بسم الله الرحمن الرحیم

جزوه کلاسی دکتر سعید پارسا

# مهندسي نرم افزار ۱

## فرایند مهندسی نرم افزار شامل :

- شناخت
- طراحی
- پیاده سازی
- پشتیبانی

## شناخت

در مرحله شناخت نیازهای کاربر مشخص میشود. و در مرحله شناخت وارد یک سازمانی میشویم که بخواهیم آن را مکانیزه بکنیم اولین کاری که انجام میگیرد ابتدا بایدآن سازمان را شناسایی کنیم و به طور کلی نیازهای یک سازمان را آنالیز میکنیم. برای اینکه مرحله سناخت را آغاز کنیم از چارت سازمانی و عملیاتی استفاده میکنیم.ابتدا چارت سازمانی را مشخص میکنیم و بعد از روی چارت سازمانی ،عملیات ممکن را مشخص میکنیم.

مثال:

عملیات حذف و اخذ در دانشگاه را در نظر میگیریم و ابتدا عملیات را بررسی میکنیم .به طور کلی در مرحله شناخت نمیتوان تنها به گفته های مشتری قناعت کرد و برای این منظور مقوله ای تحت عنوان آنالیز وظایف مطرح می شود. از آنجایی که نمیتوان صرفا به گفته های افراد در بیان نیازها متکی بود. با شرح وظایف افراد می توان این مشکل را حل کرد و با شرح وظایف افراد که به صورت دنباله ای از activity مشخص میشود، میتوان با استفاده از این عملیات این نیازمندیها را مشخص کرد.

شناخت سیستم به دو روش انجام میگیرد:

- متمرکز بر سیستم
- متمرکز بر کاربر

شناخت متمرکز بر سیستم:

در ابتدا که روش ساخت یافته بود ،سیستمها batch بود یعنی یک سیستم وجود داشت و چندین ترمینال به آن وصل بود و افراد از طریق ترمینالها به آن وصل بودند و از این

طریق ارتباط برقرار میکردند. بعد سیستمهای pc ایجاد شد. پس در این قسمت با یک روش user centric میتوان این عملیات بازرسی را انجام داده و نیازهای آن را مشخص کرد. در صورتیکه کاربر به هر علتی نتواند پاسخگو باشد شرح وظایف بهترین ابزار برای نیازمندیها است.

گزارش شناخت نیازمندیها:

۱.نام سیستم(سیستم صندوق فروشگاه)

۲.مشتری سیستم (فروشگاهای زنجیره ای )

۳.هدف سیستم که رابطه ای منطقی است بین آنچه که در حال وجود دارد و آنچه که در آتیه به وجود خواهد آمد.

مثال:

برای این حالت مثالی که میتوانیم بزنیم این است که در یک فروشگاه صدها گونه کالا وجود دارد پس در این میان ممکن است که کنترل موجودی انبار با مشکل مواجه شود و عمل کنترل موجودی انبار با مشکل مواجه میشود و عمل کنترل موجودی بدرستی انجام نشده و به هدف ممکن نمیرسیم .پس هدف این است که با ایجاد یک سیستم مکانیزه عمل ثبت و به طور کلی عملیات فروشگاه به طور اتوماتیک انجام گیرد.

۴.شرح مسئله:

۱.حوزه مسئله: افرادی که در خارج از سیستم قرار گرفته اند ،و یک سری سیستم هایی است که در خارج از یک سیتم قرار گرفته اند و در تعامل با یکدیگر هستند و این توضیح مسئله را مشخص میکند.

۲.نیاز مسئله:چه افرادی خواستار این سیتم هستند.

۳.راه حل پیشنهادی:سیستم جه راه حلی را در اختیار کاربر قرار میدهد.

۴.نقطه فروش:وجه تمایز یک سیستم با سیستمهای دیگرمیباشد.

اولین بخش در تجزیه و تحلیل ،شناخت نیازها میباشد.و این شناخت به دو صورت میباشد:

۱متمرکز بر سیستمsystem cenrtic

۲.متمرکز بر کاربر user cenrtic

اگر سیستم کامپیوتری موجود باشد دیگر پس از تعیین نیاز،کاری به چارت سازمانی و عملکرد افراد نیست. معمولا با تبدیل ورودیها یا خروجیها به فرم دیگر کار آغاز میشود. اما باز هم نیاز ها باید تعیین شود. در اینجا افراد و نیازهای آنها هدف است. اما نیازهای کاربر را به چه ترتیبی میتوان مشخص کرد؟معمولا برای اینکه بتوانیم سیستم جدید کامپیوتری را ایجاد کنیم که پاسخگوی نیاز کاربر ها باشد

۱.اهداف کلی سیستم را بدانیم.

۲.اهداف کاری افرادی را که در سیستم فعال هستند را بدانیم.بنابراین باید سیستم کلی را تا حدی شناخت. باید تعیین کرد که در حال حاضر در ارتباط با انجام کارها افراد دارند و چه انتظاراتی از سیستم جدید خواهیم داشت پس ۲ نکته مطرح میباشد:

۱.سیستم جاری

۲. سیستم آتی

سیستم جاری

ممکن است این سیستم دستی باشد و ممکن است که کامپیوتری شده باشدو یا ترکیبی از این دو باشدبنابراین لازم میباشد که سیستم کنونی مورد شناخت قرار دهیم . سیستم شناسانده میشودتا:

۱.تا نیازها برای سیستم آتی تا حدی مشخص میشود.

۲.نقصها و تعمیدها در سیستم کنونی مشخص میشود.

۳.کارهای زائد و اضافه تعیین گردد.

در سال ۱۹۸۹ ادوارد یوردون اعلام کرد که شناخت جزئیات سیستم کنونی فقط اتلاف وقت است.یوردون اعلام کرد که نباید وقت و هزینه زیادی را صرف بررسی و مدل سازی سیستم موجود کرد به خاطر اینکه در این صورت آنالیست اهداف را گم میکند.و اهداف مهم را شناسایی نمیکندو به این ترتیب پروژها با شکست مواجه میشود و یوردون تاکید کرد که بهتر است در ابتدای کار تحقیق خود را بر روی سیستم جدید قرار دهد. روشهای ساخت یافته مانند ssadm زمان زیادی را برای مدل سازی سیستم جاری انجام میدادند. تا اینکه سیستم جدید را شناسایی میکنند منظور سیستمی که در آتیه با استفاده از کامپیوتر ایجاد کنند البته اعتقاد ما بنیانگذاران uml و صاحبان متدولوژی usdp به این است که سیستم کنونی باید مورد مطالعه قرار بگیرد زیرا اعتقاد آنها :

۱. بعضی از عملکردها و یا در اصطلاح functionality سیستم میباید در سیستم جدید مکانیزه وجود داشته باشد .
۲. برخی از اطلاعات سیستم جاری مسلما به سیستم جدید انتقال داده میشود.
۳. در صورتیکه سیستم جاری مکانیزه باشد یا بعضا مکانیزه باشد باید مستندات سیستم جاری به سیستم جدید انتقال داده سود و الگوریتمها استخراج شود.ممکن است که سیستم فعلی نقصهایی داشته باشد که این نقصها باید برطرف شود.
۴. مطالعه سیستم کمک میکند تا به طور کلی سازمان را بشناسیم.
۵. برخی از عملیات جاری در سیستم آتی نیز باید منعکس شود.
۶. معمولا سعی میکنیم تا کار افراد را درک کنیم تا اینکه بتوانیم سیستم مناسب کامپیوتری را برای آنها ایجاد کنیم.
۷. میبایست با مطالعه سیستم جاری را بتوانیم میزان کارایی سیستم آتی را مشخص کنیم.
۸. به دلایل فوق برخلاف گفته یوردون ما تشخیص میدهیم که باید تا حدی سیستم جاری مورد شناخت قرار گیرد.

## نیازها

### اصولا نیازها در ارتباط با سیستم جدید مطرح میشود و به سه دسته تقسیم میشود:

۱. نیازهای عملیاتی
۲. نیازهای کیفی
۳. نیازهای قابل استفاده بودن

### نیازهای عملیاتی:

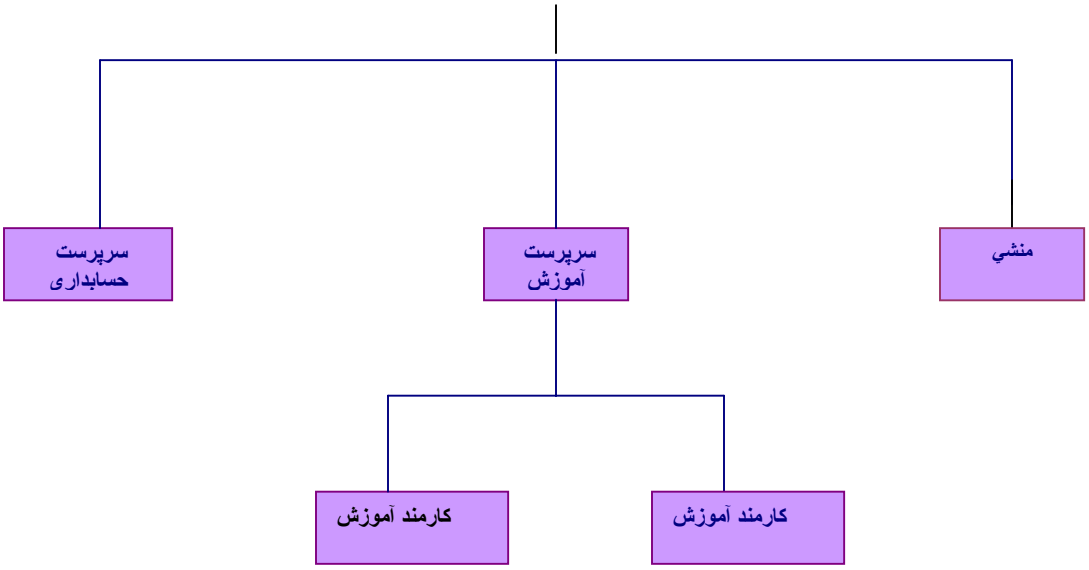
۱. مشخص میکنند که سیستم چه کار باید انجام دهد و یا چه انتظاراتی از سیستم میرود.

۲. برای این منظور در دیدگاه شی گرا در مورد های استفاده از سیستم یا سرویسهای آن مشخص میشود و سپس همچنانکه کار آنالیز پیشرفت میکند از بطن مورد ها و یا سیستم ها عملیات مورد نیاز مسخص میگردد. و بنابراین مشخص میگردد. و بنابراین عملیات سیستم همانند روشهای ساخت یافته در ایجاد سیستم جدید مبنایی میباشد . در دیدگاه شیگرا همانند دیدگاه ساخت یافته ابتدا چارت سازمانی مشخص شده و سپس جارت عملیاتی مشخص شده و بعد از این مراحل عملیات مورد نظر تعیین گردیده و این عملکردها آغاز میشود.

مثال:

در چارت سازمانی واحد آموزش دانشگاه داریم:

۱. واحد آموزشی
۲. سرپرست آموزشی




ما باید توجه داشته باشیم که عملکردها توسط افراد مختلف به انجام رسیده است و ا بدین واسطه چارت عملیاتی تا حدی متفاوت از چارت سازمانی است.

۱. اصولا نیازهای عملیاتی شامل توصیف فرایندهای عملیاتی است که سیستم آتی باید انجام دهد.

۲. جزیات ورودیها به سیستم از طریق فرمهای ورودی و یا فرمهای کاغذی ،مستندات ،ارتباط با افراد و سایر سیستمها و یا مکالمات تلفنی .

۳. جزئیات خروجیهایی که سیستم باید فراهم نماید ،خروجیها در قالب گزارشات ،فرمهای خروجی و اطلاعاتی که باید به خارج از سیستمها انتقال داده شود مشخص گردد.

۴. جزئیات داده هایی که در داخل سیستم حفظ و نگهداری شود.

## نیازهای کیفی:

در واقع در سیستمهای غیر عملیاتی مشخص میشود.کیفیت نیازهای عملیاتی باید باید به جه صورتی باشد؟در واقع توصیفی از دیدگاهای عملیاتی میباشد.نیازهای کیفی شامل:

۱. معیارهای کارایی؛ مانند زمان پاسخ گویی

(مثلا هر کارمندی در ساعت معینی کارت میزند،حداکثر زمانی که کارمند منتظر میماند ۵ دقیقه است)

۲. حجم داده ها در قالب زمان مورد پیش بینی برای ذخیره و بازیابی

۳. نیازهای امنیتی در قالب کنترل دسترسی به داده ها وامکانات

۴. سیستم یا محیط اجرایی

۵. تحمل خرابیها (مثلا دستگاه کارت خوان خراب هست پس باید کیبورد پشتیبانی کند. یعنی تحمل خطا میکند.)

### نیازهای قابل استفاده بودن:

این دسته از نیازها در واقع مشخص میکند که سیستم جدید با وظایف کاربر مطابقت دارد یا نه.

برطبق استانداردهای موجود مفهوم مشخص میشود که عبارت است از:

۱. کاربر میتواند خواسته های خود را در یک محیط به میزان قابل قبول و موثری تحت کنترل برآورده سازد.به عبارت دیکر قابلیت استفاده مطابقت میدهد و خواسته های کاربر با آنچه که سیستم جدید در اختیاری قرار دهد.معمولا مقوله (رابطه بین انسان و

کامپیوتر) در ارتباط با این مرحله مطرح میشود.برای اینکه بتوانیم از آغاز قابل استفاده بودن سیستم جدید را مد نظر داشته باشیم میباید اطلاعات زیر جمع آوری کنیم.

۱. ویژگیهایی کاربر که سیستم را مورد استفاده قرار میدهد.

۲. وظایفی که کاربر بر عهده داردو اهداف مربوطه

۳. موقعیتهایی که سیستم ممکن است تحت آنها مورد استفاده قرار بگیرد.

۴. معیارهایی که براساس آنها قابل قبول بودن سیستم مورد سنجش قرار میگیرد.

