هاي مجلات را وارد مي کند.
- متصدي وارد کننده داده، داده هاي نويسنده ها را وارد مي کند.
- متصدي وارد کننده داده، داده هاي اعضا را وارد مي کند.
- متصدي وارد کننده داده، داده هاي نمونه ها را وارد مي کند.

پس از تکميل نام Association ها وچند تاييها، کلاس متصدي وارد کننده داده و امانت دهنده بصورت زير خواهند بود:

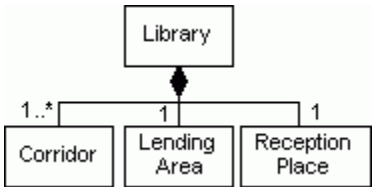


Association ها با امانت



Association ها با متصدي وارد كننده داده

ما كلاسهاي مجرد و Association ها را شكل داده ايم و ساير ابعاد سازماندهي باقيمانده اند. مرحله بعد تكميل كلاسهايي است كه جزء كلاسهاي ديگر ميباشند. يک كتابخانه بعنوان نمونه شامل يكسري بخشها است كه جالب توجه آنها چنين است: راهرو، بخش امانات و پذيرش. تصوير بعدي اين تركيب را نشان ميدهد.



تركيب در حيطه كتابخانه الكترونيک

اگر در درک روند شكل دهي Association ها و تعدد و تركيب مشكلي داريد، به درسي كه در روز دوم خوانده ايد، مراجعه نماييد:وابستگيها و عموميت دادن.

تكميل كلاس ها

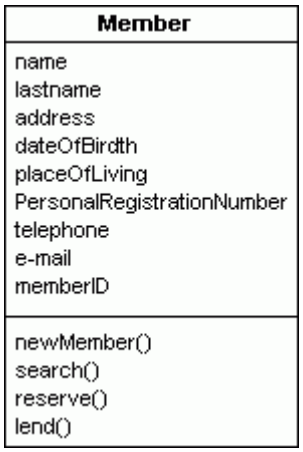
حال مي توانيم با افزودن برخي صفات و عمليات، به تصحيح كلاس ها بپردازيم.كلاس هاي مهم در اعضا، نويسنده، كارمند، كتاب، مجله و نمونه ظاهر مي شوند.

عضو

در ليست اسم برخي از اسم ها، به عنوان صفت عض، براي استفاده مناسب هستند و از ليست افعال ميتوانيم برخي عمليات مناسب را انتخاب كنيم. اين دو ليست در جدول زير داده شده است :

ليست صفات	ليست عمليات
نام	عضو جديد()
نام خانوادگي	
آدرس	جستجو()
تاريخ تولد	
محل سكونت	امانت گرفتن()
شماره رجيستري شخصي	
تلفن	رزرو كردن()
email	
ID عضو	

حال، كلاس عضو به صورت زير خواهد بود:



كلاس عضو

نویسنده

لیست عملیات و صفات نویسنده به صورت زیر خواهد بود:

لیست عملیات	لیست صفات
ایجاد نویسنده()	نام
مرجع کتاب()	نام خانوادگي
	تاریخ تولد

و کلاس نویسنده به صورت زیر است:

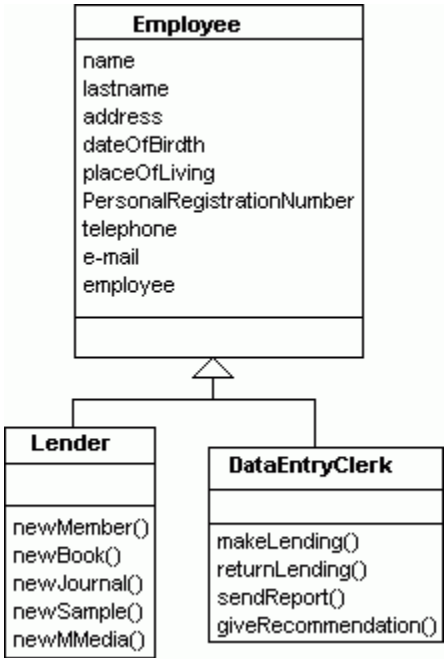
کلاس نویسنده

کارمند

فرد امانت دهنده و کارمند وارد کننده داده، همانند فرزندان کلاس کارمندهاي مجزا مي باشند. در ابتدا صفات کلاس کارمند را ذکر مي کنيم، آنگاه کلاس هاي فرزندان اين کلاس، اين صفات را از آن به ارث مي برند. همچنين بايد به خاطر داشته باشيم که هر کارمند يک مدرک اعضاء دارد، اين امر نشان مي دهد که در کلاس کارمند، ليست صفات اعضا به اضافه صفاتي که تفاوت کارمندان را از بقيه اعضا نشان دهد، نيز ظاهر خواهد شد. هچنين کلاس مربوط به کارمند وارد کننده داده، صفات جديدي خواهد داشت که نوع داده وارد شده اي که آن کارمند اجرا کرده است را معلوم مي کند. ليست صفات وعمليات کارمند وارد کننده داده و امانت دهنده در جدول زير داده شده است:

لیست صفات کلاس کارمند	لیست عملیات کلاس امانت دهنده	لیست عملیات کلاس کارمند وارد کننده داده
نام	امانت دادن()	عضو جديد
نام خانوادگي		
آدرس		کتاب جديد()
تاريخ تولد	برگشت نمونه امانت داده شده()	
محل سكونت		مجله جديد()
شماره ثبت نام شخصي		
شماره تلفن	گزارش فرستادن()	نمونه جديد()
email		
عضوID		مولتي مدياي جديد ()
كارمند		

ساختار کلاس کارمند، همراه با زیر کلاس هایش در شکل زیر نشان داده شده است :



کلاس کارمند و فرزندانش

سعي كنيد خصلت ها وعملكردهاي كلاس كتاب را پيدا كنيد ، اين كلاس بسيار شبيه كلاس مجله كه در زير به بررسي آن خواهيم پرداخت مي باشد .

مجله

ليست صفات و عمليات كلاس مجله در جدول زير داده شده است:

ليست عمليات	ليست صفات
نمونه هاي آزاد ()	عنوان
	جلد
مجله جديد ()	ناحيه
	سال انتشار
امانت دادن ()	محل انتشار
	ISBN
رزور كردن ()	وضعيت
	تعداد نمونه ها

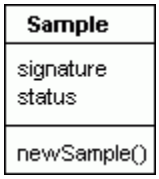
صفات وضعيت، نوع مجله يعني الكترونيكي يا كاغذي بودن آن را مشخص مي كند. كلاس مجله به صورت زير خواهد بود.

كلاس مجله

نمونه

فرآيند امانت دادن در صورتي منجر به امانت دادن كتاب يا مجله مي شود كه يک نمونه از آن به صورت آزاد دركتابخانه موجود باشد. براي اين منظور، يک كلاس نمونه بايد ارائه شود.كلاس نمونه تنها شامل دوصفت خواهد بود: امضا و وضعيت كه مربوط به امانت گرفتن و يا آزاد بودن مي باشند و تنها يک عملكرد موجود مي باشد كه عبارت از نمونه جديد () مي باشد.

كلاس نمونه در شكل زير ارائه شده است .



کلاس نمونه

وقتي نتايج مصاحبه، فرآيند هاي كسب وكار و تجزيه و تحليل هاي domain را كنار هم قرار ميدهيد، يک واژه نامه مدل را داريد

. اين واژه نامه اي از تمام واژگان يک مدل است. به شما در مراقبت از ثبات و جلوگيري از ابهام كمک مي كند. همچنين فكر جالبي نيست كه همه جزئيات مدل كلاس خود را به صورت يک نمودار داشته باشيد بنابراين به يک نمودار اصلي كه همه ارتباطات، وابستگيها وعموميت ها را نشان دهد، نياز پيدا خواهيد كرد اما بهترين كار اين است كه صفات و عمليات را از آخرشكل حذف كنيد. شما ميتوانيد با با قرار دادن كلاسهاي انتخاب شده در نمودارهاي جداگانه آنها را در ديد قرار دهيد.

توسعه بينش

اكنون زمان آن است كه تيم بر روي backbone فني كتابخانه ديجيتالي كار كند. فرآيندهاي كسب و كار و نمودارهاي كلاس بدست آورده اند. آيا گروه مي تواند كار را با كدنويسي شروع كند يا نه؟ هنوز نميتوانند نوشتن برنامه را انجام دهند، در ابتدا بايد بينش سيستم را توسعه دهند.
تيم از دانش فرآيندهاي كسب وكار خود و دانش بدست آورده در مورد قلمرو استفاده خواهد كرد تا بفهمد كه القاء تكنولوژي كجا كار كتابخانه را بهبود ميبخشد. مجريان اين مكالمه، يک تحليلگر، مدلساز،كتابدار، امانت دهنده، كارمند وارد كننده داده، يک فرد عضو، و مهندس سيستم مي باشند.
يک مسوول جلسات، ملاقات را ترتيب ميدهد. او، نسخه هايي از نمودار هاي فرآيند كسب وكار را كه در درسهاي پيشين تعميم يافتند توزيع مي كند.
بخشهاي يافتن فرآيندهاي كسب و كار ۱ و يافتن فرآيندهاي كسب و كار ۲ را ببينيد.
اين مكالمه درباره تغيير اطلاعات مي باشد برخي فرآيندها در كسب و كار كتابخانه به تغيير اطلاعات بستگي دارند. اگر بتوانيم به اين تغييرات سرعت ببخشيم، به هدف خود خواهيم رسيد. از اين مكالمه اين نتيجه به دست آمده است كه، تغيير اطلاعات زماني رخ مي دهد كه:

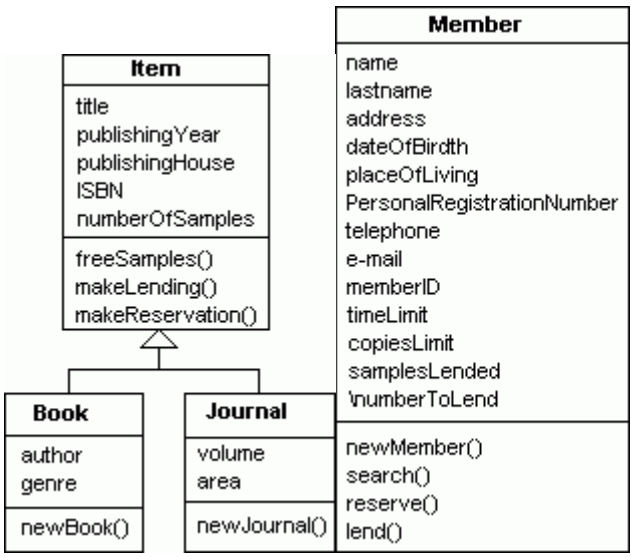
- فرد عضو كتابخانه، يک نمونه كتاب را به امانت بگيرد.
- فرد عضو كتابخانه، يک نمونه مجله را به امانت بگيرد.
- فرد امانت دهنده، يک كتاب را به فرد عضو كتابخانه ، پيشنهاد كند.
- فرد امانت دهنده، يک مجله را به فرد عضو كتابخانه ، پيشنهاد كند.
- فرد عضو كتابخانه، عضويت خود را با كارمند وارد كننده داده ، منعقد كند.

در زير بخشي از اين مكالمه آورده شده است. در اينجا، مدلساز ايده هاي جالبي درباره افزودن صفات جديد به نمودار كلاس دارد. اجازه دهيد ببينيم كه چه مواردي پيش آمده است:

تحليلگر	بياييد به سراغ فرآيند امانت دادن برويم. قبلاً گفته شده بود كه اعضا زمان محدودي براي امانت گرفتن، فرصت دارند و تعداد محدودي نمونه را مي توانند به امانت ببرند. امانت دهنده ها چطور مي فهمند كه چند نمونه را بايد به فرد عضو به امانت بدهند ؟
امانت دهنده	در ابتدا به كارت مدرک امانت گرفتن عضو نگاهي مي اندازم و تعداد مورد هايي را كه آن فرد به امانت گرفته است را محاسبه مي كنم، همچنين تاريخ امانت گرفتن را بررسي مي كنم تا بفهمم آيا از زمان امانت كتاب يا مجله اي كه به امانت گرفته شده، زماني اضافه تر بر زمان تعيين شده وجود دارد يا نه ؟
مدلساز	همزمان با اينكه به صحبت هاي شما گوش ميدهم، روي نمودار هاي كلاس هم كار