روشهای استخراج نیاز:

اصولا پنج روش برای تعیین نیازها رایج میباشدکه عبارتند از :

مطالعه اولیه

مصاحبه

مشاهده

ارجاع به مستندات

تهیه پرسش نامه

## مطالعه اولیه

برای این منظور معمولا یک آنالیست که سیستم را به صورت دستی میشناسد، اهداف کلی و کاری آن را میداند، به استخدام در میآورد. وظیفه این فرد این است که سازمان را مورد شناسایی قرار دهد و مستنداتی در ارتباط با گزارشاتی که توسط افراد در این کمپانی تهیه میشود، چارت سازمانی، سیاستهای کاری، شرح وظایف، گزارشات و مستندات سیستم جاری تهیه نماید. بدین ترتیب اهداف سازمان مورد نظر مشخص میشود و تا حدی نیازها از سیستم آتی تهیه میشود.

مثلا سیستم حسابداری را در نظر بگیرید. معمولا از فردی که در این مورد آگاهی دارد استفاده میکنند و از آنها استفاده میشود. مزایا ومشکلات:

۱. مطالعه اولیه به آنالیست کمک میکند تا درکی از یک سازمان داشته باشد؛ قبل از اینکه با افراد روبه رو شود. بنابراین آنالیست با چشم باز و اطلاعات قبلی به سراغ افراد میرود.

۲. برمبنای اطلاعات جمع شده آنالیست میتواند سوالات خود را مشخص کند و با سوال به سراغ افراد برود.

۳.مستندات سیستم جاری کمک میکند که آنالیست تا حدی نیازها را مشخص کند.

۴.معمولا مستندات سیستم با واقعیتها متفاوت میباشد و آنالیست باید آنها را جویا شود.

## مصاحبه

واضح است که برای شناسایی یک سیستم باید با افراد به مصاحبه پرداخت. در این بخش راهنماییهایی در ارتباط با چگونگی انجام مصاحبه ارائه میشود. قبل از مصاحبه میبایست وقت ملاقات مشخص شود و مدت ملاقات باید تعیین شود، موضوع ملاقات باید مشخص شود. مسلما زمان مصاحبه یک اتلاف وقت برای یک مصاحبه شونده میباشد.که باید این نکته در نظر بگیریم و باید سعی بشود با ارجاع به مدیر و تایید آنها افراد با تجربه را انتخاب نماییم. در پروژههای بزرگ معمولا یک زمان بندی برای مصاحبه ها مشخص میشود و تعیین میشود که چه افرادی باید برای چه مدتی مصاحبه شوند. اهداف برای هر مصاحبه مشخص میشود و در ابتدا سولات را مینویسند و پاسخها را برای هر سوال مشخص میکنند و مطمئن شوید که سوالات با توجه به شرح وظایف مصاحبه شونده تهیه میشود. لذا قبل از مصاحبه باید شرح وظایف افراد را مطالعه کنید و در ضمن مصاحبه باید هدف را مشخص کنید و اجازه ندهیدکه مصاحبه شونده خارج از موضوع صحبت کند و یا اینکه کنترل مصاحبه را در اختیار بگیرد. باید سوالات در ارتباط با چگونگی استفاده از کامپیوتر در ارتباط با انجام وظایف مصاحبه شونده باشد. باید سعی شود که با تهیه مثالهایی از چگونگی تهیه مستندات و پر کردن فرمها کار افراد مستند نماییم. بعد از مصاحبه باید گزارش مصاحبه را به دقت تهیه نماییم و باید سعی کنیم که به حافظه نسپاریم بلکه بر روی کاغذ مستند کنیم. در مصاحبه ها معمولا اطلاعات در مورد افراد و وظایف کاری افراد گرد آوری میشود. نیازها در ارتباط با سیستم ارتباطی جدید مطرح میگردد. مستندات و فرمهای مورد استفاده جمع آوری میشود. پرسش نامه ها در اختیار مصاحبه شونده در صورت نیاز قرار میگیرد. سعی کنید که با افراد مختلف در یک مورد یک

خاص مصاحبه نمایید تا تناقضها مشخص گردد و در برخی از روشها افراد را در یک جلسه گرد آوری میکنند و در واقع در یک زمان با چند نفر مصاحبه میشود و حاصل این بحثها اهداف سیستم آتی را مشخص میکند.

مزایا ومشکلات :

۱ . ارتباط با افراد موجب می شود آنالیست پاسخگوی نیازهای افراد باشد و خود را با آنها وفق دهد .

۲ . آنالیست ممکن استروش های دیگری را که می توان کار افراد را ساده تر انجام داد مشخص نماید .

۳ . مصاحبه مشکلی که دارد هزینه براست .

۴ . تنها انجام مصاحبه کافی نیست . آنالیست باید بر روی نتایج مصاحبه کار کند ومعمولاً در قالب تهیه فرمهای ورودی خروجی و منو ها باشد وآنالیست سعی می کند محیط کاری آتی را به کاربر نشان دهد .

۵ . مشکل مصاحبه ها این است که وابسته به نکته نظرهای مصاحبه شونده است .

## مشاهده

با مشاهدهٔ چگونگی انجام وظایف شاید بتوان بهتر ازمصاحبه بر روی سیستم و یا نیازهای آن شناخت پیدا کرد . باید دقت کنید که مشاهده بسیار طولانی می شود و باید هدف را قبلاً مشخص کرد . معمولاًمشاهده در ارتباط با جنبه های مختلف انجام یک کار می باشد . این شامل مدت زمان انجام وظیفه توسط فردوتعداد خطاهائی که فرددرانجام کار خود ایجاد می کند . فاکتورهائی کهدر ارتقاء کارائی کار افراد مهم هستند باید مشخص شوند . به طورخلاصه مزایا ومشکلات به شرح زیر است :

۱ . آشنائی به روشی که سیستم جاری کارمی کند .

۲ . میزان کارائی سیستم جاری را می توان مشخص نمود .

۳ . میزان کارائی سیستم جاری را می توان مشخص نمود .

۴ . مشاهد کننده باید فردی با تجربه باشدتاهرچیزی را مورد تأکید قرارندهد .

مشکلی که وجود دارد این است که :

افراد هنگامی که احساس می کنندتحت نظر هستند ، طبق روال عادی خود عمل نمی کند .

## ارجاع به مستندات

معمولاً باارجاع به فرمها وگزارشاتی که افراد تهیه می کنند و بخصوص درهنگام مصاحبه می توان بخش اعظمی از نیازها را مشخص کرد . با ارجاع به این فرمها میزان داده ها و فرمت دادها و نهایتاًساختار بانکهای اطلاعاتی وگزارشاتی که باید از آن استخراج شود مشخص می گردد .

### مزایا :

۱. حجم داده ها و اطلاعات مورد پردازش مشخص میگردد.

۲. فرمت داده ها تعیین میشود.

۳. مشکلات و خطاهایی که به واسطه پر کردن دستی مستندات ایجاد میشود تعیین میگردد.

۴. اگر سیستم کلا تغییر پیدا کند در این صورت مشکل به این صورت خواهد بود که مبادرت به گردآوری اطلاعاتی میکندکه ممکن است در آینده نیازی به آن نباشد.

## پرسش نامه

شامل مجموعه ای از سوالات هستند که توسط مشاورین و کسانی که با اهداف و عملکرد دستی سیستمها آشنایی دارند تهیه میشود و در یک سازمان پخش میشود وجوابها مطالعه میگردد.

### مشکلات و مزایا:

۱. با استفاده از پرسش نامه ها میتوان تعداد افراد زیادی را در بر گرفت.

۲. بر مبنای پاسخها میتوان نیازها را تعیین کرد.

۳. مشکل این است که تهیه پرسش نامه نیاز به تجربه زیاد دارد.

۴. مشکل دیگر عدم دریافت به موقع پاسخ از افراد است.

در ارتباط با تعیین نیازها برای هر شرکتی که کار آن تهیه تبلیغات میباشد؛ یکی از اولین وظایف در این مورد، تهیه یک برنامه کاری است که مشخص میکند چه اطلاعاتی باید گردآوری شود و چه تکنیکهایی باید مورد استفاده قرار بگیرد و چه افرادی برای چه مدتی درگیر کار باشد.

مثال:

یک برنامه ریزی در ارتباط با شرکت تبلیغاتی برای یافتن واقعیت به صورت زیر است:

| هدف | روش | موضوع | مدت |
|---|---|---|---|
| ۱.گردآوری اطلاعات اولیه در ارتباط با کمپانی و کلا صنعت تبلیغات | مطالعه اولیه | گزارشات کمپانی،مجلات تبلیغاتی | نصف روز |
| ۲.تعیین اهداف کاری و چگونگی ارتباط با شعب کمپانی تبلیغاتی | مصاحبه | دو نفر از مدیرها | دو روز |
| ۳.تعیین نقش هرواحدکاری،تعیین ساختارهای تیمی تعیین مصاحبه شونده ها ازبین کارمندان | مصاحبه | مدیر بخشها | دو تا یک ساعت |
| ۴.تعیین هسته کار | مصاحبه | مدیربخش طراحی، مدیربخش حسابداری | هر کدام به مدت یک ساعت و نیم |
| ۵.ادامه تکمیل برداشتهای قبلی از چگونگی انجام هر کار | مشاهده | دو نفر از افراد و یا کارمندان | نیم روز برای هر کدام |
| ۶.تعیین نقش افراد در ارتباط با هسته های کاری | مصاحبه | دو نفر از کارمندان براساس تجربه کاری انتخاب شوند. | هر کدام یک و نیم ساعت |
| ۷.برای تعیین ،و سوابق پروندههایی که نگهداری میشوند | مصاحبه | کارمند بایگانی،انباردار | ۲ تا ۱ ساعت |
| ۸.تعیین نیازهای حسابداری برای | مصاحبه | ۲ مدیر پخش و ۳ | هر کدام ۱،۵ ساعت |

| | | | |
|---|---|---|---|
| سیستم | | کارمند | |
| ۹.تعیین استفاده سیستم جاری و تعیین وظایف سیستم کنونی | مصاحبه | حسابدارو صندوق دار،کارمند بخش خرید،کارمند بخش حسابداری | ۳ ساعت |

## مستند سازی نیازها:

در این محیط با استفاده از ابزاری مورد نظر نیازها را مستند میکنند و به محض اینکه واقعیات گرد آوری میشود، میبایست اطلاعات را مستند کند. معمولا ابزار برای ایجاد مستندات به کارمیرود. البته وابسته به تجربه شرکتها ممکن است اطلاعات بیشتری برای مستند کردن نیازها لازم باشد که در چهار چوب نگنجد، معمولا مستندات اطلاعاتی در مورد حاصل مصاحبه ها و مشاهدها و شرح مشکلات ،کپی هایی از مستندات مورد استفاده (فرمها و گزارشها)شرح نیازها ، جزئیات استفاده کننده ها و بالاخره خلاصه ای از ملاقاتها میباشد. معمولا این مستندات در اختیار کاربر نیز قرار داده میشود و به تایید وی رسانده میشود. نیازها را در بانکهای اطلاعاتی نگهداری میکنند تا امکان ردیابی نیازها در طول انجام پروژه ها فراهم گردد.

## بر طبق دسته بندیهایی که انجام شده است نیازها شامل:

۱.عملیاتی

۲.قابل دسترسی

۳.قابل اطمینان

۴. قابل کارایی

۵. قابل پشتیبانی

در ارتباط با نیازهای عملیاتی قبلا بحث شده است.همان نیازهایی است که شاید به غلط امنیت در این مجموعه ذکر شده است. در صورتیکه امنیت جزو نیازهای کیفی است.قابلیت اطمینان آن نیازهایی است که مقدار قابل دسترسی به نیاز را بررسی

میکند.یعنی مقدار کمکی که وجود دارد بحث میشود.و سیستم باید در این صورت دارای زبان پاسخگویی داشته باشد.در ارتباط با این مورد فاکتور های انسانی بخث میشود.HCI
:یکی از عواملی که باعث این نیاز میشود یعنی برای فرد چه چیز قابل استفاده است .پس در ارتباط با مقوله قابل استفاده مسئله فاکتور انسانی مطرح میشود.و این مسئله بسیار مهم میباشد.یعنی وابسته به وضعیت فرد چه نوع امکاناتی در اختیار سیستم قرار بگیرد.فاکتورهای انسانی وضعیت فرد است.اگر فرد مدیر است آن نیاز دارد اطلاعات به صورت منحنی به آن داده شود.کسی که درگیر کار احتیاج دارد.جزئیات را داشته باشد.

## مفهوم فاکتور انسانی:

۱. یعنی انسان و یا کاربر مورد نظر مدام کار کند و یا چند وقت به چند وقت کار میکند و باید کاربرهای سیستم دقیقا مشخص میشود و سیستم در چه صورت قابل استفاده خواهد بود. مثلا مدیر و پزشک را در نظر بگیرید در این موقع مدیر در صورتی سیستم آن موقعی قابل استفاده خواهد بود که تمام اطلاعات را در اختیار داشته باشد و کاربر در صورتی مورد استفاده خواهد بود که به اندازه کافی اطلاعات داشته باشد.

## قابلیت اطمینان:

مثلا یک کارت ساعت را در نظر بگیرید وقتی خراب میشود باید کلا سیستم از تمامی اطلاعات پشتیبانی کند. پس باید در تمامی برنامه ها اطلاعات را در نظر بگیریم. پس نیازهای امنیتی باید در داخل سیستم قرار بگیرد.

## قابلیت بازیابی:

پس در سیستم قابل اطمینان اگر سیستم به خاطر برق گرفتگی از بین رفت آیا آن وضعیت قابلیت بازیابی دارد.پس به هر علتی سیستم قابل بازیابی را داشته باشیم.

## قابلیت کارایی:

از جمله نیازهای کیفی میباشد.

## قابلیت پشتیبانی:

معمولا شرکتهای کامپیوتر از پشتیبانی سیستم استفاده میکنندو در نتیجه سیستم پشتیبانی احتیاج دارد.

مفهوم دیگری که در این مورد نیازها بحث میشودکه به آن میپر دازیم:

### ۱. قابلیت تست

امکاناتی داشته باشد که سیستم را چک کند و تست کند و ایرادهای سیستم را رفع کند.پس در آن set up میگذاریم. و اگر بر روی آن کلیک کنیم یک سری گزارشات در آن وجود دارد.

### ۲. قابلیت توسعه

سیستم باید قابلیت توسعه داشته باشدو سیستم دارای تغییرات باشد.

### ۳. Adaptability

مثلا برای یک انبار سیستم بنویسم .سیستم انبار به صورت کلی بود یعنی این سیستم برای بقیه مطابقت داشته باشد.بعضی از سیستم ها یک سری امکاناتی دارند که مثلا از انبار میخواهیم که به وسایل در هر قسمت دسترسی پیدا کنیم یعنی با توجه به آدرس داده شده وسایل را پیدا کنیم و علامت زده میشود و بسته مورد استفاده قرار میگیرد.

## اولین دیدگاه( سیستم دانشگاه):

### ۱.هدف

هدف از این مستند تعیین نیازها به صورت کلی برای سیستم ثبت نام دانشگاه میباشد. نیازهایی که بر اساس نیاز کاربرها مشخص شده است. اولین چیزی که مشخص میشود هدف است.

### ۲.دامنه

هنگامی که دامنه یک سیستم و یا عملکرد یک سیتم را مشخص میکنیم باید موجودیتهای که در خارج از سیستم است را مشخص کنیم که سیستم ثبت نام این امکان را به دانشجویان میدهد که مستقیما ثبت نام کند. اساتید با استفاده از کامپیوتر دروسی را میخواهند ارائه دهند و مشخص کنند و نمره دانشجویان را وارد کنند. در این حال باید

سیستم کاتالوگ درسها را در بانگ اطلاعاتی خود را نگهداری کند . به عنوان بروشور در دانشگاه قابل دسترسی باشد.بنابراین باید دامنهٔ مشخص شودو اینکه چه نیازهایی دارد و چه کسانی چه امکاناتی را دارند.

### ۳.مستندات :

در این بخش مستنداتی که درجهت ایجاد سیستم به آنها ارجاع میشود مانند مستندات :شرح وظایف سازمانی و روند کاری که اغلب سیستم ها رایج است و بخصوص سازمانهایی که مدرک دارند.و به طور دقیق این مستندات را تهیه میکنند و در این جا به آن مستندات ارجاع شده و برای نیازها ،ارجاع میشود.برای نمونه در مورد سیستم ثبت نام مستندات زیر استفاده شده است.شرح عملیات ثبت نام،شماره مورد نظر و دفتر آموزش

### ۴.موقعیت :

موقعیت کاری:مثلا یک سیستم را تولید میکنید که بدون استفاده از کاغذ میباشد.پس در این قسمتها ،سیستمهای ما دستی و کاغذی میباشد.پس از طریق کامپیوتر عملیات انجام میشود.پس نامه مورد نظر به صورت کامپیوتری عمل میکند.در واقع مشخص میکند که هنگامی که سیستم کامپیوتری جایگزین سیستم فعلی شد چه شرایط کاری و چه شرایط تسهیلاتی ایجاد خواهد شد.برای مثال در ارتباط با سیستم ثبت نام به این ترتیب صحبت میشود که از این به بعد ارتباط افراد با واحد ثبت نام مستقیما وجود نخواهد دئاشت:اساتید و دانشجویان به صورت روشن و از طریق سیستم کامپیوتری عملیات خود را انجام میدهد.سیستم فعلی که در سال ۱۹۸۵ ایجاد شده است.با در نظر گرفتن تعداد دانشجویان و تعداد درسهایی که در سال ۲۰۰۰ ارائه خواهد شد.بسیار کند و نمیتواند پاسخگو باشد و سیستم جاری برروی یک قاب اصلی که از رده خارج شده است قرار گرفته است.اما در سیستم جدید از طریق شبکه و یا کامپیوتری شخصی ،اساتید و دانشجویان میتوانن به امکانات سیستم دسترسی پیدا کنند.و به این ترتیب ضاهر بسیار بهتری برای دانشگاه ایجاد میشود و دانشجویان بیشتری جذب دانشگاه خواهدشد.

### شرح مسئله:

در این جا مشکل و اثرات سوء و مزایای آن و سیستم کامپیوتری مطرح میشود.مشکل در سیستم کامپیوتر کند بودن فرایند ثبت نام میباشد که دانشجو و استاد و

اداره آموزش را تحت تاثیر قرار داده و از طرف سوء آن در قالب هزینه های زیاد ثبت نام است که در دانشگاه متقبل میشود و نارضایتی استاد و دانشجو.و مزیت آن از بین رفتن این مشکلات و ارائه جلوهای بسیار خوب است از دانشگاه میباشد.

## شرح کاربرها:

در این قسمت شرح کاربرها و نیازهای آنها مشخص میشود. کاربرهای اصلی مشخص میشود و یکی از استفاده کننده های اصلی سیستم، استاد و تعداد دانشجو است. برای هر استفاده کننده نیازها مشخص میشود. برای هر کاربر شرحی در ارتباط با سرویسی که کاربر از سیستم میگیردمشخص میشود. برای مثال در مورد دانشجویان به این ترتیب که دانجو باید فرم ثبت نام را بگیرد۲ هفته طول میکشدکه واحد ثبت نام پاسخ دهد و در این مدت ممکن است بواسطه عدم برگذاری برخی از کلاسها و یا اضافه شدن کلاسها برنامه ریزی تغییر کند. بنابراین دانشجویان نیازمند هستندکه بتوانند مستقیما ثبت نام کنند و نمرات خود را سریعا ببینند، بنابراین برای هر دسته از کاربرها نیازها در این بخش مشخص میگردد.

## توصیف محصول نرم افزاری

در این قسمت در قالب یک دیاگرام متنی این کار را انجام میدهیم. توصیفی از نرم افزاری که قرار است به مشتری تحویل داده شود یعنی قابلیتهای نرم افزار و ویژگیهای آن هزینه تولید و نکات مربوطه در این قسمت تعیین میشود. برای مثال در مورد سیستم ثبت نام امکانات سخت افزاری و نرم افزاری مشخص شده و در واقع شمایی از شبکه تعیین شده و قابلیتها در ارتباط با امکاناتی که برای ثبت نام ایجاد میشود و مشخص گردیده است.

## هزینه ها

مقدار بسیار پیچیده ای است.فرایند تولید نرم افزار شامل چه مراحلی است؟برنامه نویسی به عنوان یک پروژه میباشد.و باید تمام مراحل را مشخص کنیم و باید برای هر قسمتی باید هزینه را مشخص کنیم . پس در ارتباط با هزینه ها در مجموع باید مشخص میشود که هزینه انجام پروژه چقدر خواهد شد و درمسئله آنالیز، تمام مسائل برای

پروژههای برنامه نویسی مطرح میشودو سایر نکات را میتوانیم برای چشم انداز با مستندات مربوطه مشخص کنیم.

## چرخه حیات و یا فرایند تولید نرم افزارشامل ۴ مرحله میباشد:

۱. شناخت اولیه

۲. مرحله تشریح

۳. ایجاد

۴. انتقال

با توجه به شکل صفحه ۳۹۴ کتاب مهندسی نرم افزارمیتوانیم به این نتایج برسیم:

همانطور که در دیاگرام فوق مشخص شده است برای تولید نرم افزار عملیات مدل سازی سیستم جاری تعیین نیازها و تجزیه و تحلیل و پیاده سازی ،آزمون و نهایتا نصب نرم افزار در هر مرحله تکرار میشوداین بدین معنی است که از همان ابتدای کار درگیر کار میشوید و نباید پیاده سازی را به تعویق انداخت.اما برای هر مرحله هر یک از عملیات ،تاثیر متفاوتی دارند.برای نمونه در هر مرحله شناخت اولیه ،تاکید بسیاری بر مدل سیستم جاری و یا در واقع مدل سازی سیستم کاری است و پیاده سازی تاکید بسیار کم است.

در مرحله شناخت اولیه نهایتا قالبندی کاری مشخص میشود و عملکرد پروژه آغاز میشود.منظور در واقع میزان بازدهی پروژه از لحاظ مالی است یعنی نهایتا در مرحله شناخت اولیه هزینه مشخص میشود.و اگر هر تومان برای هر پروژه هزینه میکند چند برابر منفعت میکندبرای اینکه دامنه مشخص شود باید بازیگر ها مشخص شود و بازیگر در واقع کاربر سیستم است و میتواند یک فرد خارج از سیستم باشد مثل دانشجو میتواند از یک فرد داخل سیستم باشد.مثل مسئول ثبت نام برای اینکه حوزه مسئله تعیین شودو باید بازیگرها مشخص شود بازیگرها را با آدمک مشخص میکنند.

باید مشخص کردکه هر بازیگر چگونه با سیستم در ارتباط میباشدو این ارتباط را در قالب مورد استفاده مشخص میشود.در واقع مشخص میکند چه کسانی و چه استفاده هایی از سیستم میبرد و هر مورد استفاده برای هر بازیگر به صورت جداگانه ای مشخص میشود شرحی برای هر یک از این مورد های استفاده مشخص میشود و در قالب کاری که

نهایتا در این مرحله مشخص میشود و علاوه برتعیین میزان بازدهی مالی و عوامل ممکن و معیارهایی ارزیابی سیستم و تابع مورد نیاز تعیین میشود تا اینکه بتوان نهایتا برنامه ریزی پروژه را در این مرحله بدرستی انجام داده به طور خلاصه اهداف این مرحله عبارتند :

۱. تعیین دامنه مسئله
۲. تعیین مورد های استفاده
۳. ساختار کلی سیستم
۴. برنامه ریزی پروژه
۱ـ۴.زمان بندی
۲ـ۴.هزینه
۵. تعیین میزان ریسک برای انجام پروژه

در این مرحله گزارش سیستم و یا شرح سیستم جاری در قالب مدل سیستم جاری تعیین میگردد و در نهایتا برآوردهای مالی انجام میشود و همچنین یک لغت نامه تهیه میشود که اصطلاحات رایج در سیستم را تشریح میکند.

برای مثال، تعرفه ثبت نام در این لغت نامه توضیح داده میشود.

خروجیهای این مرحله عبارتند از:

۱. مستندات ویژن
۲. مدل موردهای استفاده از سیستم
۳. لغت نامه
۴. گزارش کلی
۵. گزارش میزان ریسک انجام پروژه
۶. برنامه ریزی پروژه

## BUSINESS MODELING (مدل سازی سیستم جاری)

مدل سازی سیستم کنونی و یا سیستم جاری شروع میشود و مقوله مدل سازی سیتم کاری آغاز میشود. ترتیبی که در این نرم افزار بیان شده است ترتیب صحیحی نیست ابتدا باید واحدهای سازمانی مشخص گردد و یا به عبارت دیگر چارت عملیاتی و یا چارت

سازمانی مشخص میشود. برای هر سیستم جاری همان نقش کارمندان میباشد. Business worker مشخص شود پس Business worker نقش کاربرها می باشد. اما برای تعیین آنها باید سرویسهای مورد نظر شناخته شوند.

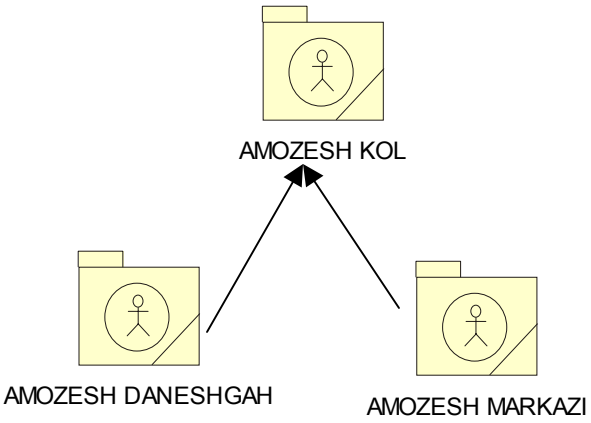
Business use case : عملیاتی میباشد که بایستی برای سرویس دهی به سایرین به انجام برسد. توجه کنید اگر فردی در داخل سیستم عملیاتی را انجام دهد حاصل آن باید مورد قرار بگیرد. مورد استفاده را اصطلاحا میگویند. مسئولیتها وابسته به موردهای استفاده مشخص میگردد. موردهای استفاده از سیستم آموزش دانشگاه برای مثال ثبت نام است که این کار توسط کارمند آموزش انجام میگیرد. در واقع مسئول ثبت نام بر روی یک موجودیت یک عملی را انجام میدهد تا اینکه آن عمل مورد استفاده قرار بگیرد.