	مي كنم و حالا چند سوال دارم: در كلاس فرد عضو، ممكن است صفات جديدي مانند محدوده زماني و تعداد نسخ كه براي هر عضو مشترک است نيز اضافه كنيم همچنين صفتي كه تعداد نمونه هايي كه قبلاً امانت گرفته شده اند و نمونه هاي امانت گرفته اي كه بعداً معرفي خواهند شد را در بر مي گيرد و صفتي كه از صفات بالا ناشي مي شود و تعداد نمونه براي امانت دادن را مشخص مي كند كه بين تعداد نمونه هاي محدود براي امانت گرفتن و تعداد نمونه هاي امانت گرفته شده متغير اس .
امانت دهنده	ايده قشنگي است و از اين طريق تعداد نمونه هايي راكه مجاز هستم به فرد عضو امانت دهم،خواهم دانست.
تحليلگر	در مورد زمان محدود چطور؟
مدلساز	صفتي داريم كه زمان امانت گرفتن را در بر مي گيرد. اگر محدوده زماني را به آن اضافه كنيم يک نتيجه بدست خواهد آمد، اگر نتيجه ازتاريخ جاري بزرگتر باشد، در مي يابيم كه آن عضو براي برگرداندن كتاب يا مجله امانت گرفته شده تاخير داشته است.
تحليلگر	در مورد داده مولتي مديا چطور؟ گفته شده كه برخي از كتابها، مجلات يا نويسنده ها از داده مولتي مديا پشتيباني مي شوند.
مدلساز	اكنون زمان بسيار مناسبي است كه يک كلاس جديد مولتي مديا، در نمودار كلاس خود اضافه كنيم. براي اين منظوركلاس مولتي مديا شامل دو صفت خواهد بود، يكي نوع (متن ، صوت ، تصوير يا عكس) وديگري فايل (محل فيزيكي فايل مولتي مديا) و شامل يک عملكرد مي باشد: مولتي مدياي جديد ()
مسوول جلسات	تحليلگر، آيا اين توضيحات شما را راضي كرده است؟
تحليلگر	بله، حالا همه چيز برايم واضح تر شده است، هر قسمت اين پازل در جاي خود قرار گرفته است.
مدلساز	يک امكان ديگر هم وجود دارد. به كلاس هاي مجله و كتاب با دقت نگاه كنيد، هر دوي آنها صفات مشتركي مانند ISBN، عنوان، سال انتشار، محل انتشار و تعداد نمونه، و صفات متفاوتي مانند داده نويسنده و طبقه براي كتاب و جلد و ناحيه براي مجله دارند و اين موقعيت مناسبي براي يک وراثت جديد مي باشد. صفات مشترک مي توانند در كلاس جديد خانواده قرار گيرند كه اگر اجازه دهيد با نام مورد از آن نام ببريم. همچنين در اين كلاس جديد برخي عمليات ميتوانند رخ دهند، مانند: نمونه هاي آزاد () ، امانت دادن () و رزرو كردن (). از اين طريق مي توانيم در database

	خود، فضاي حافظه را ذخيره كنيم.
مسوول جلسات	خيلي خوشحالم كه بهينه سازي هايي انجام داده ايم. پيش از آن كه به مرحله بعدي برويم، مي توانم پيشنهاد كنم كه به كلاس هاي جديد نگاهي بيندازيم ؟
	همه موافق هستند.



تغييرات كلاس عضو. كلاس مولتي مدياي جديد و سلسله مراتب بين كتاب ها و مجلات، كلاس والد همان كلاس جديد است- كلاس مورد

تحليلگر	در طول فرآيند رزرو كردن يا فرآيند امانت گرفتن، تاريخ جاري بايد در جايي ذخيره شود. در نمودار كلاس خود، جايي براي ذخيره اين داده ها نمي بينم، لطفاً برايم توضيح دهيد كه اطلاعات مربوط به تاريخ امانت گرفتن و تاريخ رزو كردن را كجا ذخيره مي كنيد؟
مدلساز	بر اساس associationهاي چند به چند(many-to-many) بين كلاس نمونه و كلاس عضو، به ما پيشنهاد شده كه يک كلاس جديد براي associationهاي كه داده هاي مشترک وابستگيهاي را در بر مي گيرند، اضافه كنيم. دقيقاً درهمين محل است كه ميتوانيم دو كلاس جديد اضافه كنيم : كلاس رزرو كردن و كلاس امانت گرفتن و در اينجاست كه تاريخ رزرو كردن و تاريخ امانت گرفتن جاي مي گيرند. همچنين در كلاس رزرو كردن، يک صفت براي محدوده زمان رزرو، مي توان اضافه كرد كه براي محاسبه دوره زماني اي كه رزرو براي يک عضو نگه داشته مي شود،

	ضروري است. اين كلاس هاي جديد در شكل هاي زير نشان داده شده اند :

كلاسهاي جديدي كه در نمودار كلاس ما اضافه شده اند كلاس رزرو و امانت دادن هستند

تحليلگر	بسيار خوب. فكر مي كنم تا اين لحظه، همه مشكلات مربوط به كلاس ها را حل كرده ايم. آيا حالا ميتوانيم به بررسي ايده هايي درمورد اينكه سيستم به طور مشخص چه كاري بايد انجام دهد بپردازيم؟
مسوول جلسات	البته. كسي ايده اي ندارد؟
فرد عضو	من فكر مي كنم كه اين سيستم خيلي پربار خواهد بود اگركه ما، يعني اعضا، بتوانيم از خانه خود به همه داده هاي كتابخانه دسترسي پيدا كنيم. بعضي وقتها كه به كتابخانه مي روم، بدون اينكه واقعاً قصد امانت گرفتن كتابي را داشته باشم، اين كلمات را از فرد امانت دهنده مي شنوم كه :" متاسفم ولي هيچ نمونه آزاد براي كتابي شما درخواست كرده ايد نداريم. مي توانم كتاب ديگري بهتان پيشنهاد كنم؟ " در چنين مواردي به خود مي گويم :" سرآشپزم مرا خواهد كشت. وقت زيادي در اينجا تلف كرده ام و كارم را به اتمام نرسانده ام و اميدوارم كه رستوران كمتر از هميشه شلوغ باشد .
تحليلگر	بله، سيستم ما بايد به نوعي از رفت و آمدهاي بسيار اعضا به كتابخانه جلوگيري كند. البته هنگامي اعضا از خانه خود موردي را رزور ميكنند ويک نمونه آزاد براي آنها نگه داشته مي شود، بايد به كتابخانه مراجع كنند. همچنين فرآيند امانت گرفتن در صورتي انجام مي گيرد كه فرد عضو در كتابخانه حضور پيدا كند.
مهندس سيستم	مي توانيم از طريق شبكه اينترنت جهاني، به اعضا اجازه دهيم كه از خانه يا محل كار خود به كتابخانه LAN دسترسي داشته باشند، از اين طريق اطلاعات بسيار سريع منتقل خواهد شد وهمچنين با اطمينان مي توانم بگويم كه سيستم ما با LAN كتابخانه همياري خواهد كرد اما نرم افزار جديد هرجايي اجرا خواهد شد.
تحليلگر	سيستمي كه شما در مورد آن صحبت كرديد، تعداد زيادي از پيامدها را حل خواهد

	کرد. مانند جستجوي داده از خانه و ايجاد رزرو براي کتاب يا مجله از منزل يا محل کار و اين احساس را خواهم داشت که در کتابخانه هستم و از شبکه محلي استفاده ميکنم و ديگر نگران آزاد بودن يک پايانه نيستم.
فرد عضو	خيلي خوب است. وقتي کار زيادي براي انجام دادن دارم، مي توانم کامپيوترم را روشن کنم و به کتابخانه متصل شوم و براي اينکه بفهمم يک نمونه آزاد ازکتاب يا مجله اي که مي خواهم امانت بگيرم، وجود دارد يا نه، داده هاي مورد نياز را جستجو کنم. اما نمي توانم از طريق شبکه کتابخانه، کتاب يا مجله اي کار رزور را انجام دهم، چگونه مي توانم اين کار را از خانه يا محل کارم انجام دهم؟
مهندس سيستم	قصد داريم يک رابط وب پياده سازي کنيم که به شما امکان مي دهد که به داده هاي ذخيره شده در database دسترسي پيدا کنيد واين امکان را به شما مي دهد که از خانه يا محل کار خود يک کار رزرو را انجام دهيد. به اين ترتيب که با کليک کردن بر روي دکمه اي، پيامي از طرف کامپيوترشما، با داده هاي شخصي شما و داده کتابي که مي خواهيد آنرا رزور نماييد، فرستاده مي شود. اين پيام يک مدرک رزرو جديد در database ما باز خواهد کرد که آن داده ها را تا زمانيکه مورد رزرو شده خود را دريافت کنيد و يا مدت زمان رزرو به اتمام برسد، در خود نگه مي دارد.
فرد عضو	اما کامپيوترم چگونه داده هاي شخصي مرا خواهد شناخت؟
مهندس سيستم	پيش از هر چيز رابط وب، تنها توسط اعضاي کتابخانه مي تواند مورد استفاده قرار بگيرد. در نمودار کلاس، ID اعضا داريم، عددي که همراه تمام داده هاي شخصي ضروري، بر روي کارت عضويت تان چاپ شده است. وقتي داده اي را جستجو مي کنيد انتخابي از کتاب يا مجله براي رزرو کردن، ايجاد مي کنيد، وقتي دکمه را کليک مي کنيد اين ليست هم فرستاده خواهد شد. اما اگر شما عضو کتابخانه نباشيد، پيش از همه اين کارها بايد عضو کتابخانه شويد.
کتابدار	آيا اين امکان وجود دارد که کارمندان ما، زمانيکه در کتابخانه حضور ندارند نيز از سيستم استفاده کنند؟
مهندس سيستم	البته. کارمندان وارد کننده داده حتي زمانيکه در اداره محل کارشان حضور ندارند هم مي توانند کارشان را انجام دهند و مي توانند از هر نقطه دنيا که کامپيوتر و اتصال اينترنت دارد، به داده هاي database دسترسي پيدا کنند.
کتابدار	چگونه قصد داريد اين کار را امکان پذير کنيد؟
مهندس سيستم	اجازه دهيد الان نگران اين مساله نباشيم، به زودي خواهيد فهميد که اين امر چگونه تحقق خواهد يافت.

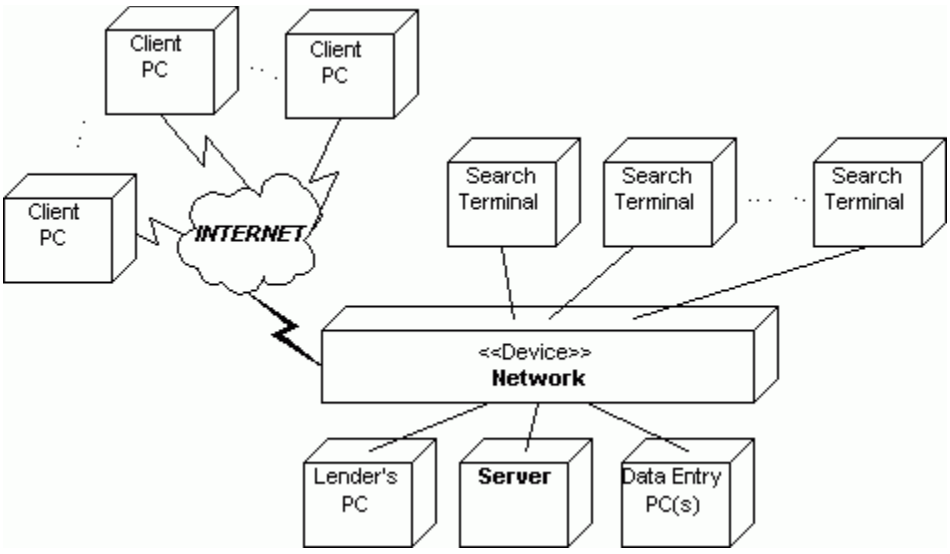
مسوول جلسات	پس همه كارها را انجام داده ايم؟ سيستم ما با شبكه محلي كتابخانه همكاري خواهد كرد و حالا وقتش رسيده كه براي سيستم خود، نامي انتخاب كنيم.
مهندس سيستم	كتابخانه ديجيتالي چطور است؟
مسوول جلسات	آيا همه با اين نام موافقند؟
	همه موافق هستند.
مسوول جلسات	بسيارخوب. فكر مي كنم كارمان در اينجا به پايان رسيده است.

جلسه JAD

حال که تيم کاري بينشي از سيستم بدست آورده است، آيا برنامه نويسان مي توانند برنامه بنويسند و مهندسين سيستم مهندسي کنند؟ نه کاملاً. تيم بايد در محدوده نيازهاي کاربران براي سيستم کتابخانه ديجيتالي تمرکز کند، نه در محدوده تکنولوژي خيلي بالا. گرچه آنها با ملاقات گروهي تا حدي بصيرت لازم را بدست آورده اند اما مفهوم کتابخانه ديجيتالي را براي گروه کارمندان و کاربران نهايي - اعضا براي بدست آوردن بازخورد و ايده هايي از نقطه نظر کاربر، به نمايش نگذاشته اند. کار بعدي که بايد با آن دست به گريبان شويم همين است. در جلسه مشترک توسعه نرم افزار (JAD)، تيم نيازهاي سيستم را جمع آوري و آنها را مستندسازي خواهد کرد.

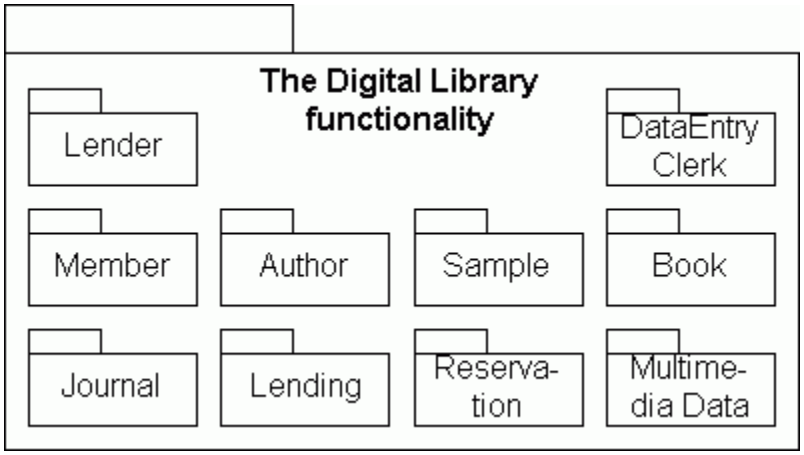
جلسه JAD در اتاق کنفرانس انجام مي شود که توسط يک مسوول جلسات هدايت ميگردد و به اين علت مشترک (joint) ناميده مي شود که اعضاي تيم توسعه را با کاربرهاي بالقوه سيستم و متخصصين حوزه، گرد هم مي آورد. اعضاي تيم توسعه شامل تحليلگر، مدلساز، دو برنامه نويس و يک مهندس سيستم مي باشند. کاربرهاي بالقوه شامل يک امانت دهنده، دو کارمند وارد کننده داده و يک کتابدار مي باشند. اين ملاقات يک نمودار package را توليد خواهد کرد که کارکردهاي اصلي سيستم را نشان مي دهد.