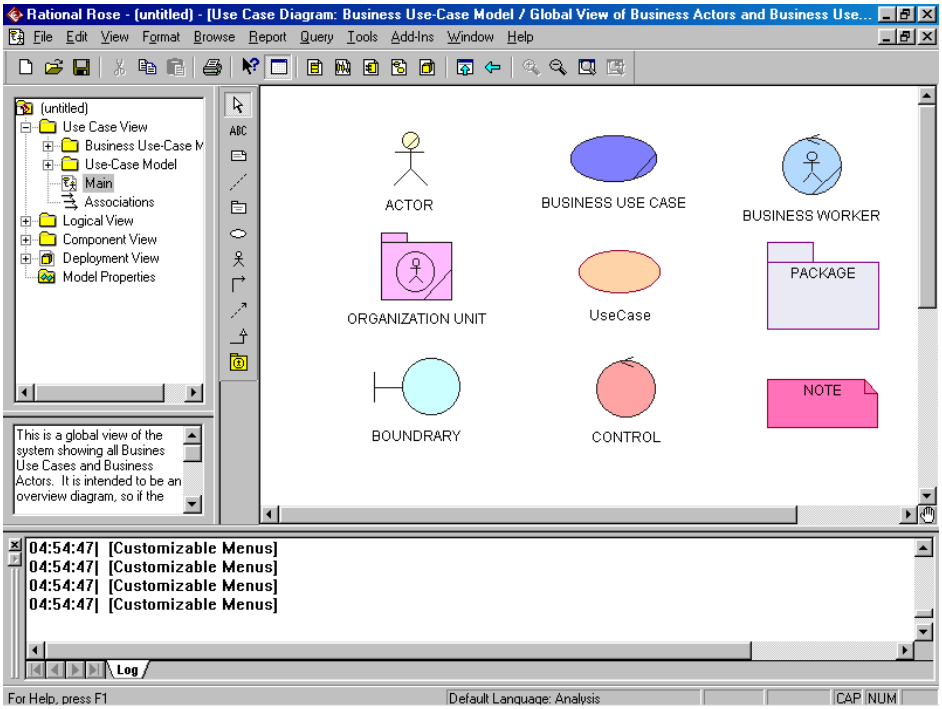
برای اینکه یک عمل ثبت نام انجام گیرد باید مراحل زیر را انجام دهیم:

۱. تعیین واحدهای عملیاتی
۲. تعیین مسئولیتها و یا نقش کاری
۳. تعیین مستندات کاری
۴. تعیین روش اعمال و یا انجام موردهای استفاده
۵. ساختن مدل ارتباطی اشیا کاری
۶. ارزیابی نتایج

## تعیین موردهای استفاده:

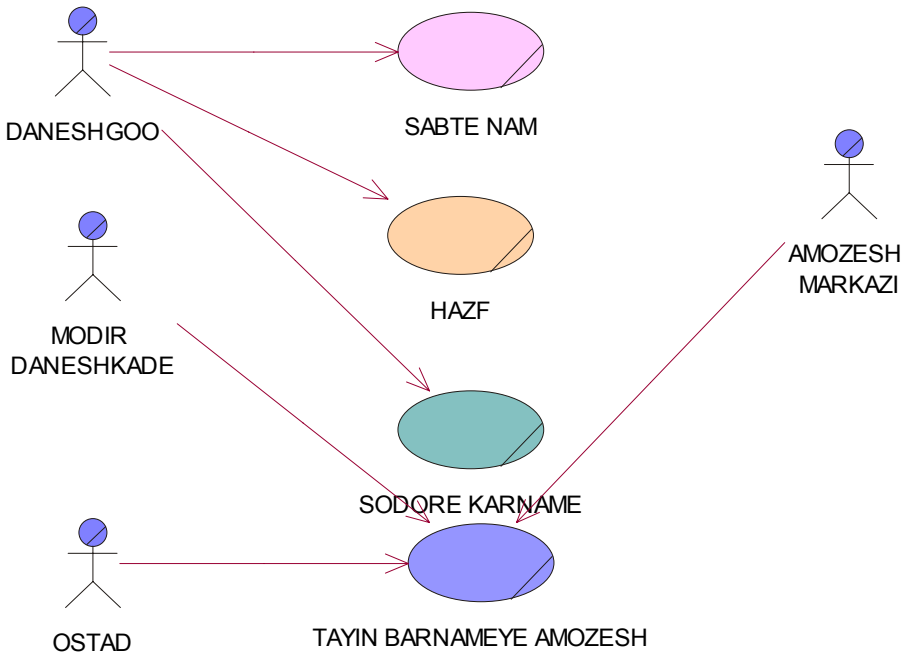
مورداستفاده شاخص سرویسی است که سیستم برای بازیگر های خود فراهم میکند. در واقع عملکرد مستندها بر اساس سرویسهایی که فراهم میکنند مشخص میشود. هر سرویس را مورد استفاده بازیگر و یا یک مورد استفاده مینامند. بنابراین پس از تعیین هر واحد عملیاتی باید مشخص نمود.و هر واحد عملیاتی چه سرویسهایی را برای خود عمل میکندو سروسی گیرنده را بازیگر و یا اکتور نیز مینامند. اگر سیستم کنونی مطرح باشد؛ مورد استفاده را مورد استفاده کاری و سرویس گیرنده را سرویس گیرنده کاری مینامند. برای نمونه اگر سیستم آموزس دانشگاه را در نظر بگیرید. در زیر سیستم آموزش، آموزش دانشکده آموزش مرکزی قرار میگیرند. بصورت زیر:



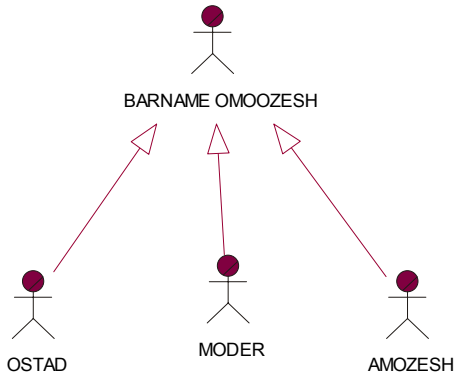




در قسمت بالا سیستم آموزش دانشکده بررسی میشود که سیستم آموزش دانشکده و سیستم آموزش مرکزی به طور کلی از سیستم آموزش استفاده میکند.

آموزش دانشکده دارای موردهای استفاده زیر است:




در این قسمت چون تعداد اکتورها زیاد میباشد و همچنین از همدیگر به ارث میبرند در نتیجه میتوانیم به این صورت رسم کنیم.




کلا مدیریت دانشگاه و استاد و بعضی از اعضا را که به همدیگر ربط دارند را در داخل دیاگرام قرار دهیم و چون دانشجو گویایی مسئله را نشان میدهد پس نمیتوان آن را در داخل دیاگرام قرار داد.

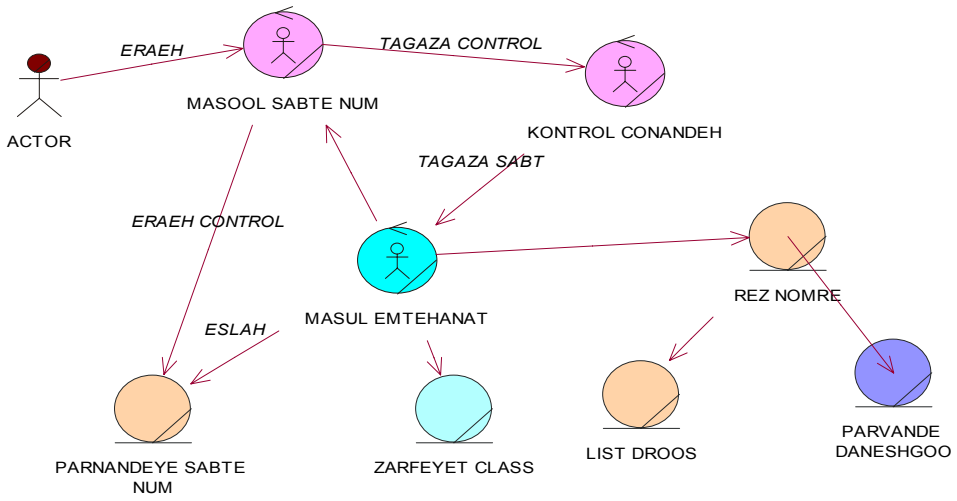
حال به طور خلاصه یک مسئله را باهم حل میکنیم. برای مثال وظیفه ثبت نام ،معمولا توسط چند کارمند آموزس و در نقشهای کنترل کننده تقاضا انجام میرسد. برای انجام هر وظیفه شرح وظیفه نیز وجود دارد که مراحل انجام کار را مشخص میکند. بنابراین برای توصیف موردهای استفاده کاری از شرح وظایف استفاده کرد. برای نمونه در مورد ثبت نام، وظیفه و یا مورد استفاده به شرح زیر میباشد.

۱. دریافت تقاضای دانشجو:مسئول ثبت نام تعرفه را از متقاضی ثبت نام دریافت میکند.
۲. مسئول ثبت نام تعرفه را جهت بررسی به مسئول پرونده های دانشجویی میدهد.
۳. مسئول پروندها وضعیت درسی دانشجو را و پیش نیازها را براساس تعرفه کنترل میکند.
۴. کنترل کننده پس از تایید و یا رد دروس و یا به عبارت دیگر اصلاح تعرفه آن را به کنترل کننده کلاسها تحویل میدهد.
۵. کنترل کننده با در نظر گرفتن ظرفیت دانشجو ثبت نام مینماید.
۶. کنترل کننده فرم تکمیل شده را تایید و به مسئول ثبت نام تحویل میدهد.
۷. مسئول ثبت نام فرم را به دانشجو تحویل میدهد.

اگر به توصیف مورد استفاده توجه کنید مشاهده میکنید که در این توصیف کارکنها همان مسئول ثبت نام و کنترل کننده کلاسها میباشد.که ممکن است همگی یک فرد باشد که با انجام عملیات برروی موجودیتها که شامل تعرفه و وضعیت درسی دانشجو و پیش نیازها و پرونده دانشجو و وضعیت کلاسها و یا به عبارت دیگر آمار ثبت نام در هر کلاس و بالاخره فرم تکمیل شده ثبت نام، تمام میشود و در واقع این همکاری خود نوع مدل سیستم جاری را نشان میدهد.

باید به تنوع همکاری بین این موجودیتها و سایر اعضا توجه کنیم. پس به طور خلاصه کار با یک چارت سازمانی کار آغاز شده و بعد چارت عملیاتی ایجاد میشود و در آخر کارها به صورت عملیات سیستم جاری ایجاد میشود و واحدهای کاری به صورت بخشهای جدا از هم عمل کرده و با یک سمبل خاصی نمایش داده میشود و برای هر واحد

کاری عملیات به صورت مورد های استفاده سیستم جاری مشخص میشود و هر یک از اینها دنباله ای از عملیات میباشد.

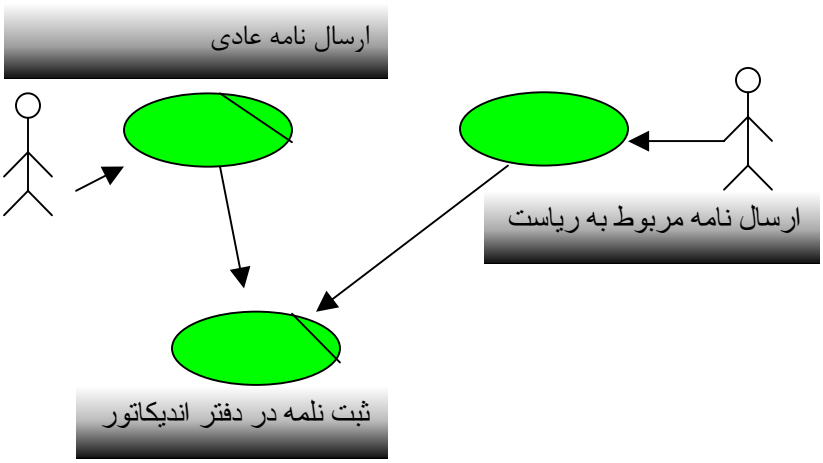



اکنون میبایست در مورد این نکته تفکر کرد که چگونه موردهای استفاده و موجودیتها در کامپیوتر پیاده سازی میشود. معمولا کارکنها تبدیل به کلاسهای کنترلی میشود و کلاسهای کنترلی با این علامت خاصی مشخص میشود.کلاسهای کنترلی وظیفه کنترل عملیات را برعهده دارد.موجودیت مبنایی برای بانکهای اطلاعاتی میباشدو موجودیتها تبدیل به کلاسهایی ازنوع موجود خواهد بود.اما در ارتباط با سیتم کامپیوتری نوع دیگری نیز از کلاسها تحت عنوان کلاسهای باندر یو یا فرمهای ورودی خروجی مطرح میشود.کلاسهای سرحدی در واقع شاخص فرمهای ورودی و خروجی رابطه ها میباشد. در واقع اینها مستقیم با محیط جانبی ارتباط دارند.کلاس فرم خود نوعی کلاس سرحدی میباشد.که ارتباط کاربر را از طریق کیبورد با برنامه مورد نظر را برقرار میکند.کلاسهای شبکه که مطرح میشود نوع دیگری از کلاسها میباشد که در این حالت این نوع کلاسها زمانی مورد استفاده قرار میگیرند که از شبکه استفاده بکنیم. به طور کلی هر چیزی که در آن اطلاعات موجود باشد و از آن استفاده کنیم کلاسهای موجودیتی گفته میشود که شامل پروندهها و یادداشتها و هر گونه اطلاعات میباشد. که در سیستم آتی مورد استفاده قرار میگیرد.کلاسهای از نوع موجودیت به صورت یک فایل میباشند.کارکنها به صورت

کلاسهای کنترلی میباشند و صرفا عملیاتی را روی موجودیتها انجام میدهند. به طور کلی در تبدیل یک سیتم جاری و یا دستی به یک سیتم کامپیوتری مورد استفادهای سیستم جاری به سیستم آتی تبدیل میشود. اکتور سیستم جاری به اکتور سیستم آتی تبدیل میشود. هر زیر سیستمی ممکن است شامل مورد استفاده باشد. پس ممکن است که این تبدیلات به صورت مستقیم انجام شود. هر بخش و یا زیر سیستم، سیستم جاری با بسته های نرم افزاری مشخص میشود و هر یک از این بسته ها به صورت یک زیر سیستم میباشد. کلاسهای سرحدی به صورت فرمهای ورودی و خروجی عمل میکنند و ممکن است یک business use case مستقیما تبدیل به یک use case شود و business entityمستقیما تبدیل به packageو یا بسته های نرم افزاری جایگزین میشوند.و هر package یک زیر سیستم کامپیوتری است و در عمل پیاده سازی package مشابه به package زبان java میباشد و به طور کلی کلاسهای سرحدی وظیفه برقراری ارتباط برنامه و یا سیستم کامپیوتری با محیط جانبی آن را بر عهده دارد و به عنوان فرم ورودی و خروجی میباشد. در این قسمت به جای type از stereotype برای تعیین نوع کلاس استفاده می شود.

Use case ها نباید در ارتباط مستقیم با یکدیگر قرار بگیرند.

ارتباط بین use case ها از طریق یک بانک اطلاعاتی صورت میگیرد که در مدل use case diagram مشخص میگردد.

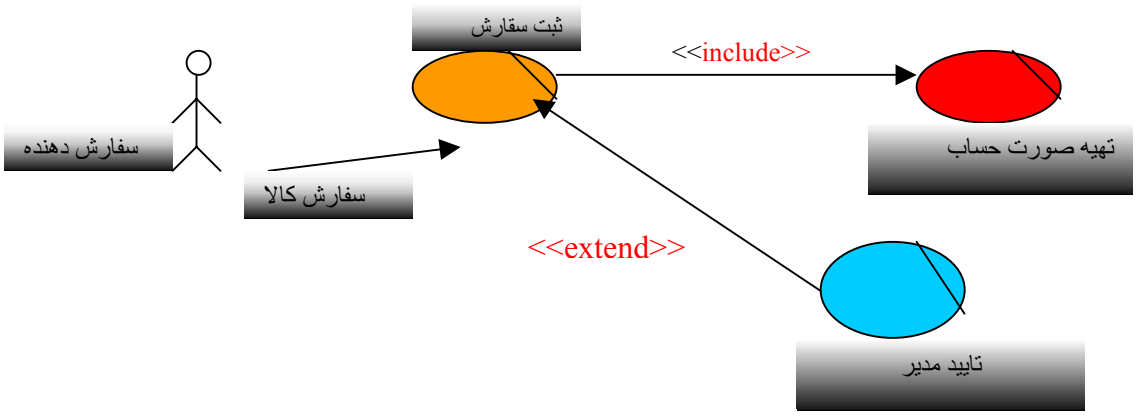


برای نمونه برای دو use case ارسال نامه عادی و ارسال نامه به ریاست عمل مشترک ثبت نامه در دفتر اندیکاتور است که به عنوان یک use case مشترک در اینجا مشخص می گردد.

اما در طرح جامع سیستمها عملیات مشترک را باید مشخص و از تکرار آنها جلوگیری کرد در این صورت عملیات مشترک را در داخل یک use case تعیین ودر رابطهinclude با use case قبلی قرار می دهیم.

علاوه بر include رابطه دیگر به نام extend وجود دارد که در واقع نشان میدهد که در حالت استثنائی عملیات یک use case چگونه توسط یک use case دیگر توسعه یا extend داده میشود .

در سیستم ثبت سفارش ، سفارشها از افرادی که پرونده دارند پذیرفته میشود در حالت استثنائی که کاربر پرونده نداشته باشد در صورتی که مدیر تائید کند سفارش مورد قبول قرار میگیرد .



هر واحد عملیاتی به عنوان actor برای واحد عملیاتی دیگر میباشد چون از یکدیگر سرویس میگیرند .

## موردهای استفاده از سیستم جاری ثبت نام :

۴. مقدمه

۳. ۱ـ۱ . هدف

۲ـ۱. حوزه عملکرد

۳ـ۱.اصطلاحها

۴ـ۱.مراجع

۵ـ۱. شرح مختصر

۱. نام مورد استفاده جاری

ثبت نام

۱ـ۱ . شرح مختصر

دانشجو تعرفه خود را ارائه و پس از بررسی ،تائیدیه ثبت نام به وی داده میشود

کارائی مورد نظر :

هدف ارائه حد اکثر دروس بدون تلاقی به دانشجو است .

گردش کار :

۱ـ۳. گردش اصلی :

| | |
|---|---|
| ۱. دانشجو تعرفه خود را جهت ثبت نام به مسئول ثبت نام ارئه میدهد . | ۲. مسئول ثبت نام تعرفه را کنترل میکند |
| | ۳. مسئول ثبت نام از مسئول پرونده ها تقاضای کنترل شرایط دانشجو را میکند |
| | ۴. مسئول پرونده ها معدل کل دانشجو را از کارنامه وی مشخص میکند |
| | ۵. دروس پیش نیاز از کاتالوگ دروس برای هر درس درون تعرفه مشخص میشود |
| | ۶. مسئول پرونده تعرفه ها راکنترل میکند |
| | ۷. مسئول پرونده ها از مسئول کنترل کننده کلاسها تقاضای تعیین کلاسها را میکند |
| | ۸. بر اساس ظرفیت کلاسها و برنامه هفتگی ،فرم ثبت نام ایجاد میشود |
| | ۹.مسئول ثبت نام فرم را به دانشجو تحویل میدهد |

۲ـ۳. موارد دیگر :

۱. در صورتیکه معدل دانشجو کمتر از ۱۴ باشد مسئول ثبت نام ۱۶ واحد به او ارائه میدهد .

۲. در صورتیکه دروس پیش نیاز را نگذرانده باشد به او درس تحویل داده نمیشود .

۳-۳. نیازهای عملیاتی

۴-۳. نقاط توسعه

## Realization :

ازطریق همکاری business worker و ثبت عملیات وبررسی business entity ها انجام میشود.

Use case realization یا به تحقق پیوستن موردهای استفاده :

برای اینکه موردهای استفاده به واقعیت بپیوندند میبایست تعدادی object با یکدیگر همکاری نمایند ودر نتیجه همکاری اینها نهایتا سرویس مورد نظر برای actor فراهم گردد.

در اینجا object ها در قالب business worker ها و business entity ها مشخص میشوند. interaction یا محاوره این object ها با یک دیگر در قالب دیاگرامهائی تحت عنوان interaction diagram میسر میگردد .

سازماندهی این object ها در قالب object model و با مدل ارتباطی کلاسها مشخص میشود .

برای تشبیه عملکرد Use case ها سناریو Use case و یا activity diagram استفاده میشود که در مورد سناریو قبلا صحبت شد .

## مدل ارتباطی کلاسها :

همانگونه که در مثال ثبت نام دانشگاه مشاهده گردید جهت انجام عمل ثبت نام کلاسها با یکدیگر در ارتباط قرار میگرفتند کلاسهائی تحت عنوان دسته بندی کلی ( sterio type )

Business worker یا کار کن business entity یاموجودیتهای کاری .

همان گونه که چارت سازمانی روابط بین افراد را نشان میدهد مدل ارتباطی بین کلاسها چگونگی رابطه بین Business worker ها و business entity هارا مشخص میکند .

اصولا مدل ارتباطی کلاسها یک نوع conceptual model یا مدل مفهومی است .

هرکلاس بعنوان یک مفهوم از ارتباط کلاسها conceptual model حاصل میشود.

Conceptual modeling یا مدل سازی مفهومی ابزاری جهت شناخت و ارائه دانش در فلسفه شناخت و هوش مصنوعی است .

در اینجا دانش ما در قالب سناریو Use case مستند گردید corept ها یا مفاهیم Business worker و business entity ها هستند .

جهت نمایش دانش مدل سازی مفهومی conceptual model استفاده میشود که بیا نگرچگونگی برقراری ارتباط بین کلاسها یا مفاهیم در جهت میل به اهداف Use case است .

## رابطه بین کلاسها :

اصولا در حالت کلی رابطه بین دو کلاس راassociation یا اجتماع بین آن دو کلاس مینامند .

رابطه بین object ها را link یا پیوند بین اشیاء مینامند .

× اما ویژگیهای اجتماع بین کلاسها چیست؟

همان گونه که در شکل ۱ـ۱ مشاهده میکنید (مثال مدل ارتباطی کلاسهای کاری برای سیستم ثبت نام ).

برای هر اجتماعی یک نام مشخص میکنند برای مثال :

مسئول ثبت نام در ارتباط کنترل با تعرفه ثبت نام قرار دارد .

برای اجتماع میتوانیم نقش مشخص کنیم مسئول ثبت نام در نقشه کنترل کننده وتعرفه ثبت نام در نقش کنترل شونده قرار دارد .

برای هر اجتماعی میتوانیم arity یا تعداد مشخص کنیم برای نمونه یک کنترل کننده ممکن است چند تعرفه ثبت نام را کنترل نماید لذا رابطه یک به چند است .

بطور خلاصه برای اجتماعها چهار جزء مشخص میشود .

- نام
- جهت

- arity

در اجتماع بسیار مهم میباشند لذا به جای اینکه نام آن را بنویسند برای آنها علامت خاصی قرار داده اند .

این دو اجتماع تحت عنوان is a kind of نوعی از و دیگری تحت عنوان is a part of یا بخشی از شناخته شده اند .

is a kind of را رابطه کل به جزء و یا رابطه وراثت نیز مینامند .

هر نوع مشترکاتی را در قالب semantics net تعریف نمیکنند super class باید بیانگر جنس super class ها باشد

رابطه دیگر رابطه کل به جزء یا is a part of یا whole -part و یا رابطه aggregation است .

برای نمونه پرونده دانشجو

ریز نمره در ارتباط به یک پرونده باشد

یک پرونده دانشجو قرار میگیرد با چند ریز نمرات

مطالب فوق را میتوانید در فصل ۳ از کتاب به طور کامل پیدا کنید .

## دیاگرامهای محاوره interaction diagram

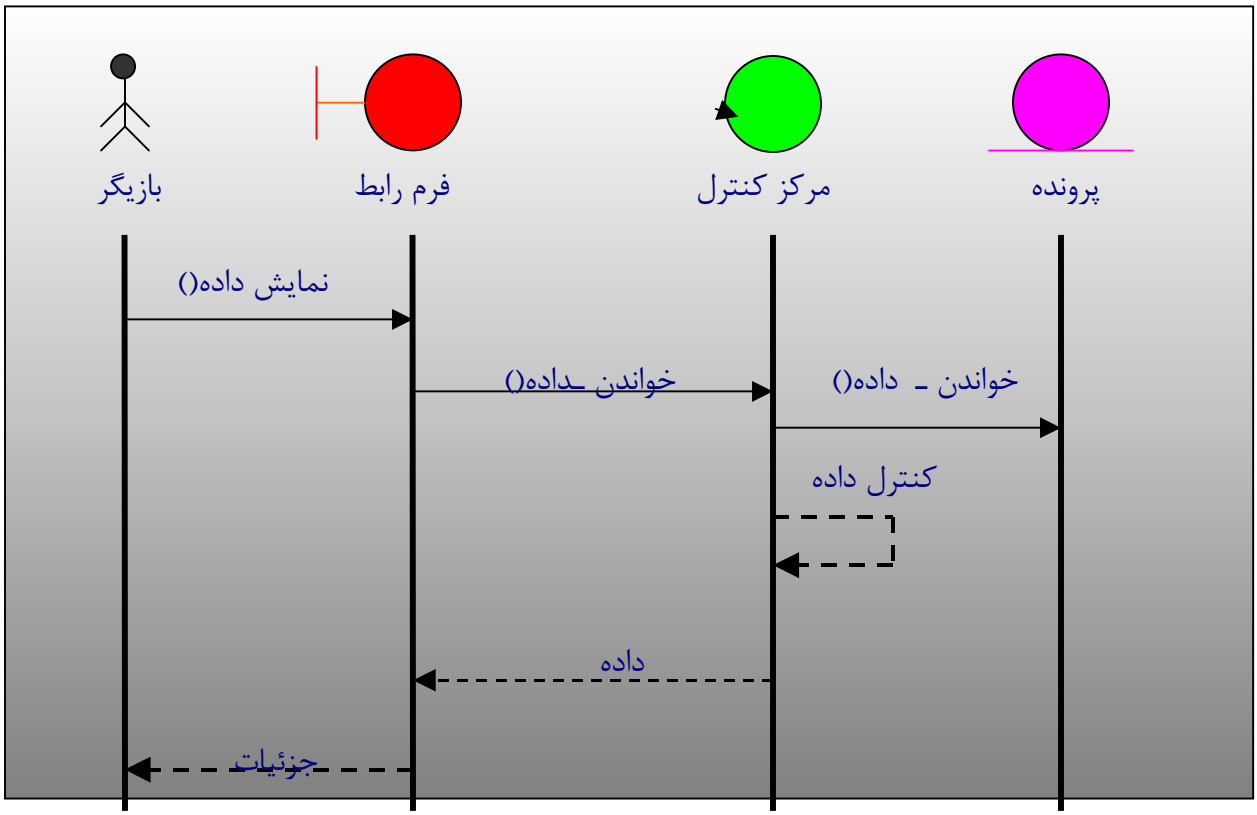
دیاگرامهای محاوره نشان میدهند که هر object کدام متد از object دیگر را فراخوانی میکند تا یک Use case به تحقق بپیوندد .

Interaction دیاگرامها شاخص دنباله ای از پیامها یا deployment یا display یا به خدمت گرفتنها و یا فراخوانی ها است که در بین object های متفاوت صورت میگیرد تا یک سرویس یا یک Use case ارائه گردد .

## دیاگرام توالی :

یک دیاگرام توالی خواندن اطلاعات از یک table و دیاگرام توالی دوم در ارتباط با نوشتن اطلاعات در داخل یک table است.

ساختار یک دیگرام توالی در فرم کلی:




شکل فوق بیانگر یک دیاگرام توالی کلی جهت نمایش جزئیات یک پرونده است .

همانگونه که در این فرم مشاهده میکنید کاربر یا actor درمقابل کامپیوتر قرار میگیرد.

فرم رابط که یک کلاس از نوع boundary class است برای وی نمایش داده میشود کاربر تقاضای نمایش داده را بصورت پیام یا فراخوانی برای این فرم ارسال میکند.

در دنیای کامپیوتر معمولا فرمهای ورودی – خروجی ابزار ارتباط با کاربر هستند .اینها همگی از کلاس فرم میباشند اما در uml در قالب boundary class مشخص میشوند .

در دنیای واقعی Business worker ها یا کار کنها عملیات را انجام میدهند اما در داخل کامپیوتر این نقشهای تصمیم گیری و کنترل عملیات بر عهده یک کلاس از نوع

کنترل میباشد لذا درخواست خواندن داده به مرکز کنترل داده میشود واین مرکز کنترل است که درخواست را برای پرونده مورد نظر ارسال میکند .