اين جلسه با سخنراني مسوول جلسات پيرامون مفهوم کتابخانه ديجيتالي آغاز مي شود. ساختار کتابخانه ديجيتالي در شکل زير نشان داده شده است که به همه افراد شرکت کننده در نشست داده شده است.



ساختار فيزيکي و ارتباطات بين اجزاي کتابخانه ديجيتال

گروه گفتگوهاي خود را با برآورد اينكه بخش هاي اصلي كاركردي كه سيستم بايد دارا باشد، چيست، آغاز مي كند. اين كار براي استفاده از كاركردهاي ذيل در سيستم انجام مي شود:



بسته كاركردي براي كتابخانه ديجيتالي

در زير بخشي از مكالمه انجام شده بين برخي از افراد شركت كننده آورده شده است:

مسوول جلسات	حال كه بخش هاي اصلي را داريم آيا كسي براي شروع جايي را مقدم ميداند؟
عضو اول	بخش عضو چطور است؟
مسوول جلسات	به نظر خوب مي آيد. بسيار خوب، چه وظيفه خاصي را مايليد در اينpackage ببينيد؟ به ياد داشته باشيد كه تنها به اين دليل اين بخش را انجام مي دهيم كه با كار ويژه شما همزمان نباشد و از ديدگاه هر كسي استقبال خواهد شد.
عضو اول	من قصد دارم كه جستجويي را از خانه خود نمايش دهم و كتاب يا مجله انتخاب شده اي را رزرو كنم، تا زمانيكه به كتابخانه بروم و آنها را به امانت بگيرم.
مسوول جلسات	بسيار خوب، ديگر چه؟
عضو دوم	من مي خواهم كتاب يا مجله الكترونيكي اي را مستقيماً از اداره خود print كنم.
تحليلگر	اتفاقاً، من اطلاعاتي راجع به مورد هاي الكترونيكي در نمودار هاي كلاس نمي بينم.

	چطور مي توانيم اين موارد را اداره كنيم؟
مدلساز	حق با شماست. در كلاس مورد بايد صفت جديدي براي يک كتاب يا مجله الكترونيكي اضافه كنيم. وضعيت يكي خواهد بود. از اين طريق مي توانيم نوع مورد يعني الكترونيكي بودن يا كاغذي بودن آنها را مشخص كنيم. همچنين بايد صفتي اضافه كنيم كه مكان مورد الكترونيكي را دربر داشته باشد (در انبار محلي يا در اينترنت) كه اين صفت لينک ناميده خواهد شد.
مسوول جلسات	آيا فرد ديگري وظيفه ديگري را مي خواهد اضافه كند؟
عضو اول	من مي خواهم ليستي از كتاب ها و مجلات پيشنهاد شده دركامپيوتر خودم داشته باشم.
مسوول جلسات	بسيار خوب، متوجه خواهيد شد كه اين موارد را در يک برچسب تخم مرغي شكل مينويسم. ما به اين موارد بعنوان مورد كاربردها باز مي گريم. از شما دعوت خواهيم كرد كه در ملاقات ديگري به اينجا برگرديد و ما را در تجزيه و تحليل مورد كاربرد هاي مورد استفاده ياري كنيد.

روزششم

در اين روز شما مطالب زير را خواهيد خواند:

- ✓ **تحليل كتابخانه ديجيتالي**
- ✓ تحليل مورد هاي كاربرد
- ✓ تعاملات در كتابخانه ديجيتالي
- ✓ يكپارچه كردن با ساير سيستم هاي
- ✓ **طراحي كتابخانه ديجيتالي**

تحليل سيستم

جمع آوري فرآيند هاي مورد نياز يک مدل کتابخانه ديجيتالي را به اتمام رسانده ايم. در طول اين جلسه، تحليل يک مدل کتابخانه ديجيتالي را انجام خواهيم داد.
در ابتدا با تحليلهاي مورد کاربرد مواجه خواهيم شد وپس از آن، اهدافي را که نمودارهاي مورد کاربرد براي هر بسته بدست خواهند آورد، بررسي خواهيم کرد. سپس قصد داريم که تعاملات بين اشياء را در اين کتابخانه ديجيتالي کشف و تحليل نماييم و نمودارهاي همکاري را توليد کنيم. در انتهاي اين جلسه تعاملات بين کتابخانه ديجيتالي خود را با سيستم هاي موجود مانند شبکه محلي کتابخانه بررسي خواهيم کرد و نمودار پياده سازي را با جزئيات توليد خواهيم نمود و تاحدي رابط هاي کاربر معرفي خواهند شد.
بياييد کارمان را آغاز کنيم و موارد داده شده در اين آخرين جلسه دوره UML را بياموزيم.

توسعه مورد کاربرد

مورد کاربردهاي نمودار package ، در درس " جلسه JAD " از جلسه پنجم، تصوير خوبي از آنچه كه سيستم بايد انجام دهد را ارائه داده است. تيم به تدريج از درك قلمرو به سمت درك سيستم حركت نمو ده اند. مورد كاربردها بين آنها پلي فراهم خواهند كرد.

در هيچ نقطه اي از جلسه JAD، تيم توسعه بحثي درباره اينكه چگونه سيستم مي تواند تمام فعاليت هاي تعيين شده در موردكاربردها را به انجام برساند، نكرده است. اين ايده، تمام مورد كاربردهاي ممكن را به تک تک به حساب آورده است. چنانچه مورد كاربردها در اين جلسه مورد بررسي قرار مي گيرند، توجه داشته باشيد كه چگونه اجزاي سيستم كتابخانه ديجيتالي جامه عمل به خود خواهند پوشيد. در اين مرحله كه تلاش مان بر توسعه مي باشد، آنچه مركز توجه مان قرار مي گيرد، سيستم است.

براي تحليل مورد كاربردها ما نياز به برگزاري جلسه ديگري داريم. مباحث در اين جلسه JAD به منظور استنتاج تحليلها براي هر مورد كاربرد است.

جلسه JAD مورد كاربرد معمولاً سخت ترين آنهاست، به طوريكه براي شركت كنندگان- كه كاربرهاي بالقوهٔ سيستم به اتمام رسيده هستند- به اين نام خوانده ميشود كه بياييد تحليلگر شويم. هر يک از آنها متخصص حوزه خود هستند و شما بايد از تخصص آنها بهره برداري كنيد. مخصوصاً آنها عادت ندارند آنچه را كه مي دانند به طور شفاهي بيان كنند و يا تجزيه كنند. همچنين ممكن است اين قضيه درست باشد كه آنها قبلاً بخشي از يک طراحي سيستم نبوده اند و ممكن است تلاش براي تعيين آنچه يک سيستم مي تواند براي كمک به كار آنها انجام دهد برايشان نا خوشايند باشد.

مراحل وابسته به سيستم در اين سناريو بسيار حائز اهميت است زيرا نشان ميدهد سيستم براي كار كردن چگونه فرض شده است. وقتي شركت كننده هاي جلسه JAD، اين مراحل را به ما اعلام مي كنند، در واقع به ما مي گويند كه سيستم در نهايت بايد چگونه باشد. پس از اين جلسه JAD، ما بايد تصور خوبي راجع به اجزاي سيستم داشته باشيم.

در درس " معرفي يک مدل مورد كاربرد "، در درس دوم ديديم كه هر مورد كاربرد، يك توالي از سناريوها مي باشد و هر سناريو توالي از مراحل است. مي خواهيم براي هر سناريو در هر مورد كاربرد، موارد زير را نشان دهيم:

- شرح مختصري از سناريو.
- فرض هاي موجود براي سناريو.
- Actorي كه مورد كاربرد را آغاز مي كند.
- پيش شرطهاي مورد كاربرد.
- مراحل وابسته به سيستم در سناريو.

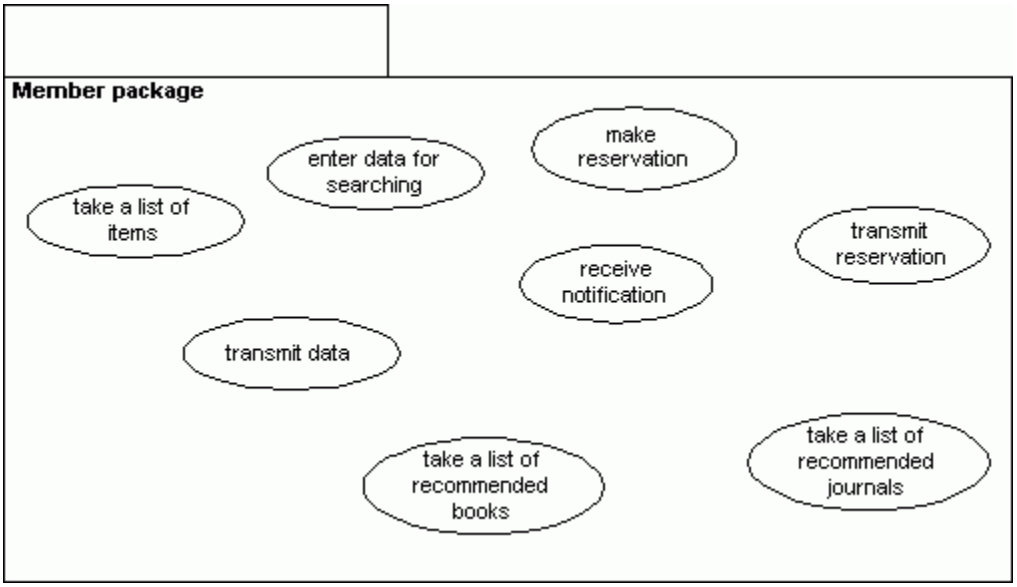
- شرط هاي پسين وقتي سناريو كامل ميشود.
- Actorي كه از مورد كاربرد سود مي برد.

مورد كاربردهايي كه قصد امتحان كردن آنها را داريم بسيار زيادند و به همين علت است كه وقتي قصد انجام تحليل يک مورد كاربرد را داريم، تعدادي از آنها به تفصيل بررسي مي شوند و بقيه آنها به عنوان تمرين به خودتان واگذار مي شوند. حال اجازه دهيد به تحقيق و تفحص در روند تحليل مورد كاربرد بپردازيم.

تحليل مورد كاربرد – بخش اول

در اين درس با فرآيند تحليل مورد كاربرد مواجه خواهيم شد. قصد داريم كه يک package عضو مورد كاربرد وpackage كتاب مورد كاربرد را بررسي كنيم و بقيه آنها به خودتان واگذار مي شوند تا خودتان براي تحليل مورد كاربردها تلاش كنيد.

package عضو



package عضو مورد كاربردها كه در مراحل اوليه تحليل سيستم توسعه يافته است

وارد كردن داده براي جستجو

عبارت مناسبي كه اين مورد كاربرد را توصيف كند مي تواند اين باشد: عضو داده اي را براي جستجو درون فرم جستجو، وارد مي كند.

در اينجا actor آغازكننده، يك عضو خواهد بود كه در واقع actor سود برنده نيز مي باشد.

فرض بر اين است كه يک عضو مي خواهد كه كتاب يا مجله اي را بيابد، فرض ديگر اين است كه رابط كتابخانه ديجيتالي، فرم اختصاصي براي وارد كردن داده هاي جستجوي داشته باشد.

پيش شرط اوليه اين است كه يک فرد عضو، يک داده ضروري براي جستجو داشته باشد و شرط پسين اين است كه عض، يک داده ضروري را به كتابخانه ديجيتالي وارد كند.

مراحل مورد کاربرد عبارتند از:

- عضو، از شبکه LAN کتابخانه یا از اداره یا منزل خود، رابط کاربر را براي جستجوي وارد کردن، فعال مي کند.
- فرم جستجو در صفحه نمايش ظاهر مي شود.
- عضو، داده ضروري را درون فرم جستجو وارد مي کند.

انتقال داده براي جستجو

عبارت مناسبي که اين مورد کاربرد را توصيف کند مي تواند اين باشد: داده وارد شده در فرم جستجو را بردار وآن را به موتور جستجوي databse کتابخانه ديجيتالي ارسال کن.

فرض براين است که با اينترنت يا شبکه LAN کتابخانه در ارتباط هستيم و در رابط عضو يک فرم جستجو داريم.

بايد برخي از فرضيات را دوباره بيان کنيم زيرا هر مورد کاربردي سرانجام در يک صفحه مجزا در مستند طراحي شده ظاهر خواهد شد که به عنوان مرجعي درباره سيستم بکار گرفته خواهد شد.

پيش فرض اوليه، وارد شدن داده در فرم جستجو و شرط انتهايي، رسيدن داده به موتور جستجوي databaseکتابخانه ديجيتالي مي باشد.

مراحل مورد کاربرد به صورت زير است:

- بر روي دکمهٔ تائيد SUBMIT که بر فرم جستجو قرار دارد کليک کنيد.
- مکانيزم انتقال داده را در شبکه فعال کنيد.
- داده به موتور جستجويdatabase کتابخانه ديجيتالي مي رسد.
- در صفحه نمايش اعضا، پيغامي مبني بر اينکه داده فرستاده شده است، و پيغامي مبني بر اينکه در طول رسيدن داده لحظه اي صبر شود، ديده مي شود.