در واقع خواندن داده یک متد ارزشی مرکز کنترل است که توسط object از نوع فرم رابط فراخوانی شده است لذا در اصل ، دیگرامهای توالی دنباله ای از فراخوانیهارا مشخص میکنند

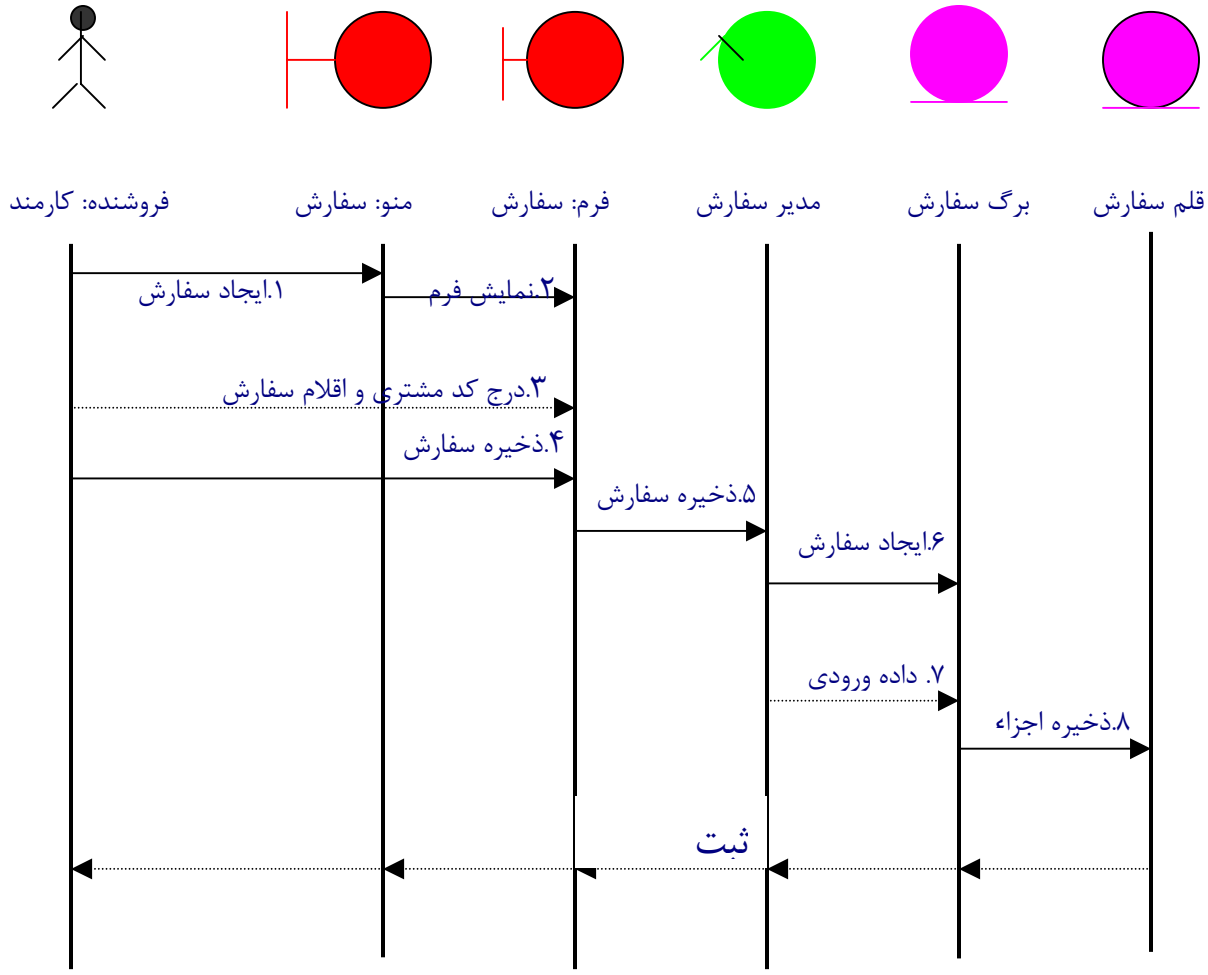
دیگرام توالی دقیقا مشخص میکند که چه object هائی در داخل سیستم کامپیوتری با یکدیگر همکاری میکنند تا یک سرویس یا مورد استفاده برای هر actor فراهم گردد .

فهمیدیم که کنترل object ها در اینجا همان Business worker ها هستند.

فرمهای مورد استفاده در اینجا boundary class ها مشخص شده اند هر گونه پرونده بایگانی یادداشت و کلا business entity در قالب business class مشخص شده است .

در واقع بوسیله یک دیاگرام توالی عملیات یک Use case بصورت دنباله ای از فرا خوانیها مشخص میشوننند.

برای نمونه در شکل زیر توجه نمایید:




همانگونه که در بالا مشاهده میکنید فرو شنده یک شی ء از نوع actor از کلاس کارمند است که در سیستم کامپیوتری خارج از سیستم قرار میگیرد در business use case که برای use case سفارش ایجاد شده به جای فروشنده، مشتری actor قرار میگیرد .

فروشنده Business worker بوده است .

معمولا actor خارج از سیستم است در سیستم دستی فروشنده در داخل سیستم می باشد اما در سیستم کامپیوتری این طور نیست و فروشنده در خارج از سیستم قرار دارد .

فروشنده از منوی خود گزینه سفارش را انتخاب میکند و با انتخاب این گزینه فرم نمایش داده میشود و پیام نمایش فرم به object از فرم سفارش ارسال میشود و بلافاصله فرم کد مشتری و اقلام سفارش در داخل فرم درج میگردد وبا دادن پیام ذخیره عملیات آغاز میگردد .

معمولا دیاگرام توالی در دو مرحله کشیده میشود:

۱- در یک مرحله دیاگرام توالی برای تطابق با سناریو با جملات فارسی به عنوان پیامهای رد و بدل شونده ایجاد میشود .

۲- در مرحله دوم دیاگرام توالی به صورت دنباله ای از فرا خوانیها ایجاد میشود.

تا به حال چه خوانده ایم و در کجا هستیم:

۱- تعیین چارت عملیاتی و چارت سازمانی

۲- تعیین هدف – شرح مساله

۳- تعیین نیازهای عملیاتی و کیفی برای هر واحد عملیاتی

۴- تعیین business actor ها یعنی کسانی که قرار است در خارج از سیستم از آن استفاده نمایند .

توجه: هر زیر سیستم یا واحد عملیاتی به عنوان یک actor برای هرواحد عملیاتی دیگر در نظر گرفته میشود.

۵- برای هر business actor، use case ها مشخص میگردد برای هر business use case یک سناریو در داخل فایل word نوشته میشود .

۶- use case هاکه متعلق به یک واحد عملیاتی هستند همگی در زیر package مربوط به آن واحدعملیاتی قرار میگیرند.

۷- در logical view تحت business object model واحدهای عملیاتی یا business unites مشخص میگردد

- در اینجا در واقع چارت عملیاتی که در قالب درختی است به صورت واحدهای عملیاتی تو در تو مشخص میگردد.

برای هر واحد عملیاتی بر اساس وظیفه آن واحد و use case های مربوطه همانطور که در کلاس تدریس شد business object model میکشید .

به این ترتیب مرحله شناخت اولیه تمام میشود مرحله تشریح با آنالیز شناخت آغاز میگردد توجه کنید که با شناخت سیستم کاری میتوانید به عقب برگشت کنید و نیازهای عملیاتی و کیفی تکمیل کنید .

اکنون مطالب گردآوری شده را مورد تحلیل قرار میدهیم تا بتوانیم بر اساس تحلیل خود برای سیستم آتی موردهای استفاده را مشخص کنیم وبرای هر مورد استفاده بر اساس دیاگرام توالی عملیات را مشخص کنیم .

تحت آنالیزیز مدل باید تعیین کنیم که هر use case سیستم کامپیوتری کدام use case از سیستم کاری را به واقعیت نزدیک میکند و یا محقق مینماید ویا realize مینماید .

Parsa /umldocs/tutorials / rose totorial / modifer /completed

هر فردی که جلوی کامپیوتر بنشیند یک use case یا سرویس از آن می خواهد .

در هنگام آنالیز در داخل انالیزیز مدل میبایست بخشی را به use case realization تخصیص دهیم به این صورت که مشخص کردیم هر use case ، business model توسط چه use case کامپیوتری realize وبه واقعیت پیوند داده شده است.

به طور مثال در برنامه rose totorial که در cd است و دایرکتوری آن را در بالا مشخص کرده ایم تحت آنالیزیز مدل برای سیستم pointof syc (pos) مشخص شده که use case های دستی با کدام use case های کامپیوتری realize میگردند.

use case هائی که realize میگردند در واقع use case هائی هستند که با خط پر مشخص شده وuse case های realize با خط چین مشخص شده اند .

باید توجه کنید که تحت logical view سه package آنالیزیز مدل و business model و design model قرار میگیرد .

تحت business object model یک business unit

۳- بر هرunit business object model، مشخص میشود .

در واقع business unit یک واحد عملیاتی است که در دیاگرام عملیاتی مشخص گردیده است.

business object model شاخص عملکرد و یا ساختار عملیاتی آن واحد است .

rational unified process/view/browser/use case view /business use case

از use case باید ببریم در داخل logical view

مدل ارتباطی کلاسها باید در logical view کشیده شود

به ازاء هر table لازم نیست کلاس اضافه کنید

دیاگرام توالی میکشیم use case view

۱- مدل سازی سیستم جاری

۲- براساس سیستم جاری مستندی تعین میکنیم به نام vision

درvision نیازها قرار میگیرد و بر اساس نیازها قابلیت سیستم جدید تعین میشود

Require analysis می کنیم.

Use caseیک functionمی باشد نه یک کلاس

Use case های سیستم آتی را می شنا سیم که مر حله شنا خت می باشد.

Use case viewنشان می دهد در سیستم آتی چه عملیا تی انجام می دهیم.

در فاز طراحی یک قسمت usecase realizationداریم.

کلیه نیازها و پاسخ به نیازها یعنی قابلیتهای سیستم در مستنداتی به نام vision یا چشم انداز پروژه می بایست گنجانده شود در چشم انداز شرح مسئله گنجانده میشود منظور از business opportunity در این مستند در واقع امکانات جدید کاری است که سیستم آتی برای مشترک ایجاذ خواهد کرد .

به طور مثال میزان فروش دو برابر خواهد شد فروش را میتوانیم با وجود سیستم کامپیوتری در سراسر کشور امکان پذیر کنیم.

و یا اینکه در مورد سیستم آموزشی میتوانیم برای مثال بگوئیم با وجود یک سیستم کامپیوتری امکان ایجادکلاسهای مولتی مدیا و تست دانشجویان به طور ا توماتیک وجود خواهد داشت .

در این مستند کاربرها در قالب stakeholder و user مشخص میشوند stakeholder در واقع فردی است که برای سازمان مورد نظر کار میکند و user فردی است که از سیستم کامپیوتری استفاده خواهد کرد .

در قسمت product overview در این مستند چشم اندازی از محصول نرم افزاری ارائه میشود هدف از مستند چشم انداز همین میباشد که تعین کنیم سیستم آتی چه امکاناتی برای کار بر خواهد داشت .

در اینجا what مطرح است how مطرح نیست .

یعنی در چشم انداز مشخص میکنیم که سیستم چه انجام خواهد داد . چگونگی مسئله طراحی است .

در بخش product perspective علاوه بر این خلاصه ای از قابلیتهای اصلی summery of culpabilities مسئله هزینها و هر یک از قابلیتها به طور مجزا و هزینه ارائه خواهد شد .

باید توجه کنید که قابلیتها در واقع پاسخی به نیازها است . نیازها بعد از قابلیتها مشخص میشود شاید بهتر بود قبل از انها مشخص میشد .

در اینجا ارجاع به راهنمای user و راهنمای نصب که در انتهای پروژه به کاربر تحویل میشود ، تهیه میگردد بنابر این میتوانید مستندات vision خود را اکنون با در دست داشتن لیست نیازها تکمیل نمایید .

نهایتا بعد از تعیین نیازهاو قابلیتها use case view با ایجاد use case model کامل می شود در واقع use case model ورودی مرحله تحلیل و طراحیست.

use case model نشان میدهد که سیستم آتی چه function ها و عملیاتی باید انجام دهد وبر مبنای use case کار تحلیل وطراحی اغاز میشود حال این use case های آتی در مرحله میبایست realize شود یا به اهمیت به پیوندد . ممکن است یک use case در مدل آنالیز با سه یا چهار use case جایگزین شود.

در واقع بعد از کشیدن use case view وارد logical view شده و مستندات را در این قسمت وارد میکنیم .

در مرحله طراحی باید مشخص شود چه use case هائی میبایست مورد پیاده سازی قرار بگیرد برای این منظور برای هر use case با استفاده sequence diagram و مدل ارتباطی کلاسها مشخص شود که آن use case چگونه پیاده سازی خواهد شد .

البته اگر بخواهیم اینگونه پیش بریم کارها بسیار طولانی خواهد شد .

مشکل ما این است که اغلب use case model را با مرحله طراحی یکی میکنند منطق به ما میگوید ابتدا مشخص کنید چه قابلیتهائی باید وجود داشته باشد این قابلیتها نهایتا باید در قالبuse case mew مشخص میشود و سپس باید چگونگی مشخص شود.

چگونگی در قالب تحلیل و طراحی سیستم تعیین میگردد.

Parsa / undocs / tutorials / rose totorial / modifer / completed

برای هر use case نیاز نیست دیاگرام توالی بکشیم.

## مستندات VISION

1.Bussines Modeling

۲.تعیین لیست نیازمندیها بر اساس سیستم جاری

۳.use case های آتی

۴.بر مبنای این دو کاتالوگ نیازمندیها کامل شد

۵.use case view (چه استفاده هایی از سیستم کامپیوتری)

به این ترتیب مرحلهٔ طراحی آغاز میشود.

اولین مرحله در این مرحله تجزیه و تحلیل است.

pipeline, Package ها(sub system ها) به طریقی به هم متصلند.

امکانی ایجاد کرده است که از یک قسمت به جای اینکه ورودی را از keyboard بگیرد، از خروجی قسمت دیگر میگیرد.

معماری سه لایه : Application-user interface-بانک اطلاعاتی

هر لایهٔ زیرین مستقل از لایهٔ بالایی ساخته می شود.

Pattern ها مطرح میشوند. الگوی Abserver : مثلا مدیر اگر می خواهد اطلاعات را ببیند، در قالب چارت باشه، سپس آن را از لایهٔ پایینی جدا کن.

Get() ای دارد که اطلاعات را میگیرد.

تهیهٔ اطلاعات ⟵ مربوط به Application

. next، First

First اطلاعات دانشجوی اول و next بعدی.

این تولید اطلاعات در Interface صورت میگیرد.

Application باید این اطلاعات را از بانکهای اطلاعاتی مختلف بدست آورد که این کار را Application انجام می دهد.

در طراحی سه لایه ، سیستمها به سه Package تقسیم میشوند، که عبارتند از:

۱-بانک اطلاعاتی

۲-لایهٔ Application ، منطقی یک لایهٔ کاری که کارها را انجام میدهد. برای هر کاری دو تابع first وnext ، first اولین اطلاعات را می گیرد و next اطلاعات بعدی را می گیرد.

در توابع لایهٔ presentation (Interface ) فراخوانی می شوند.

۳-user interface

مزیت: هرجا بخواهیم، میتوانیم بدون اینکه برنامه را تغییر دهیم، تنها در لایهٔ user interface تغییراتی را اعمال کنیم.

پس در طراحی ۳ لایه سیستم به ۳ قسمت یا ۳ package تقسیم میشود.

## مدل ارتباطي كلاسها

مدل ارتباطی کلاسها را قبلا در قالب مدل ارتباطی اشیاء مشاهده نمودید.سؤال اینجاست:

۱ـمدل ارتباطی کلاسها چیست؟

۲ـچرا مدل ارتباطی کلاسها مطرح است؟

۳ـ چگونه مدل ارتباطی کلاسها را ایجاد میکنند؟

مدل ارتباطی کلاسها در واقع ساختار برنامهٔ شئ گرا را مشخص میکند. نشان میدهد که به جای افراد چه کسانی یا چه کلاسهایی در سیستم وجود دارند درست مثل پستهای سازمانی .برای هر پست سازمانی شرح وظایف تعیین میشود، برای کلاسها نیز شرح وظایف مشخص میشود. شرح وظایف را در قالب متدهای کلاس پیاده سازی میکنند. کلاسی که فقط متدهای آن تعیین شده باشد بدون پیاده سازی به نام Abstract class شناسایی میشود.

در دنیای واقعی چارت سازمانی و در دنیای ش‌ء گرا مدل ارتباطی کلاسها مطرح است. اما سؤال اینجاست مدل ارتباطی کلاسها یا به عبارت دیگر ارتباط کلاسها چگونه به وجود میآید و ارتباط کلاسها چیست؟

سؤال اول اینکه کلاسها چگونه مشخص میشوند؟

شما یک use case دارید، use case سرویسی است که به کاربر

قرار است ارائه شود این سرویس لازمهٔ ارائه اش تعدادی کلاس میباشد.

Business worker ها تبدیل به کلاسهای کنترلی میشوند (البته در صورت نیاز.)

Business entity ها به کلاسهای entity تبدیل میشوند.

کلا میخواهیم از شرح سناریو اسامی را استخراج کنیم معمولا اسامی همان اسامی کلاسها میشوند در واقع این کلاسها هستند که با همدیگر همکاری میکنند تا use case چه جاری، چه آتی وظیفه اش را خیلی خوب انجام دهد.

برای نمونه به صفحه ۵۸ از بخش دوم کتاب مراجعه شود.

مدل شرح سناریو برای سیستم فروشگاه مشخص شده و از داخل سناریو، اسامی کلاسها مشخص شده، برای نمونه در شرح سناریو کلمهٔ کالا استفاده شده ، پس کالا به عنوان یک کلاس در نظر گرفته شده است.

| | |
|---|---|
| اکتور | سیستم |
| اجناس مورد خرید | تعیین قیمت کالا واضافه کردن |
| صندوق..... | اطلاعات اشیاء برای اجرای تراکنش فروش |
| فروشگاه..... | |

کلاسهارا در logical view مشخص می کنیم.

برای هر کلاس مسئولیتها را مشخص می نمائیم.

پس از تعیین کلاسها می بایست برای هر کلاس مسئولیتها را مشخص کرد و تعیین نمود که آن کلاس با چه کلاسهای دیگر همکاری میکند تا مسؤلیتهای خود را به انجام برساند

در دنیای واقعی یک use case وجود دارد یعنی یک سرویسی باید ارائه شود. برای ارائه سرویس لازم است تا تعدادی افراد مشخص شوند. این افراد کلاسهای ما را نشان میدهند، هر فردی شرح وظیفه ای دارد،اینها وظایف کلاسهای ما میشوند که بعدا تبدیل به متدهای کلاس میشوند.

هر فردی با سایرین همکاری میکند تا سرویس کلی ارائه شود اینها همکارهای کلاس هستند.برای مثال در زیر در قالب فرمهایی به نام CRC یا به عبارت دیگر class responsibility collaboration کارت به صورت زیر مسئولیتهای کلاس وهمکاری های وی مشخص میشود.

در ادامه ۳ عدد CRC کارت برای سیستم کتابخانه مشخص شده است.

*عضو کتابخانه*(یک کلاس است)

| مسئولیت | همکارها |
| --- | --- |
| نکهداری اطلاعات نسخه ها | |
| در امانت عضو دریافت تقاضای | نسخه |
| برگشت یا امانت نسخه ها | |

*کتاب*

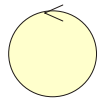
| مسئولیت | همکارها |
| --- | --- |
| نگهداری اطلاعات یک کتاب | نسخه |
| آگاهی از وجود نسخه قابل امانت | |

نسخه

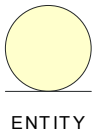
| مسئولیت | همکارها |
| --- | --- |
| نگهداری اطلاعات به یک نسخه از کتاب | کتاب |
| اطلاع امانت یا برگشت نسخه به کتاب | |

مثلا اگر یک کلاس لیست انتظار داشته باشیم و یک کلاس نوبت،
که کلاس لیست انتظار کلاس نوبت را CALL میکند یعنی Find.
متد کلاسی در داخل کلاس دیگری call شود، آنگاه آن کلاس، کلاس همکار است.

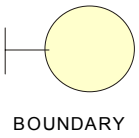
هنگام بدست آوردن مدل ارتباطی کلاسها سیستم را سازماندهی میکنیم.یعنی چه کسانی چه وظایفی دارند؟چه ارتباطی با هم دارند؟
یک مسأله پیدا کردن کلاسها بود،که در بالا نشان دادیم چگونه در قالب کارتهای CRC بر حسب وظایف کلاسها مشخص میشوند.
توجه:
میبایست حتما برای هر کلاس وظایف کلاس مشخص گردند.
کلاسها در حالت کلی به سه دسته تقسیم میشوند که عبارتند از:
کلاس کنترلی




کلاس از نوع موجودیت Entity



کلاس از نوع سرحدی یا Boundary




هر کلاس یک نوع دارد که به آن stereotype کلاس میگویند.

برای use case ها CRC cards در میآوریم.

Use case : وظیفه ای که باید انجام شود.

کلاسها کسانی که باید با هم همکاری کنند تا این وظیفه به نتیجه برسد collaboration .

خلاصه:

چگونگی تعیین ارتباط بین کلاسها مسئله دوم پس از تعیین کلاسها است، کلاسها با یکدیگر همکاری میکنند، همکاری بین کلاسهاست که موجب میشود وظایف به انجام رسند، وظایف در قالب use case ها مشخص شده اند.

کلاسها را میبایست در قالب logical view وتحت design model مشخص نمایید.

برای هر package به صورت جداگانه کلاسها را مشخص میکنند.

اگر نیاز باشد، به ازاء هر use case یکpackage تعیین می کنید و برای آن package کلاسها را مشخص می کنید در واقع این کلاسها هستند که به کمک همدیگر use case را به واقعیت می رسانند.

مدل ارتباطی کلاسها را علاوه بر کارتهای CRC می توانید مستقیما از sequence دیاگرام استخراج کنید.اگر در داخل کارتهای CRC کلاسA همکاری به نام کلاس B دارد یک خط ارتباط دهنده بین کلاسA و کلاسB می بایست در مدل ارتباطی کلاسها تحت logical view ترسیم شود.

به همین ترتیب از روی دیاگرام توالی اگر یک object از روی کلاس A ، متدی از یک object از نوع کلاس B را فراخوانی می کند، بین این دو کلاس یک خط واصل ارتباطی ترسیم می شود.

بنابراین برای هر use case بر اساس sequence دیاگرام می توان مدل ارتباطی کلاسها را درآورد.

برای نمونه به دیاگرام توالی صفحهٔ ۳۸۴ کتاب ارجاع کنید.

در این دیاگرام توالی برای مثال کلاس Torderform در ارتباط با کلاس Tcustomer از طریق متد show از کلاس Torderform فراخوانی می شود و بدین وسیله مشاهده میکنیم که دو کلاس در یک ارتباط قرار دارند ارتباط بین دو کلاس find می باشد، ارتباط بین دو کلاس را association یا اجتماع کلاسها نیز مینامند.

اما برای هر association یا اجتماع بین دو کلاسسسس چه ویژگی هایی در نظر میگیرند:

۱ـ نام اجتماع برای مثال در مورد فوق کلاس Torderform در اجتماع یافتن با کلاس Tcustomer قرار داشت.

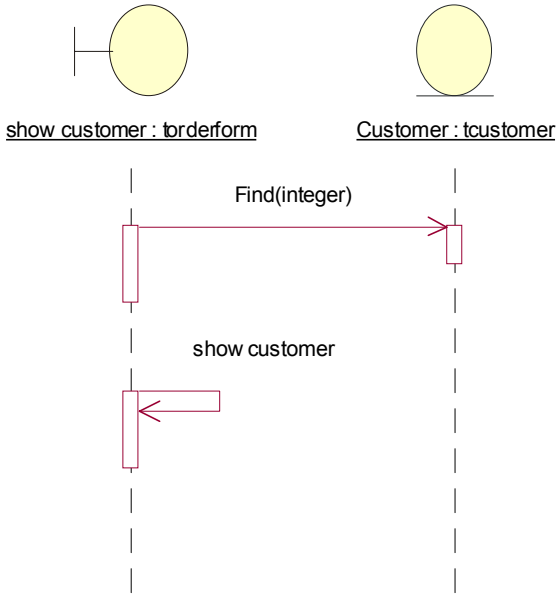
۲ـ جهت اجتماع

در مثال فوق برای نمونه جهت از سمت Torderform به سمت Tcustomer بود.

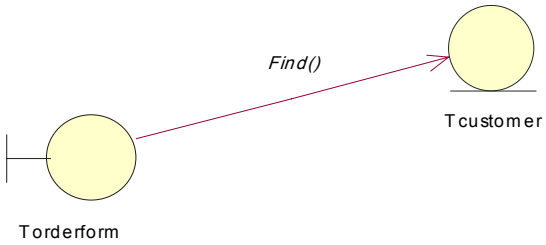
۳- arity یا چند به چند بودن اجتماع

قبل از بیان مطلب فوق به نکتهٔ زیر توجه کنید ، اگر کلاسA با کلاسB در ارتباط قرار دارد ببینیم در هنگام تولید کد چه اتفاقی می افتد.

با توجه به دیاگرام توالی :




مدل ارتباطی کلاسها حاصل از sequence فوق به صورت زیر است:




می خواهیم از طریق کلاس Torderform متد find از کلاس Tcustomer را فراخوانی کنیم. کلاس Torderform دارای Attribiute زیر است:

No
Data
Supplier
Total

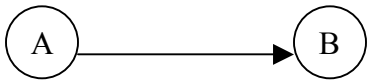
برای اینکه بتوانیم متدfind را فراخوانی کنیم نیاز به یک object از نوع Tcustomer داریم لذا در هنگام تولید کد رشنال رز بطور اتوماتیک فیلد زیر را به Attribute های کلاس Torderform اضافه میکند.

Dim the Customer As T customer

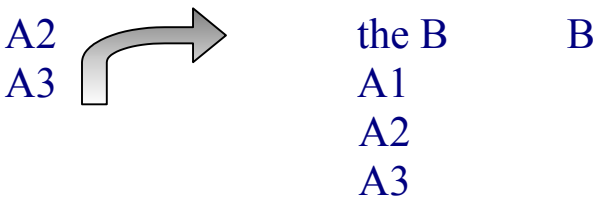
بدین ترتیب هنگامی که میخواهیم داخل متد show() از کلاسTorderform متد Find را فراخوانی کنیم دستور العمل زیر را قرار می دهیم. The Customer.Find(cno)

cno:customer number

بطور خلاصه اگر داشته باشیم این کلاس A در ارتباط با B است،




وفیلدهای A به ترتیب A3,A2,A1 باشند درهنگام تولید کد مولد کد فیلد the B از جنس B را به فیلدهای کلاس A اضافه می کند،به این ترتیب فیلدهای کلاس A به صورت زیر خواهند شد. the B As B




در مثال فوق ارتباط بین کلاس A وB یک ارتباط یک به یک است. ارتباط میتواند یک به چند باشد.

در هر اجتماع هر کلاس دارای یک نقش می تواند باشد. برای مثال

یافتن ویژگی مشتری Rot A و Rot B می باشد

Torderform نمایش دهنده : Rol A

Tcustomer نمایش یابنده : Rol B

می توان نقش کلاس Torderform را در ارتباط فوق نمایش دهنده ونقش کلاس Tcustomer را یابنده نامید.

رل A و B را مشخص میکنیم:

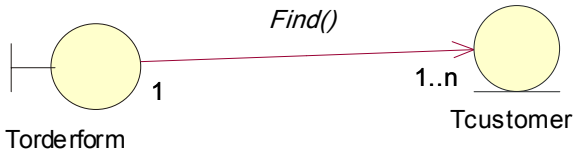
Tcustomer: یابنده جزئیات

Torderform: نمایش دهنده جزئیات

یا برای نمونه: می توان نقش کلاس Torder را در ارتباط فوق،نمایش دهنده ونقش کلاس Tcustomer یابنده نامید.

این نقشها را به خاطر گویایی مشخص می کنند.

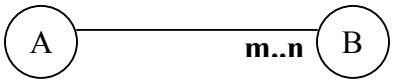
می توانیم پس ازکلیک کردن برروی خط واصل بین دو کلاس Multiplicity یا درجه ارتباط را برای هر کلاس مشخص کنیم،برای نمونه در شکل زیر، یک شئ از جنس Torderform در ارتباط با یک تا چند شئ (1..n) شئ از جنس Tcustomer است.