به علت اينکه اين مورد کاربرد با مورد کاربرد قبلي وابسته است، بايد تغييراتي در package عضو ايجاد کنيم که وابستگي <<include>> بين اين مورد کاربرد و مورد کاربرد " وارد کردن داده براي جستجو" را نشان دهد.

گرفتن ليستي از موارد جستجو شده

اين مورد کاربرد، آخرين بخش تکميل فرآيند جستجو مي باشد. عبارت مناسبي که اين مورد کاربرد را توصيف کند مي تواند اين باشد: دريافت ليستي از کتاب يا مجلات، پس از وارد کردن داده در فرم جستجو و فرستادن آن به database کتابخانه ديجيتالي.

فرض بر اين است که فرد عضو از يک فرم جستجو استفاده مي کند و ارتباط با شبکه برقرار است.

پيش شرط اوليه اين است كه بر روي دكمه تائيد SUBMIT در فرم جستجو كليک شده باشد و شرط انتهايي چاپ كردن ليست موارد بر صفحه نمايش عضو مي باشد. در اينجا نيزactor فرد عضو مي باشد.

مراحل مورد كاربرد به صورت زير مي باشد:

- صبر كنيد تا جستجو انجام گيرد(به موازات يک كتابخانه ديجيتالي، موتور جستجو، جستجورا انجام ميدهد.)
- database كتابخانه ديجيتالي، ليستي از موارد را از طريق شبكه به رابط عضو انتقال مي دهد.
- فرد عضو ليستي از مواردي كه در داده هاي جستجوي وارد شده، صدق مي كند، دريافت مي نمايد .

Inclusion نيز دراين مورد كاربرد ظاهر خواهد شد. « include » اين مورد كاربرد را به مورد كاربرد قبلي وصل مي كند.

رزرو كردن

عبارت مناسبي كه اين مورد كاربرد را توصيف كند مي تواند اين باشد: فرد عضو مي خواهد براي موردي كه از طريق فرآيند جستجو و يا فرآيند پيشنهاد بدست آمده است درخواست رزرو نمايد.

چنانچه مي بينيد اين مورد كاربرد به دو مورد كاربرد متفاوت وابسته است: جستجو و پيشنهاد، بنابراين فرآيند رزرو كردن ممكن است زماني رخ دهد كه به يكي از اين دو فرآيند رجوع شده باشد. به همين علت است كه، include بين اين سه مورد كاربرد ارائه شده است.

فرضيات اوليه بر ليستي از موارد و دكمه اي براي رزرو كردن استوار است.

پيش فرض اوليه، كليک كردن دكمه "رزرو" و شرط انتهايي انتقال پرونده رزرو به كتابخانه ديجيتالي ميباشد.

مراحل مورد كاربرد به صورت زير است :

- فرد عضو ليستي از موارد، از طريق جستجو و يا از طريق پيشنهاد، دريافت مي كند.
- فرد عضو براي موردي كه مي خواهد رزرو كند بر روي دكمه رزرو كليک مي كند.
- فرد عضو اطلاعات فرمي را كه براي رزرو كردن ظاهر مي شود، پر ميكند.

ارسال رزرو کردن

اين مورد کاربرد به تفصيل شرح داده نمي شود، زيرا تقريباً شبيه مورد کاربرد " انتقال داه براي جستجو " مي باشد. فقط ميگوييم که اين مورد کاربرد با مورد کاربردهاي "رزرو کردن " و "دريافت پيام" در ارتباط است. ارتباط نيز <<include>> مي باشد.

دريافت پيام

عبارت مناسبي که اين مورد کاربرد را توصيف کند مي تواند اين باشد: پس از فشردن دکمه تائيد SUBMIT در فرم رزرو باید ا پيامي از کتابخانه ديجيتالي مبني بر اينکه: رزرو پذيرفته شده است، وبراي آگاه کردن عضو از وضعيت رزرو، برسد.

در اينجا منظور از وضعيت رزرو، داشتن نمونه آزاد مورد رزرو شده، مي باشد و اگر يک نمونه آزاد در کتابخانه موجود باشد، آنگاه آن نمونه به مدت چند هفته براي فرد عضو کنار نگه داشته مي شود.

فرض بر وجود ارتباط مناسب در شبکه(ها) و کليک دکمه تائيد مي باشد.

پيش فرض اوليه، کليک کردن دکمهٔ " تائيد " و شرط انتهايي دريافت وضعيت رزرو است.

مراحل مورد کاربرد به صورت زير مي باشد:

- تا زمانيکه اخطار برسد صبر کنيد.
- اطلاعات ضروري را از طريق شبکه به رابط عضو انتقال دهيد.
- فرد عضو، وضعيت رزرو را دريافت مي کند.

گرفتن ليستي از موردهاي پيشنهاد شده

عبارت مناسبي که اين مورد کاربرد را توصيف کند مي تواند اين باشد: فرد عضو بخش پيشنهاد سيستم را فعال مي کند و به ليستي ازموارد پيشنهاد شده دست مي يابد.

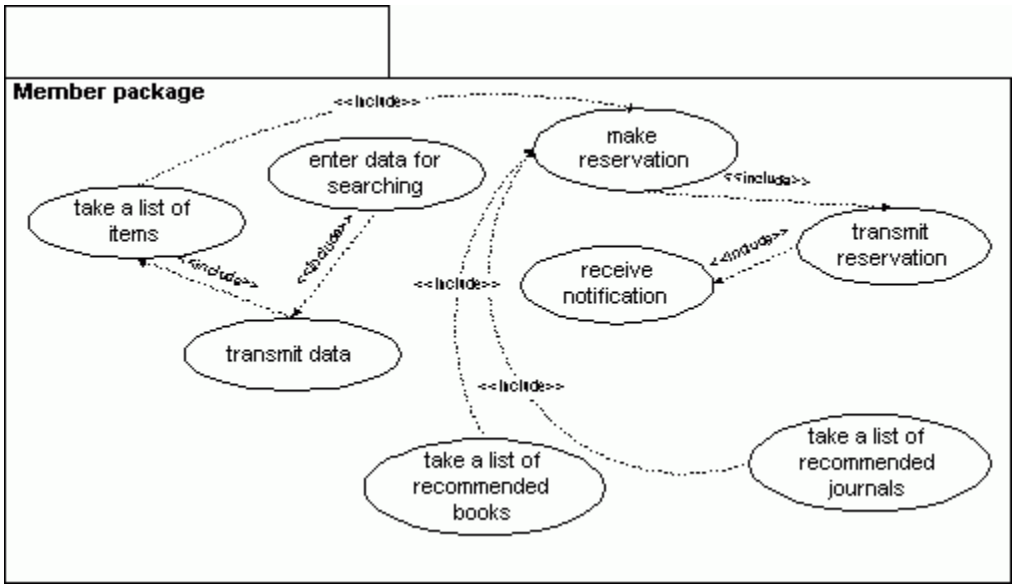
فرض بر کليک دکمه پيشنهاد و دايربودن يک الگوريتم پيشنهاد، مي باشد.

پيش فرض اوليه، کليک کردن دکمه " پيشنهاد " و شرط انتهايي دريافت ليستي از موارد مي باشد.

آغاز کننده و سود برنده اين کار، فرد عضو مي باشد.

مراحل مورد کاربرد به صورت زير است:

- فعال کردن بخش پيشنهاد از رابط کاربر.
- کتابخانه ديجيتالي الگوريتم پيشنهاد را فعال مي کند، که اين الگوريتم، ليستي از موارد را فعال مي کند.
- چاپ کردن اين ليست از موارد در صفحه نمايش فرد عضو.

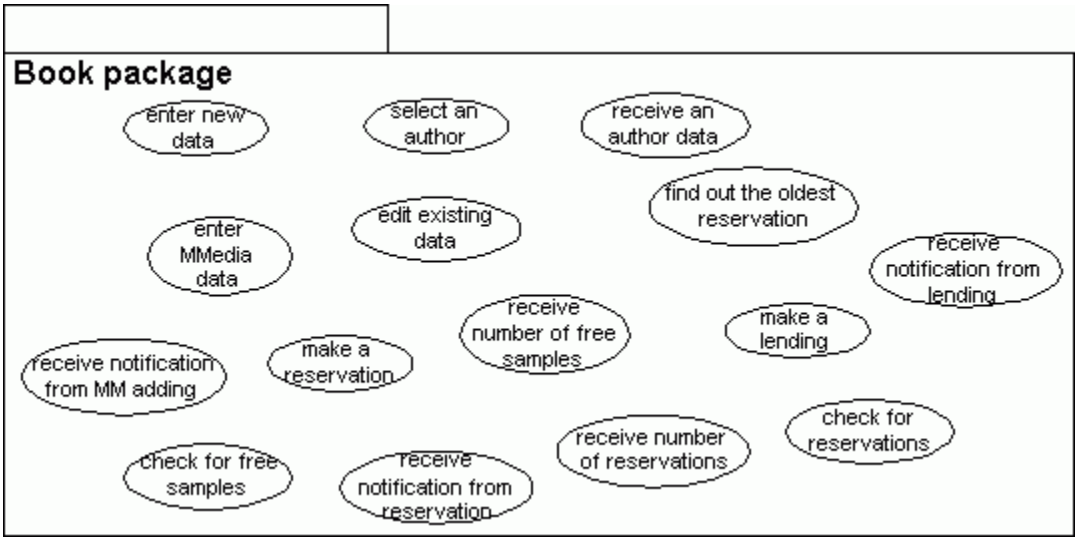


نمودار مورد کاربرد به روز شده براي package عضو

تحليل مورد كاربرد – بخش دوم

حال بياييد به مورد كاربرد هاي package كتاب نگاهي بياندازيم و آنها را به طور مجزا تحليل كنيم.

package كتاب



package كتاب مورد كاربردهايي كه در مراحل اوليه تحليل سيستم توسعه يافته است.

وارد كردن داده براي يک كتاب جديد

توضيح: داده را در فرم ثبت كتاب پر كن.

آغاز كننده اين كار، كارمند وارد كننده داده مي باشد.

فرضيات: فرم ثبت در رابط كارمند وارد كننده داده فعال شده و اطلاعات وارد شده است.

پيش فرض اوليه، داده مربوط به كتاب جديد و شرط پسين وارد شدن داده درdatabase كتابخانه ديجيتالي ميباشد.

مراحل چنين ميباشند:

- از طريق شبكه LAN كتابخانه يا از طريق اينترنت، كارمند وارد كننده داده، رابط را براي ثبت كتاب، فعال مي كند.
- فرم در صفحه نمايش ظاهر خواهد شد.
- كارمند ، اطلاعات ضروري را وارد مي كند.

انتخاب یک نویسنده

توضیح: انتخاب یک نویسنده از لیست نویسنده.

آغاز کننده این کار ، کارمند وارد کننده داده مي باشد .

فرضیات: در database، مدارک نویسنده وجود دارد و کاربر از آن لیست انتخاب مي کند.

پیش فرض اولیه، داده نویسنده کتاب و شرط انتهايي انتخاب شدن داده نویسنده مي باشد.

مراحل چنین میباشند:

- کارمند ، در فرم ثبت کتاب ، نویسنده اي را از لیست نویسنده ها انتخاب مي کند.
- این اطلاعات به database فرستاده مي شود.
- جستجوي این اطلاعات در database انجام مي شود.

یک inclusion مي تواند در اینجا معرفي شود. این مورد کاربرد مراحلي از مورد کاربرد "ویرایش داده براي یک کتاب موجود " و مورد کاربرد " وارد کردن داده براي کتاب جدید" را شامل مي شود.

اصلاح داده نویسنده

توضیح: اصلاح تمام اطلاعات ضروري درباره نویسنده انتخاب شده در مورد کاربرد قبلي.

فرضیات : کاربر انتخابي از لیست نویسنده انجام مي دهد.

پیش فرض اولیه، انتخاب نویسنده و شرط پسین بازیافت اطلاعات نویسنده مي باشد.

مراحل مورد کاربرد به صورت زیر مي باشد:

- پس از جستجو، database پرونده نویسنده را انتخاب مي کند و اطلاعات ضروري نویسنده را استخراج مي کند.
- این اطلاعات به فرم وارد کردن دادهٔ کتاب فرستاده مي شود.
- چاپ اطلاعات در صفحه نمایش کاربر
- این مورد کاربرد، مورد کاربرد "انتخاب یک نویسنده " را شامل مي شود.

وارد کردن دادهٔ مولتي مدیا

توضیح: اگر کتاب، داده مولتي مدیا دارد، پس در مدارک کتاب وارد شده است

فرضیات: کتاب داده مولتي مدیا دارد و فایل یا فایل هاي مولتي مدیا، جایي در سرور ذخیره شده اند.

پیش فرض اولیه، این است که کتاب داده مولتي مدیا داشته باشد ودکمهٔMMEDIA کلیک شده باشد و شرط انتهايي وارد شدن داده مولتي مدیا مي باشد.

مراحل مورد کاربرد به صورت زیر است:

- اگر كتاب دادهٔ مولتي مديا داشته باشد، دكمهٔ MMEDIA كليک شده و فرم وارد كردن برصفحه نمايش ظاهر خواهد شد .
- كاربر نوع داده مولتي مديا و مكاني بر روي سرور، كه داده در آنجا ذخيره مي شود را وارد ميكند .
- دكمه تائيد كليک شده و پرونده مولتي مديا به database انتقال داده مي شود .
- يک inclusion مي تواند در اينجا معرفي شود، اين مورد كاربرد مراحلي از مورد كاربرد "ويرايش داده براي يک كتاب موجود " و مورد كاربرد "وارد كردن داده براي كتاب جديد " را شامل مي شود.

دريافت پيام پس از افزودن مولتي مديا

توضيح – پس از پر كردن فرم واردكردن مولتي مديا، database پيغامي مبني بر اينكه موفقيت فرايند افزودن داده مولتي مدياي جديد چگونه نمايش داده شده است، مي فرستد.

فرضيات – پركردن فرم وارد كردن داده مولتي مديا، ارتباط خوب از طريق شبكه، وجود فايل مولتي مديا .

پيش فرض – انتقال فرم وارد شده مولتي مديا به database

فرض انتهايي – دريافت موقعيت اين انتقال

مراحل انجام اين كار به صورت زير مي باشد:

- داده مولتي مديا به database مي رسد و فرايند ايجاد پرونده جديد آغاز مي شود.
- اين فرآيند مي تواند به نتيجه برسد وپايان يابد يا به شكست منجر شود.
- چگونگي خاتمه فرآيند به فرم وارد كردن داده كتاب، مربوط مي شود.

اين مورد كاربرد، مورد كاربرد قبلي، " وارد كردن داده مولتي مديا " را در بر مي گيرد.

ويرايش داده براي يك كتاب موجود

توضيح – اگر اطلاعات يك كتاب اشتباه وارد شده باشد، امكان تغيير آن اطلاعات وجود دارد.

فرضيات – وجود پرونده كتاب براي تغيير دادن و كليك بر روي دكمه EDIT

پيش فرض – اطلاعات وارد شده در پرونده كتاب نادرست باشد.

فرض انتهايي – تغيير پرونده كتاب

مراحل انجام اين كار به صورت زير مي باشد:

- کاربر دکمه EDIT را کليك مي کند و فرم را با اطلاعات قديمي کتاب که بر صفحه نمايش ظاهر مي شود، پر مي کند.
- اطلاعات نادرست انتخاب و تغيير داده مي شوند.
- اگر اطلاعاتي که بايد تغيير داده شود مربوط به اطلاعات نويسنده يا اطلاعات مولتي مديا باشد، فرآيند مشابه اي مانند وارد کردن داده جديد، تکرار مي شود.

ايجاد يك رزرو

توضيح – فرد عضو مي خواهد نمونه اي از کتاب را رزرو کند و پس از آن امانت گرفتن در کتابخانه مي تواند اجرا شود.

فرضيات – کتاب بايد درطول فرآيند پيشنهاد يا جستجو در ليست گرفتن کتاب ها حضور داشته باشد، کتاب ها، بايد نمونه اي در کتابخانه، موجود داشته باشند.

پيش فرض – فرد عضو مي خواهد رزروي براي يك کتاب ايجاد کند.

فرض انتهايي – داده هاي ضروري کتاب و عضو، به کتابخانه ديجيتالي انتقال داده مي شود.

مراحل انجام اين کار به صورت زير مي باشد :

- عضو دکمه RESERVE را کليك مي کند و اطلاعات ضروري را پر مي کند تا رزرو کردن انجام شود.
- database کتابخانه ديجيتالي درخواست رزرو را دريافت مي کند.
- مراحل بررسي داده با پرونده هاي database ايجاد مي شود .
- کتابخانه ديجيتالي موجود بودن نمونه آزاد را بررسي مي کند و رزرو جديدي را اجرا مي کند.
- پس از آن، هنگاميکه پيام رزرو به انجام مي رسد، دوره انتظاري معرفي مي شود.

دريافت اخطار رزرو

توضيح – پس از کليك کردن دکمه RESERVE و پر کردن فرم، کتابخانه ديجيتالي درخواستي براي رزرو دريافت مي کند و پس از انجام فرآيند رزرو، پيغام موقعيتي دريافت مي شود.

فرضيات – دکمه RESERVE و فرستادن درخواستي براي رزرو

پيش فرض – کليك کردن دکمه RESERVE

فرض انتهايي – موقعيت رزرو

مراحل انجام اين کار به صورت زير مي باشد:

- موقعيت انجام يك رزرو جديد کسب مي شود.
- اين موقعيت از طريق شبکه براي کاربر فرستاده مي شود.

- پيغام موقعيت در صفحه نمايشprint مي شود.

اين مورد كاربرد، مورد كاربرد " ايجاد رزرو " را در بر دارد.

بررسي نمونه هاي آزاد

توضيح – شمردن اينكه چه تعداد نمونه آزاد از آن كتاب قابل دسترس مي باشد و ارجاع آن تعداد به فرآيند رزرو

فرضيات – درخواست رزرو، نمونه (نمونه ها) براي كتاب

پيش فرض – رزرو جديد

فرض انتهايي – تعداد نمونه هاي آزاد

مراحل انجام اين كار به صورت زير مي باشد:

- دريافت درخواستي براي تعداد نمونه هاي آزاد
- شمردن تعداد نمونه هاي قابل دسترس
- فرستادن تعداد براي كتابخانه ديجيتالي

اين مورد كاربرد، مورد كاربرد " ايجاد رزرو " را در بر دارد.

بازگرداندن تعداد نمونه هاي آزاد

توضيح – دادن تعداد نمونه هاي آزاد

فرضيات – نمونه هاي آن كتاب، درخواست براي تعداد نمونه هاي آزاد

پيش فرض – درخواست براي تعداد نمونه هاي آزاد

فرض انتهايي – انتقال تعداد نمونه هاي آزاد

مراحل انجام اين كار به صورت زير مي باشد:

- دريافت تعداد نمونه هاي آزاد انتقال داده شده
- بررسي نتيجه براي تعيين موقعيت رزرو
- انتقال موقعيت رزرو به كتابخانه ديجيتالي

اين مورد كاربرد، مورد كاربرد " بررسي نمونه هاي آزاد " را در بر دارد.

امانت گرفتن

توضيح – عضو مي خواهد كتابي امانت بگيرد.

فرضيات – كاربر بايد در كتابخانه حاضر باشد و كتاب بايد به شكل كاغذي باشد و نمونه آزادي براي كتاب در دسترس باشد.

فرد آغاز كننده، امانت دهنده و سود برنده فرد عضو مي باشد.

پيش فرض – درخواست امانت گرفتن

فرض انتهايي – انتقال درخواست امانت گرفتن به كتابخانه ديجيتالي

مراحل انجام اين كار به صورت زير مي باشد:

- امانت دهنده رابط را براي امانت گرفتن و امانت دادن فعال مي كند.
- امانت دهنده داده هاي نمونه و عضو را در فرم وارد كردن داده وارد مي كند.
- پس از فشردن دكمه LEND فرآيند امانت جديدي فعال مي شود.

دريافت پيام براي امانت گرفتن

توصيف – پس از فشردن دكمه LEND در فرم امانت، پرونده رزرو جديدي در database نوشته مي شود و نتيجه اين معامله به رابط امانت دهنده بازگردانده مي شود.

فرضيات – ارتباط از طريق شبكه، دكمه LEND در رابط موجود باشد.

پيش فرض – وارد كردن داده در فرم امانت

فرض انتهايي – موقعيت انجام معامله

مراحل انجام اين كار به صورت زير مي باشد:

- تا زمان خاتمه فرآيند امانت صبر كنيد و موقعيت نتيجه امانت را دريافت كنيد.
- انتقال موقعيت از طريق شبكه به صفحه نمايش امانت دهنده
- اگر نمونه آزادي موجود باشد، فرد يكي از آنها را امانت مي گيرد.

اين مورد كاربرد، مورد كاربرد " ايجاد امانت " را در بر دارد.

مورد كاربردهاي مرتبط ديگر با فرآيند امانت، عبارتند از

دريافت تعداد رزرو: اين مورد كاربرد زماني رخ مي دهد كه پرونده امانت جديدي ايجاد مي شود. اگر كتابي رزرو معلق داشته باشد (نمونه هاي آزاد با "status="reserved در دسترس باشند) و هيچ نمونه آزاد "status="available با موجود نباشد، آنگاه فرآيند امانت با شكست مواجه مي شود.

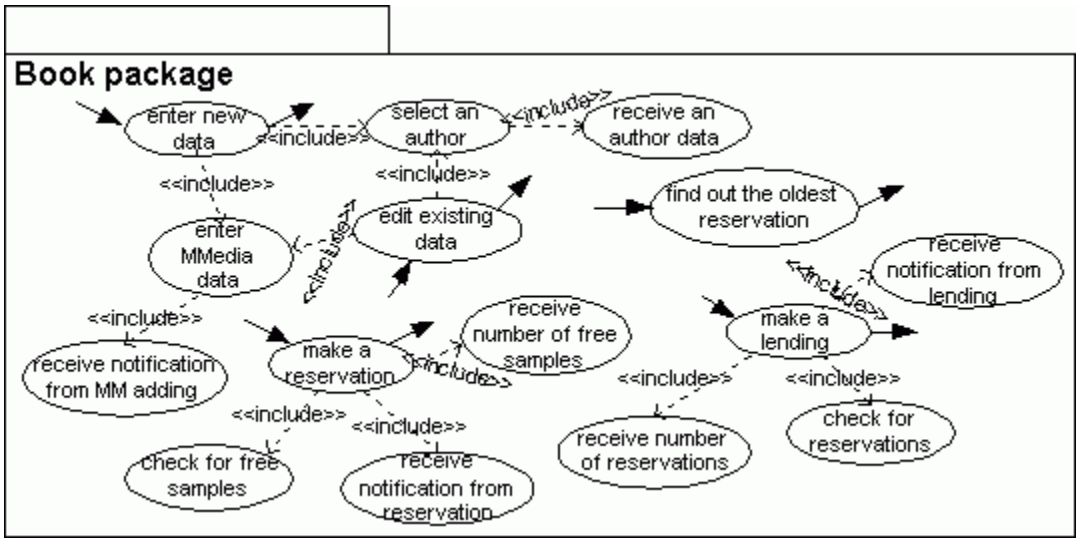
بررسي رزرو

اگر عضو رزروي براي نمونه اي از يك كتاب داشته باشد، آنگاه اگر نمونه نگهداشته شده اي در دسترس باشد، عضو مي تواند آن را به امانت بگيرد، در غير اين صورت فرآيند امانت با شكست مواجه مي شود.

يافتن قديمي ترين رزرو

اين مورد كاربرد در فرآيند به روز كردن پرونده هاي رزرو استفاده مي شود. وقتي عضوي درخواستي براي رزرو ايجاد مي كند و نمونه آزادي از آن موجود نمي باشد آنگاه هنگاميكه تنها يك نمونه از آن آزاد مي شود، اين مورد كاربرد رخ مي دهد و قديمي ترين رزرو را براي آن كتاب پيدا مي كند و موقعيت رزرو را از "available" به "reserved" تغيير مي دهد. نمونه هنگامي مي تواند آزاد شود كه از طرف فردي كه امانت گرفته شده است، باز گردانده شود.

مورد كاربردهايي كه در اينجا ظاهر مي شوند بين مورد كاربردهاي " ايجاد امانت " و " بررسي رزرو " و مورد كاربردهاي " ايجاد امانت " و " دريافت تعداد رزروها " مي باشند.

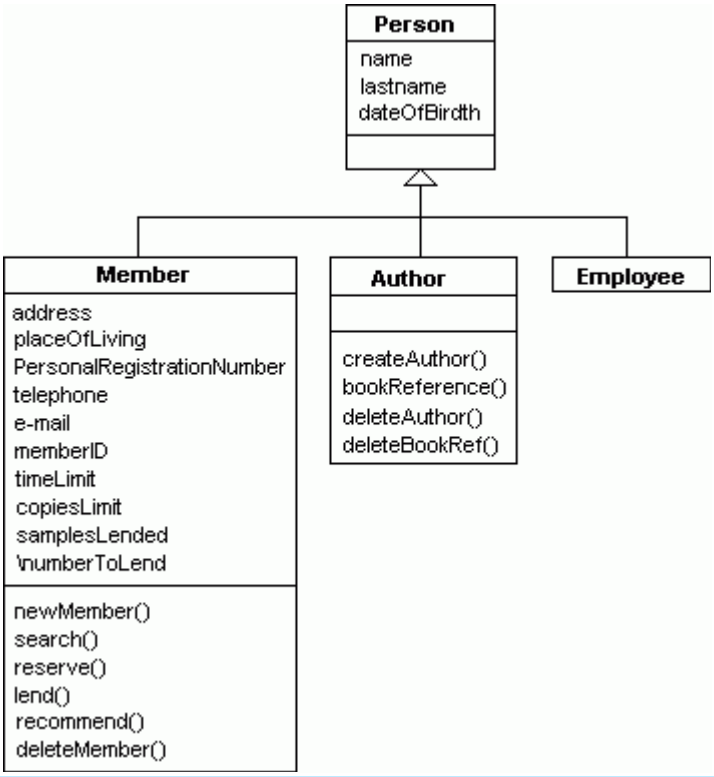


مورد كاربرد به روز شده براي package كتاب

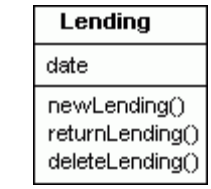
تحليل مورد كاربردها به تصور بخش هاي كاري سيستم كمك كرده است. هر چند مسائل بسياري درباره مورد كاربردها مي دانيم، هنوز بايد چگونگي كاركردن بخش هايي كه با بخش هاي ديگر در تبادل هستند و چگونگي تغيير حالت آنها، را مدلسازي كنيم. عبور از اين اطلاعات كار برنامه نويسان را آسان تر مي سازد. آنها بايد نگرشي بر چگونگي كدگذاري كلاس ها و كار كردن آنها با يكديگر داشته باشند.

تصحيح نمودار كلاس

حال زمان آن رسيده كه نمودار كلاس خود را كامل كنيم. هر كلاسي كه از تحليل هاي قبلي بدست آمده است را تحليل خواهيم كرد .
واضح است كه بايد يک فرآيند حذف در مقابل فرآيند ايجاد كردن داشته باشيم زيرا براي اغلب كلاس ها عملياتي براي ايجاد كردن داريم، بنابراين عملياتي براي حذف يک موضوع نيز بايد در هر كلاسي فراهم شود.
صفت هاي جديدي بايد به كلاس عضو اضافه شود كه عبارتند از: نام مستعار ، رمز عبور و وضعيت. نام مستعار و رمز عبور صفت هايي هستند كه براي وارد شدن به سيستم به كار مي روند. واضح است كه رابط اينترنت به تعداد بسياري از افراد، اجازه دستيابي به كتابخانه ديجيتالي را مي دهد اما تنها آن افرادي كه قرارداد عضويت در كتابخانه را دارند مي توانند داده ها را مرور كنند، رزرو انجام دهند و موردهاي الكترونيكي را Print كنند .
صفت وضعيت براي تمايز بين يک عضو معمولي و عضوي كه كارمند كتابخانه مي باشد، معرفي شده است، زيرا قبلاًگفته بوديم كه هر كارمند كتابخانه، مي تواند عضو كتابخانه هم باشد.
حال بر اساس تحليل مورد كاربرد قبلي، مي توانيم عملياتهاي كلاس مورد را معرفي كنيم. اين عملياتها عبارتند از: يافتن قديمي ترين رزرو ()، بررسي رزرو () و دريافت تعداد رزروها.
كه اين عملياتها را در درس قبلي شرح داديم.
اگرنگاهي به صفت هاي كلاس عضو، صفت هاي كلاس كارمند و صفت هاي كلاس نويسنده بياندازيم، خواهيم ديد كه بيشتر آنها براي اين سه كلاس مشترک هستند. بنابراين به نوعي صفت هاي ارثي تعريف مي شوند و براي اين منظوركلاس جديدي به نام كلاس "شخص" ايجاد مي كنيم كه صفت هاي نام ، نام خانوادگي و تاريخ تولد را در بر داشته باشد.
توصيف كلاس ها خيلي ضروري نمي باشد زيرا همه كلاس ها را با حالت ابتدا و انتها خواهيد ديد كه در ليست كلاس هاي درزير نشان داده شده اند:

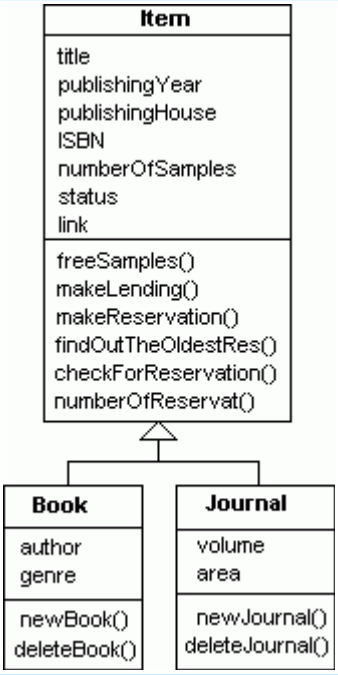


کلاس والد فرد به همراه فرزندان آن

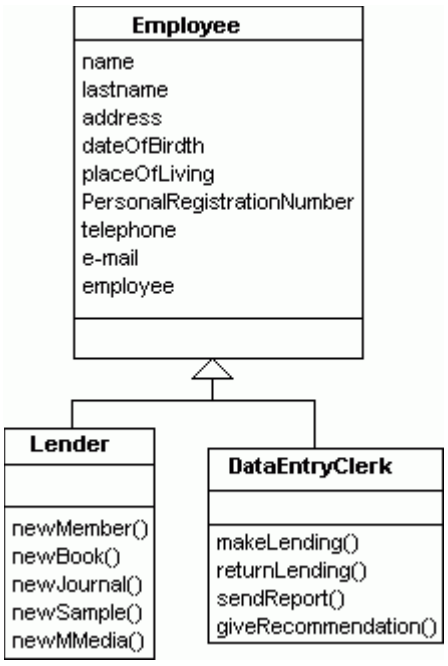


کلاس امانت گيري

کلاس مولتي مديا

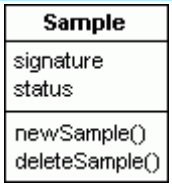


وراثت کلاس مورد



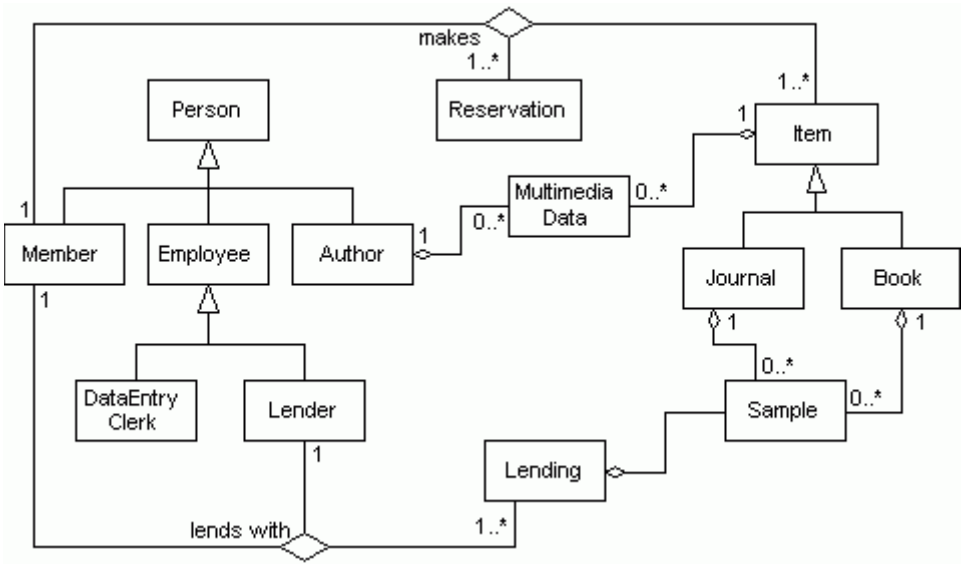
کلاس کارمند و فرزندان آن

کلاس رزرو کردن



کلاس نمونه

Association هاي بين اين كلاس بصورت زير است:



تعاملات در سیستم – بخش اول

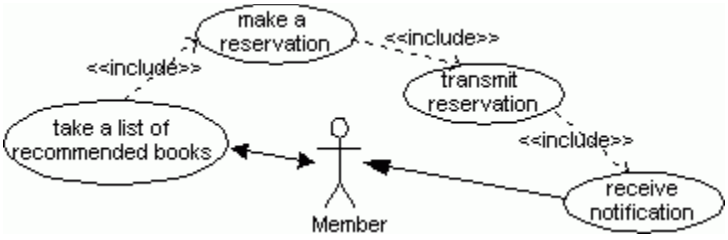
یک راه براي پي بردن به تعاملات در سیستم این است که اجزاء پیشنهاد شده سیستم را در هر Package مورد کاربرد سیستم، یک یک برشماریم. براي انجام این امر باید نشان داده شود که اجزاي سیستم ، براي کامل کردن هر مورد کاربرد چگونه رفتار مي کنند.گرچه در درس گذشته کاملاً تمام مورد کاربردها را در همه Packageها تحلیل نکردیم اما هنوز مي توانیم اجزاي سیستم را براي آن مورد کاربردهایي که انجام داده ایم، استخراج کنیم. اجازه دهید Package هایي را که در درس هاي گذشته تحلیل کرده ایم، دوباره تحلیل کنیم. Package هاي درس گذشته، Package عضو و Package کتاب بودند. قصد داریم که تعاملات بین اجزاء ترکیب کننده این Package ها را تنها براي فرآیند امانت گرفتن و رزرو کردن بیابیم زیرا این دوره کوتاه تر از آن است که تمام تعاملات بین اجزا را در بر داشته باشد، بنابراین تنها این دو تعاملات را تحلیل خواهیم کرد و بقیه آنهایي که نیاز خواهید داشت که انجام دهید، به عنوان تمرین، در بخش کارگاه این جلسه آورده خواهد شد.

Package عضو

مورد کاربرد هاي Package عضو

- وارد کردن داده براي جستجو
- انتقال داده به database کتابخانه
- گرفتن لیستي از آیتم ها
- ایجاد رزرو براي بعضي از آنها
- انتقال رزرو به database کتابخانه
- دریافت اخطار از database کتابخانه
- گرفتن لیستي از کتابهاي پیشنهاد شده
- گرفتن لیستي از مجلات پیشنهاد شده

بیایید فرآیند رزرو کتاب را دوباره بررسي کنیم. فرض کنید که مي خواهید از روي لیست کتاب هاي پیشنهاد شده، کتابي را رزرو کنید. حال بیایید به مراحل لازم براي انجام این کار نگاهي بیاندازیم:



بخشي از نمودار مورد کاربرد در Package عضو

گرفتن ليستي از كتابهاي پيشنهاد شده

مراحل اين مورد كاربرد عبارتند از:

١. فعال كردن بخش پيشنهاد از رابط عضو
٢. پس از كليک كردن دكمه پيشنهاد بر روي رابط، علامتي به كتابخانه ديجيتالي فرستاده مي شود تا الگوريتم پيشنهاد دادن را فعال كند.
٣. كتابخانه ديجيتالي، الگوريتم پيشنهاد دادن كه ليستي از كتاب هاي پيشنهاد داده شده را توليد مي كند، فعال مي كند.
٤. كتابخانه ديجيتالي اين ليست كتابها را به رابط عضو مي فرستد.
٥. Print كردن اين ليست از موارد به صفحه نمايش عضو.

براي فعال كردن فرآيند رزرو، دكمه رزرو بايد براي هر مورد موجود در ليست فراهم باشد. با فشردن اين دكمه، توالي از مورد كاربردهاي ديگر هم فعال مي شوند. اين توالي شامل: مورد كاربرد رزرو كردن، مورد كاربرد انتقال رزرو و مورد كاربرد دريافت پيام مي باشند. مراحل هر يک از اين مورد كاربرد ها در جدول زير آمده است:

رزرو كردن

١. فرد عضو ليستي از موارد را دريافت مي كند (از طريق جستجو يا پيشنهاد)
٢. فرد عضو دكمه رزرو را براي موردي كه مي خواهد رزرو كند كليک مي كند.
٣. فرم تصديق رزرو ظاهر خواهد شد .
٤. فرد عضو اطلاعات موجود در اين فرم را پر مي كند.

انتقال دادن رزرو

١. كليک دكمه تائيد بر روي فرم رزرو.
٢. كتابخانه ديجيتالي درخواست رزرو را به database رزرو انتقال مي دهد .
٣. درخواست به database رزرو مي رسد .
٤. در صفحه نمايش عضو ،پيغامي مبني بر اينكه " درخواست فرستاده شده است " و "صبر كنيد تا تصديق شود " ظاهر مي شود .

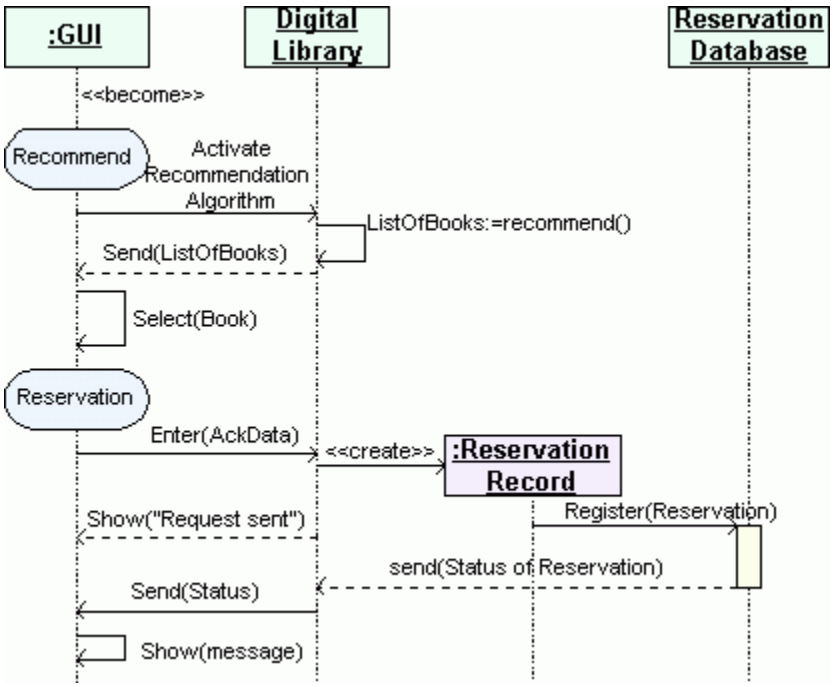
پيش از پرداختن به مورد كاربرد دريافت پيام، نگاهي به مورد كاربرد Package رزرو خواهيم داشت. مورد كاربرد هاي " رزرو جديد " و "انتقال پيام " در فرآيند تعاملات اين بخش جاي مي گيرند.

رزرو جديد

۱. درخواستي به database رزرو مي رسد و فرآيند ايجاد رزرو جديد، فعال مي شود.
۲. نتيجه اين فرآيند، وضعيت رزروي كه براي كتابخانه ديجيتالي فرستاده شده است مي باشد.
۳. فرستادن وضعيت رزرو به كتابخانه ديجيتالي.

ارسال پيام

۱. كتابخانه ديجيتالي وضعيت رزرو را به رابط عضو مي فرستد.
۲. موقعيت شناسايي ميشود.
۳. يک پيام بر روي نمايشگر عضو در ارتباط با شناسايي موقعيت رزرو نشان داده ميشود.



نمودار توالي براي روند رزرو كتاب پيشنهاد شده

همانطور كه ميبينيد برخي مراحل موردهاي كاربرد اصلاح شده است. اين اصلاحات مثالهايي از راهي كه يک فاز از يک پروژه ميتواند بر ديگري تاثير بگذارد است.
همچنين به تغييرات حالت بر روي اولين خط حيات توجه نماييد. اين مورد براي روشن سازي چگونگي استقرار رابط كاربر به جهت سرو كار داشتن با نوع ويژه اي از فعاليت در نظر گرفته شده است. ما ميتوانيم تمام تغيير حالات ممكن را بصورت نمودارهاي حالت بطور جداگانه قرار دهيم، اما اين كار زياده روي است. آنها را در نمودار توالي قرار دهيد تا اقتصادي تر ظاهر شوند.

تعاملات در سيستم – بخش دوم

فرض كنيد كه عضوي مي خواهد كتابي امانت بگيرد. پيش از هر چيز اين فرد بايد در كتابخانه حضور يابد، كتاب مورد نظر بايد به شكل كاغذي باشد و نمونه آزاد آن كتاب بايد در منبع كتابخانه موجود باشد. اگر تمام اين شرايط برقرار بود آنگاه نمونه اي از آن كتاب مي تواند به فرد عضو امانت داده شود.
براي اجتناب از رفتن به كتابخانه و جستجوي نمونه آزاد، عضو مي تواند رزروي براي سيستم كتابخانه ديجيتالي، انجام دهد و پس از دريافت اخطار آزاد بودن نمونه اي از موردي كه مد نظر دارد در كتابخانه، به كتابخانه برود و امانت گرفتن را انجام دهد .
بياييد به مورد كاربرد هاي Package كتاب كه براي اين عملياتها مورد نياز هستند، نگاهي بياندازيم :

Package كتاب

مورد كاربرد هاي Package كتاب

- انتخاب نويسنده
- دريافت دادهٔ نويسنده
- وارد كردن داده براي كتاب جديد
- ويرايش داده براي كتاب موجود
- وارد كردن دادهٔ مولتي مديا
- دريافت اخطار پس از وارد كردن دادهٔ مولتي مديا
- بررسي نمونه هاي آزاد
- دريافت تعداد نمونه هايي كه آزاد هستند
- بررسي رزرو
- دريافت تعداد رزرو ها
- يافتن قديمي ترين رزرو
- ايجاد رزرو
- دريافت اخطار براي رزرو
- شروع امانت گرفتن
- انتقال داده امانت گرفتن
- دريافت اخطار براي امانت گرفتن

براي فرآيند امانت گرفتن به اين مورد كاربرد ها نياز داريم: " امانت گرفتن " ، "انتقال داده امانت گرفتن"، " دريافت تعداد رزرو ها" " بررسي رزرو" و " دريافت اخطار براي امانت گرفتن ". بياييد به مراحل هر يک از اين مورد كاربرد ها نگاهي بياندازيم :

امانت گرفتن

۱. فعال كردن بخش امانت گرفتن يا برگرداندن امانت در رابط امانت دهنده.
۲. فرم امانت در صفحه نمايش امانت دهنده ظاهر مي شود.

٣. امانت دهنده، داده هاي ضروري را وارد مي كند.
٤. داده هاي امانت به كتابخانه ديجيتالي انتقال داده مي شود.

انتقال داده هاي امانت

١. امانت دهنده بر روي دكمه تائيد كه روي فرم امانت قرار دارد كليک مي كند و داده امانت به كتابخانه ديجيتالي فرستاده مي شود.
٢. كتابخانه ديجيتالي داده ها را دريافت مي كند و درخواستي براي امانت جديد به database امانت، فرستاده مي شود.
٣. درخواست به database امانت فرستاده مي شود و تصديقي دريافت مي شود.

وقتي داده به database امانت انتقال داده مي شود، مكانيزمي براي افزودن پرونده جديد در database فعال مي شود. اين عمليات با مورد كاربرد هاي موجود در Package امانت همكاري مي كند كه عبارتند از: مورد كاربردهاي "دريافت درخواست براي امانت جديد "، " اجراي امانت " و " انتقال نتيجهٔ امانت ".

وقتي امانت انجام مي شود نتايج مورد كاربرد Package كتاب نيز بدست مي آيند. اين فرآيند به نتايجي كه در طول رزرو توسط عضو براي كتابي ايجاد شده است، نياز دارد و اين امر در صورتي تحقق مي يابد كه نتيجه مورد كاربرد " دريافت تعداد رزرو " برابر صفر نباشد. بررسي در طول رزرو ضروري مي باشد زيرا اگر رزروي براي كتابي ايجاد شده باشد (نمونه اي از آن كتاب براي عضو نگه داشته مي شود) آنگاه هنگاميكه كتاب به امانت ميرود آن رزرو بايد از database رزرو حذف شود.

دريافت درخواست براي امانت جديد

١. database امانت، درخواستي براي اجراي فرآيند ايجاد پرونده جديد، دريافت مي كند.
٢. اين امرفرآيند ايجاد و افزودن پرونده جديد را در database فعال مي كند.

اجراي عمل امانت دادن

١. **database**، مكانيزم هايش را براي ايجاد پرونده جديد فعال مي كند.
٢. درنتيجه ايجاد پرونده جديد، مورد هدف قرار مي گيرد.
٣. اين نتيجه به كتابخانه ديجيتالي فرستاده مي شود.

انتقال نتيجه امانت

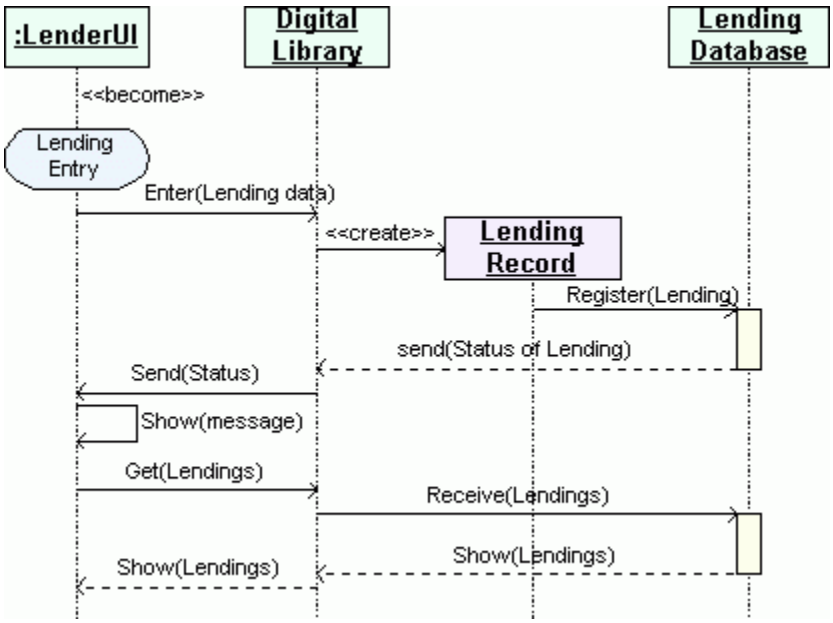
١. نتيجه امانت به كتابخانه ديجيتالي فرستاده مي شود.
٢. اين نتيجه به رابط امانت دهنده، انتقال داده مي شود.

حال مي توانيم مورد كاربردهاي Package كتاب را ادامه دهيم :

دريافت پيام امانت

١. رابط امانت دهنده، نتيجه امانت را دريافت كرده است.
٢. اين نتيجه رمز گشايي شده است و پيغامي بر صفحه نمايش نشان شده است.
٣. درخواستي از وضعيت جاري پرونده امانت عضو، به database امانت، فرستاده مي شود.
٤. ليستي ازپرونده ها، از database امانت گرفته مي شود.
٥. اين ليست در صفحه نمايش امانت دهنده نشان داده مي شود.
٦.

توالي نمودارها براي مراحل اين مورد كاربرد ها در شكل زير نشان داده شده اند:



نمودار توالي براي "امانت دادن"

چنانچه مي توانيد ببينيد برخي از مورد كاربردهاي داده شده در ليست براي اين هدف استفاده نشده اند، آنها در فعال سازي موارد موجود در database امانت بكار مي روند. عمليات ايجاد يک پرونده، با اين مورد كاربردها همكاري مي كند و پس از آن نتيجه بصورت خط چين پيام كه از فعال سازي خط حيات DB (database) امانت دادن خارج شده، منعكس ميشود.

به عنوان تمرين، سعي كنيد نمودار هاي توالي مورد كاربرد " بازگرداندن كتاب " از Package كتاب و مورد كاربرد " وارد كردن داده براي مجله جديد " از Package مجله، را رسم كنيد.

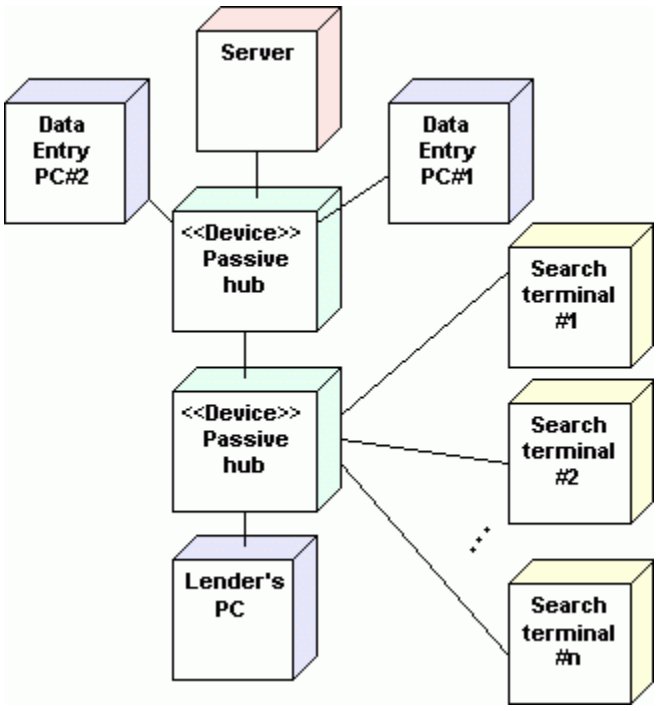
پس از آنكه تعاملات را از ميان اجزا طراحي كرديد، سيستم به حقيقت خود نزديک تر مي شود. همانطوركه تعاملات را طراحي مي كنيد، ممكن است دريابيد كه اصلاح اين مورد كاربردها بر اساس اين تعاملات بسيارمناسب است. نتايج اين تحليل، كدنويسي اشياء مختلف سيستم و چگونگي ارتباط آنها با بقيه را آسان مي سازد.

تحليل يكپارچگي با سيستمهاي مشترک

پس از آنکه مفاهيم کلي درباره سيستم کتابخانه ديجيتالي به دست آمد، مهندس سيستم درباره اينکه ساختار فيزيکي بايد چگونه باشد، خواهند انديشيد و توپولوژي شبکه هاي ديگر را بررسي خواهد کرد و اينکه چگونه آنها را تکميل کند و به انجام برساند واينکه هريک از اجزاي نرم افزار به چه node ي در شبکه تعلق دارند را به تصوير ميکشند.

تيم توسعه تصميم گرفته است که شبکه محلي کتابخانه در سيستم کتابخانه ديجيتالي قرار داشته باشد آن هم يک رابط اينترنت که بر روي سرور LAN اجرا شود.

بياييد در ابتدا به معماري فيزيکي شبکه محلي کتابخانه نگاهي بياندازيم:



نمودار استقرار براي LAN کتابخانه

كتابخانه شامل اجزاء زير مي باشد:LANچنانچه مي توانيد در اين نمودار استقرار مشاهده كنيد

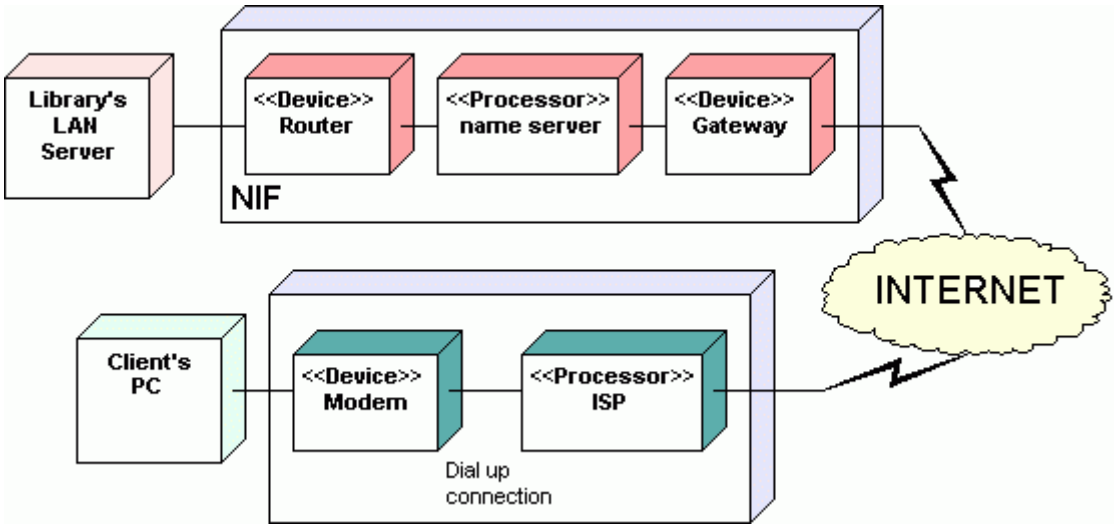
نام اجزاء	مقدار
سرور	١
PC (personal computer) وارد كننده داده	٢
PC امانت دهنده	١
پايانه جستجو	٦
hub غير فعال	٢

سرور و PC هاي وارد كننده داده در اتاق يكساني قرار دارند همچنين يک hub غير فعال نيز در اين اتاق قرار دارد. كابلي از اين hub غير فعال، از اتاق خارج و به راهرو مي رود . hub غير فعال ديگري وجود دارد كه از آن، كابل هايي خارج مي شوند كه به پايانه هاي جستجويي كه در راهرو قرار دارند، و PC امانت دهنده هايي كه در ناحيه امانت قرار دارند، مي روند.

حال كه معماري فيزيكي را پيش روي خود داريم، بياييد نگاهي به تكميل رابط اينترنت بر روي سرور بياندازيم: اين رابط از پايانه هاي جستجو، ازPC هاي امانت دهنده و PC هاي وارد كننده داده و از طريق اينترنت خانه يا اداره فرد عضو استفاده خواهند شد. بنابراين سخت افزار جديدي بايد در LAN كتابخانه پياده سازي شود.

LAN كتابخانه ساختار يكساني خواهد داشت. سخت افزار جديد فقط در كنار سرور اضافه خواهد شد و يک وسيله اتصال داخلي شبكه Network Interconnection Facility (NIF) است. NIF شامل موارد زير مي باشد: يک نام سرور(databaseي كه اتصالات را معتبر مي سازد)، يک router (دستگاهي براي اتصال شبكه به همديگر)، و يک مدخل (دستگاهي براي ترجمه اطلاعات از يک پروتكل ارتباطي به ديگري). مدخل به اعضا اينترنت اجازه دسترسي به كتابخانه ديجيتالي را از خانه يا اداره آنها ميدهد.

در كنار اعضا، فقط كامپيوتري با اتصال اينترنت و مرورگر اينترنت، نياز است. همچنين يک عضو بايد نام مستعار و رمز عبور داشته باشد تا بتواند به صفحه وب كتابخانه ديجيتالي، دسترسي داشته باشد. براي آنكه استقرار با مثال مشخص شود، مهندس سيستم نمودار استقرار نشان داده شده در شكل زير را تحويل داده است:



نمودار استقرار سیستم کتابخانه دیجیتالي

نگاه طراحي (GIU=Graphical User Interface)

مورد كاربردهاي بسياري را كه از تحليل برگرفته شده بودند پشت سر گذاشتيد.حال بايد به برخي از جنبه هاي تحليل سيستم كه با مورد كاربردها قابل تعقيب هستند، نگاهي بيانداريد و هر دوي آنها براي محصول نهايي بسيار مهم هستند. رابط هاي كاربر گرافيكي (GUI) ، قابليت استفاده سيستم را تعيين مي كنند. حال بياييد به تفحص در فرآيند طراحي GUI بپردازيم.

برخي اصول عمومي براي طراحي GUI

طراحي رابط كاربر، به طورمساوي ازهنر وعلم برگرفته شده است كه بر اساس خيال و بصيرت گرافيست، رسم مي شوند و يافته هاي پژوهشگران فاكتورهاي انساني و درک مستقيم كاربر بالقوه هستند.
برخي اصول عمومي براي طراحي GUI عبارتند از :

١. درک كنيد كه كاربر چه كاري بايد انجام دهد. طراحان رابط كاربر، تحليلي براي فهميدن ماهيت كار كاربران، انجام مي دهند. تحليل مورد كاربرد ما كاملاً با اين امر در ارتباط است.
٢. كاربر را وادار كند كه كنترل تعاملات برايش محسوس باشد. كاربر هميشه توانايي لغو كردن تعاملي را كه آغاز شده است را داشته باشد.
٣. كاربر، چندين راه مختلف براي انجام هر عمل وابسته به رابط (مانند بستن يک پنجره يا فايل) داشته باشد و خطاهاي كاربر به طور مطبوعي بخشيده شود .
٤. به علت تاثيرات فرهنگي، چشم ما به گوشه بالا سمت راست صفحه نمايش كشيده ميشود. اطلاعاتي را كه برتري بيشتري دارند در آنجا قرار دهيد.
٥. از ارتباط هاي فضايي سود ببريد. اجزاي صفحه نمايش كه به هم مربوط اند، نزديک يكديگر ظاهر مي شوند و ممكن است با جعبه اي كه دور آنهاست ظاهر شوند.
٦. به خوانايي و در ک كردن اهميت دهيد. از صداهاي فعال براي ارتباط ايده ها و مفاهيم استفاده كنيد .
٧. رنگ هايي را كه استفاده مي كنيد محدود كنيد. تعداد آنها را شديداً محدود كنيد. تعداد رنگ هاي زياد حواس كاربر را حين انجام وظيفه پرت مي كند.
٨. اگر به استفاده از رنگ ها براي تفكيک منظورهاي خود مي انديشيد، به ياد داشته باشيد كه هميشه براي كاربر آسان نيست كه وابستگي بين يک رنگ و منظور شما را ببينند.

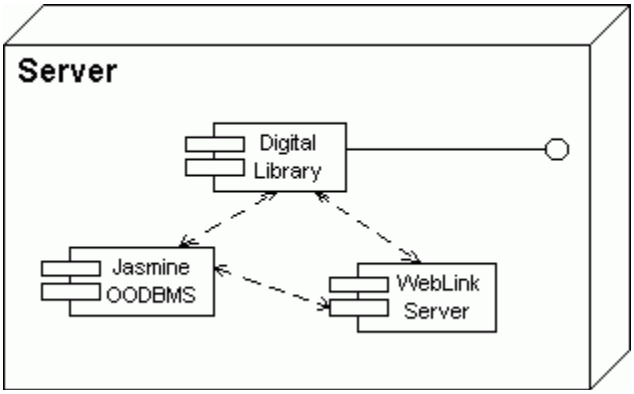
۹. اگر از موردي با رنگ استفاده مي كنيد، استفاده فونت ها را محدود كنيد و از فونت هاي ايتاليک و فونت هاي بيش از حد پرهيز كنيد.
۱۰. تا آنجا كه ممكن است سعي كنيد اجزا را درسايزهاي يكساني نگهداريد. اگر از اجزاء با سايز هاي متفاوت، رنگ هاي چندتايي و فونت هاي متغيراستفاده مي كنيد، آش شله قلم كاري ايجاد خواهيد كرد كه متخصصين GUI آن را يک طرح clown-pants مي نامند .
۱۱. اجزاء Right-align و فيلدهاي داده – آنها را بر اساس لبه هاي سمت راست مرتب كنيد. اين عمل زمانيكه كاربر بايد صفحه نمايش را به طور اجمالي مرور كند، حركات چشم را به حداقل مي رساند.
۱۲. زمانيكه كاربر بايد اطلاعات را بخواند و فرآيند كند و سپس دكمه اي را كليک كند، دكمه ها را در يک ستون در سمت راست اطلاعات يا در سطري در زيرو سمت راست آنها قرار دهيد. اين امر با تمايل طبيعي از راست به چپ خواندن سازگار است. اگر يكي از دكمه ها، دكمه قراردادي است آن را پررنگ كنيد و آن را اولين دكمه در مجموعه قرار دهيد.

نكاتي درباره استقرار كتابخانه ديجيتالي

وقتي تمام تحليلهاي مقتضي و طراحي بخش هاي مختلف تكميل شد، تيم نتايج آن را به صورت يک مستند طراحي ارائه مي كند و كپي هايي از آن را برايclient ها و برنامه نويسان نگه مي دارد. سپس برنامه نويسان طرح ها را به كد تبديل مي كنند، و پس از نوشتن كدها، كدها بر اساس آزمايشات دوباره امتحان مي شوند و باز نويسي مي شوند و اين فرآيند تا زمانيكه كد، همه آزمايشات را پشت سر بگذارد، ادامه خواهد داشت. تحليلهاي مورد كاربردها، اساس تستها را شكل مي دهند.

يكي از كتابخانه هاي ديجيتالي كامل شده كه پس از كارگاه امروز، خواهيد ديد در Jasmine OODBMS كدگذاري شده است. Jasmine به جهت اينكه يک زبان شئ گراست، امكان افزودن داده هاي مولتي مديا به database شما را دارد و به شما اجازه ايجاد يک نرم افزار مجزا يا نرم افزارهايي كه رابطهاي Web دارند را ميدهد، بكار ميرود. فايده ديگر استفاده از رابطهاي web اين است كه نرم افزار client بر روي دستگاه Client نصب نميشود و نرم افزار web به سكوي سخت افزاري كه روي آن راه اندازي شده بستگي ندارد.

سمت سرور، در نمودار استقرار ، كه به منظور توسعه كتابخانه ديجيتالي، توسعه داده شده است، مانند شكل زير است :



Node سرور با بخشي از اجزاي آن

متخصصان مستند سازي، مستند كردن سيستم را آغاز مي كنند و تلاش براي ايجاد يک مستند خوب ، مانند تلاش براي توسعه يک سيستم خوب، از برنامه بادقت، تحليل و آزمايش ناشي مي شود و ميتواند در فرآيند توسعه آغاز شود.

ايده اصلي طرح يک سيستم ، تمركز برتلاش هاي جدي تحليلها و طراحي مي باشد كه اجراي آن با چالش هاي كمي مواجه مي شود و نتايج پروژه، سيستمي است كه به طور تمام وقت با احتياجات client مواجه شود.

پرسش وپاسخ

به انتهاي اين دوره آموزشي رسيده ايم. به اين سوالات پاسخ دهيد و پاسخ خود را با پاسخ هاي صحيح كه در انتها آمده اند مقايسه كنيد.

١. براي هر سناريوي مورد كاربرد، به منظور اجراي تحليل مورد كاربرد، چه چيزي را مي خواهيم نشان دهيم؟
٢. بخش هاي مختلف يک نمودار مورد كاربرد چيست؟
٣. چرا مورد كاربردهاي اصلي، براي تسخير اختلافات جزئي در مكان اول با شكست مواجه ميشوند ؟
٤. در يک نمودار توالي چگونه مي توانيد يک فعال سازي را نشان دهيد و اين فعال سازي چه چيزي ارائه مي دهد؟
٥. كدام نمودار هاي UML براي توصيف رابط هاي كاربر مفيد هستند؟

پاسخ ها

۱. در هر سناريوي مورد كاربرد هدف نشان دادن موارد زير مي باشد :

- شرح مختصري از سناريو
- فرضيات سناريو
- actorي كه مورد كاربرد را آغاز مي كند.
- پيش شرط هاي اوليهٔ هر مورد كاربرد.
- مراحل وابسته به سيستم در هر سناريو.
- شرط هاي نهايي وقتي كه سناريو به پايان مي رسد.

۲. بخش هاي مختلف هر نمودار مورد كاربرد عبارتند از :
   - actor آغاز كننده
   - مورد كاربرد
   - actor سود برنده

۳. زيرا نتايج بخش JAD با كاربرهاي سيستم مي باشند نه توسعه دهندگان سيستم. شما تمام افزايش ها و تغييراتي كه وابسته به سيستم بودند را متوجه خواهيد شد نه وابسته به كسب و كار. پس از اتمام بخش هاي كاربران بالقوه و داشتن شانس تحليل مورد كاربرد ها اصلاحاتي مانند آنچه ظاهر شده ، غير معمولي نيستند.

٤. يک فعال سازي به صورت مستطيل پيكان مانند كوچكي بر روي شاهراه هاي اشيا نمايش داده شده است و در طول اجراي يک عمل دوره زماني را نشان مي دهد.

۵. نمودار هاي حالت و نمودار اجزا كه بخش هايي از نمودار هاي كلاس هستند زمانيكه رابط هاي كاربر طراحي مي شوند، مناسب هستند.

کارگاه

اين آخرين كارگاهي مي باشد كه انجام مي دهيد. لطفاً به اين سوالات پاسخ دهيد و مسايل را حل كنيد، فكر نمي كنم مشكلي داشته باشيد.

١. آيا اصلاح نمودار هاي مورد كاربرد و نمودار كلاس معمولاً رخ مي دهند؟
٢. چرا نمودار هاي همكاري در فرآيند تحليل كتابخانه ديجيتالي استفاده نمي شوند؟
٣. سه دليل براي محدود بودن رنگ در يک GUI بياوريد.
٤. مورد كاربردهاي Package عضو و كتاب را تحليل كرديم. سعي كنيد خودتان در نقش يک تحليلگر، تحليلي براي Package هاي كارمند و امانت دهنده انجام دهيد.
٥. سعي كنيد نمودار هايي براي مورد كاربرد " برگرداندن كتاب " از Package كتاب، و مورد كاربرد " وارد كردن داده براي مجله جديد " از Package مجله، رسم كنيد.