بدین ترتیب در هنگام تولید کد مشاهده میکنیم که در داخل کلاس Torderform فیلد زیر اضافه شده است:

The Customer : array [1..n] of Tcustomer

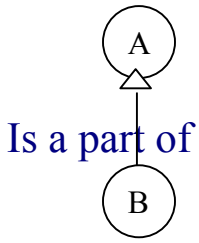
اگر داشته باشیم:

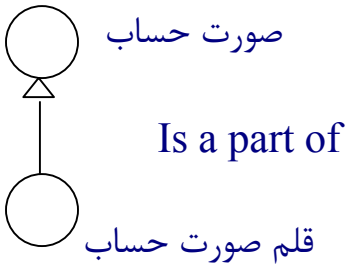



در هنگام تولید کد چنا نچه At های A,A1,A2,A3 باشد، بعد از تولید کد، فیلدها به صورت زیر خواهند شد:

| قبل از تولید کد | بعد از تولید کد |
|---|---|
| A1:T1 | A1:T1 |
| A2:T2 | A2: T2 |
| A3:T3 | A3:T3 |
| | the B : array [m..n] of B |

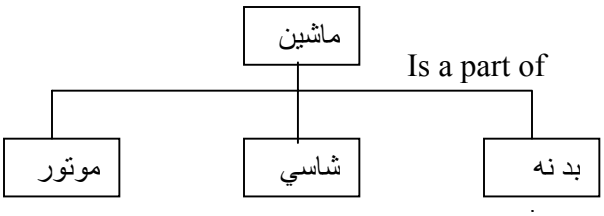
برای دو اجتماع خاص علائم خاصی در نظر گرفته اند.



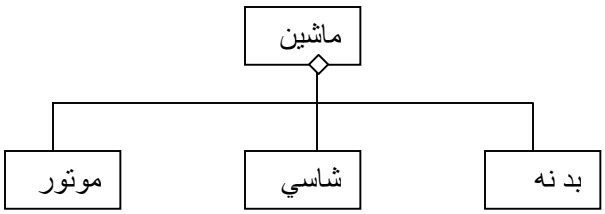




اجتماع فوق را تجمع یا aggregation مینامند.

برای مثال یک ماشین یک شئ مجتمع است.مجتمع است از : شاسی ، موتور، بدنه




رابطهٔ فوق را رابطه تجمع و ماشین را مجتمع می نامند. این رابطه را با یک لوزی به صورت زیر میتوان مشخص کرد:




اصولا دو نوع رابطه تجمع وجود دارد یکی رابطهٔ ظرف و مظروف و دومی رابطهٔ Part Assembly .

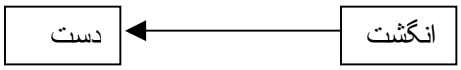
کلا رابطه تجمع را رابطهٔ کل به جزء یا All Parts نیز می نامند.

برای نمونه به مثالهای زیر توجه کنید:

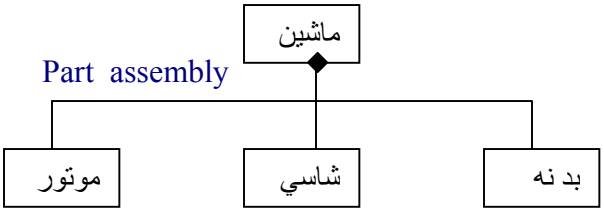
ارتباط بین دو object = link

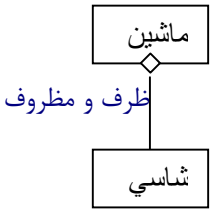
ارتباط بین دو کلاس = Association

## اجتماع:

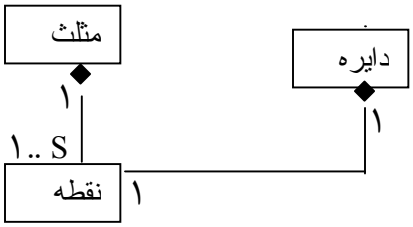



در رابطهٔ Part Assembly ، شئ کل حتما از شئ جزء تشکیل می گردد و بدون شئ جزء نا مفهوم است. برای مثال رابطه انگشت و دست را در نظر بگیرید:

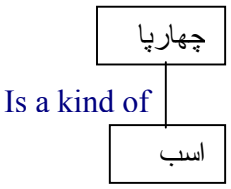


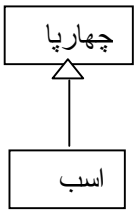

Graphical object یک super class است. و point آن را extend می کند.



نوع دیگر از روابط شناخته شده ، روابط is a یا رابطهٔ is a kind of می باشد.




به رابطهٔ is a kind of (نوعی است از) رابطهٔ وراثت هم گفته میشود و آن را به صورت زیر نیز نمایش می دهند.




اصولا رابطهٔ بین super class و class یا رابطهٔ بین کلاس ما فوق و زیر کلاس آن (sub class ) رابطهٔ is a kind of است. کلیهٔ ویژگیهای super class در داخل sub class هاست.

کلا دو رابطه کل به جزء و کلی به جزئی وجود دارد. رابطه کلی به جزئی بیانگر ارتباط بین super class و sub class ، و رابطه کل به جزء رابطه بین شئ مجتمع و اجزاء تشکیل دهنده و یا درونی آن است.

اصولا شرط شناخت و توصیف اشیاء را کلیات خمس می نامند.

| انسان | حیوان | ناطق | است. |
|---|---|---|---|
| کلیت نوع | کلیت جنس | کلیت فصل | |

| حمار | حیوان | ساهل | است. |
|---|---|---|---|
| ↓ | ↓ | ↓ | |
| Class | super class | ویژگیهای خاص class | |

انسان، چهارپایان ، پرندگان ، همگی از جنس حیوان هستند.حیوان در همه این موجودات وجود دارد و از آنها جدا نیست.

باید توجه کنید که کلاس ما فوق باید بیانگر جنس یا ذات زیر کلاسها باشد و هر مشترکاتی را نمی توان به عنوان supper class در نظر گرفت.

پس از اینکه مدل ارتباطی کلاسها را مشخص کردیم، آنگاه می بایست برای اینکه مدل ارتباطی کلاسها گویاتر شود، مشترکات بین کلاسها یا جنس کلاسها را در قالب supper class مشخص کنیم.

## Data model تولید مدل بانک اطلاعاتی

مدل بانک اطلاعاتی را می توان از Entity class ها استخراج نمود. Entity class ها کلاسهایی بودند که داده های آن (ویژگیهای) آنها باید در داخل یک فایل ذخیره شود. مدل بانک اطلاعاتی میتواند Relational یا object oriented و یا هر مدلی باشد. و ربطی به روش تجزیه تحلیل ندارند.

Entity class ها بیانگر موجودیتها یا Entity ها و عملیاتی که بر روی Entity ها انجام میگیرد.

اما Entity چیست؟

یک Entity یا موجودیت به آن چیزی اتلاق می شود که داده ای را در خود نگهداری میکند. به عبارت دیگر یک Entity بیانگر ساختار یک Table است. باید دارای یک هویت یگانه باشد. تعدادی صفت یا ویژگی دارد، صفتها یا ویژگیها در قالب فیلدهای Table و هویت یگانه در قالب کلید دسترسی یگانه ( unique Access key)

اصولا برای یک Table کلید دسترسی یا unique است به عبارت دیگر primary می باشد.

هر Entityدارای یک primary است و یک secondary key (کلید ثانویه) دارد، و همچنین دارای Foreign key است.

Foreign key : کلید اصلی برای پرونده دیگر و به عنوان کلید خارجی برای ما است.

Secondary key یا کلید ثانویه unique نیست.درست مثل نام دانشجو که کلید ثانویه برای وی می باشد. شماره دانشجو کلید اصلی است و کلید خارجی برای دانشجو، کد دانشگاه میباشد. در واقع کلید خارجی، کلید اصلی برای فایل دیگر یا Entity دیگری

است که خارج از این Entity قرار دارد. برای نمونه اگر فیلد F5 برای Entity ،A یک foreign key باشد در ارتباط با A هیچگونه index یا فایلی برای آن ایجاد نمیشود. فیلد F5 کلید اصلی برای دیگری مثل B می باشد.

کلید foreign key ارتباط بین موجودیتها برقرار میشود.

بطور خلاصه هر مدیریت دارای مشخصات زیر است:

۱.تعدادی صفت یا ویژگی دارد.

۲.یک هویت یگانه می باشد.

۳. دارای تکرار میباشد و یکی نیست.

برای مثال دانشجو یک مدیریت است. صفات آن شامل، شماره دانشجویی ، نام، نام خانوادگی، کد دانشگاه و آدرس است.

دانشجو دارای تکرار است.

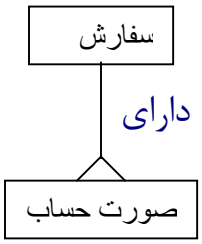
هویت یگانه : کد دانشجویی

Primary key (کلید اصلی) : شماره دانشجو

کلید ثانویه : نام دانشجو

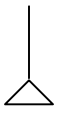
کلید خارجی: کد دانشکده

اما رابطه بین موجودیتها را چگونه بدست می آورند؟



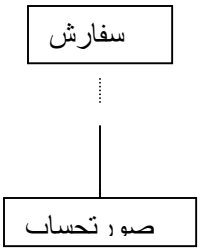

هر سفارش باید دارای یک صورتحساب باشد

هر صورتحساب باید برای یک سفارش باشد.

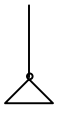


هر سفارش باید برای یک یا چند صورتحساب باشد.

برای پیاده سازی ارتباط همانگونه که قبلا گفته شد، از Foreign key استفاده می شود. برای این منظور کد سفارش در داخل صورتحساب به عنوان یک فیلد باید درج شود تا مشخص شود هر صورتحساب برای کدام سفارش است.




هر برگ سفارش ممکن است دارای صفر یا بیشتر صورتحساب باشد.



سفارش: کد سفارش+کد سفارش دهنده + نام سفارش دهنده + تلفن + تاریخ +{

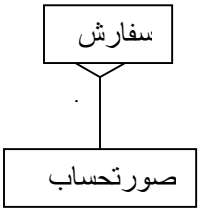
صورتحساب: کد صورتحساب + تاریخ + کد مأمور خرید + نام مأمور خرید + کد سفارش دهنده + نام سفارش دهنده + {کدکالا + نام کالا + واحدشمارش + قیمت کل + توضیحات}+ مبلغ کل صورتحساب}

رابطه یک به چند از طریق کلید خارجی کد سفارش ایجاد میگردد.

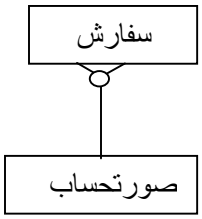
کد صورتحساب برای برگ صورتحساب، کلید اصلی(primary ) است.

یک صورتحساب برای چند سفارش : چند لاستیک و ...

هر صورتحساب باید برای یک یا چند برگ سفارش باشد.

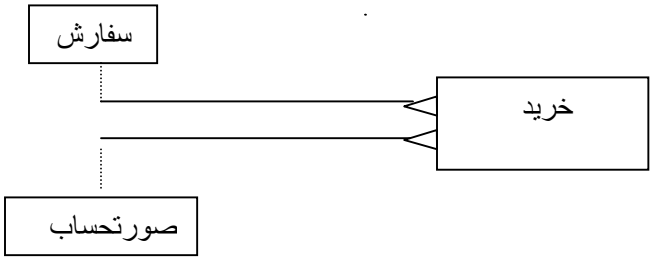



صورتحسابی است که دارای برگ سفارش نیست.




رابطه یک به چند بین سفارش و صورتحساب را از طریق درج کد سفارش در فایل حال رابطه یک به چند صورتحساب وسفارش را چگونه پیاده سازی کنیم. نمی توانیم مثل حالت قبل عمل کنیم، یعنی اینکه بیائیم در داخل هر سفارش ، کد صورتحساب را قرار دهیم.البته در این صورت مشخص میشود که هر سفارش متعلق به کدام صورتحساب است. یعنی رابطه یک به چند بین صورتحساب و سفارش مشخص میشود. اما مشکل اینجاست که هر سفارش چند صورتحساب دارد. برای رفع مشکل از یک موجودیت مشترک استفاده میکنیم. به نام "خرید"

کلید اصلی خرید، کد سفارش + کد صورتحساب است.




در اینجا به این ترتیب اگر بخواهیم کلیه صورتحسابهای مربوط به سفارش با کد ۲۵ را بدست آوریم، به صورت زیر عمل میکنیم:

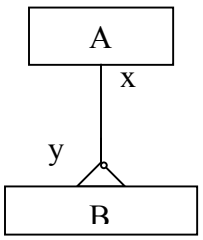
| کد سفارش | کد صورتحساب |
|---|---|
| ۲۵ | ۴۶ |
| ۲۵ | ۱۷ |
| ۲۵ | ۱۸ |

select * from خرید
where کد صورتحساب =46

| | |
|---|---|
| 25 | 46 |
| 34 | 46 |
| 80 | 46 |

بطور کلی چنانچه موجودیت A با موجودیت B در ارتباط قرار داشته باشد،ارتباط میتواند چند به چند، یک به یک، یا یک به چند باشد.

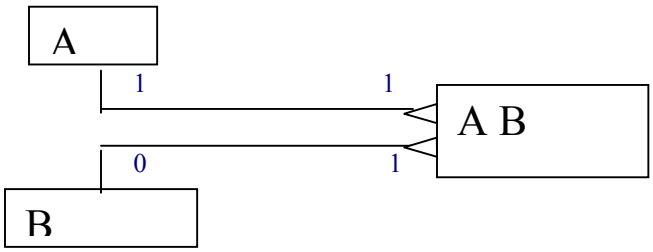
برای نمونه شکل زیر را در نظر بگیرید:




هرAممکن است x صفر یا چند B باشد.

هر Bباید Y یک A باشد.

در حالت کلی رابطه چند به چند بین دو موجودیت A ، B به صورت زیر تبدیل میشود:




رابطه چند به چند به دو رابطه یک به چند تبدیل میشود.

چنانچه کلید اصلی A ،a1 و کلید اصلی B، b1 باشد، آنگاه کلید اصلی AB یک کلید ترکیبی یا composite key است که از ترکیب a1+b1 ایجاد میشود.

جهت ذخیره سازی موجودیتها و به حداقل رساندن فضای لازم از سه مرحله نرمالیزه کردن استفاده میشود.

این صورتحسابها همگی به صورت برگه های مجزا در پرونده ای به نام صورتحساب نگهداری میشوند.می خواهیم پرونده صورتحساب را در داخل کامپیوتر ذخیره کنیم، اما اگر به همان شکلی که در واقعیت وجود دارد، ذخیره کنیم، افزونگی بسیاری در داده های ذخیره شده وجود دارد.

در داخل کامپیوتر هر صورتحسابی به صورت یک رکورد مشخص میشود. اما فیلدهای این رکورد چه میباشد؟!

کدها در دنیای واقعی وجود ندارند.

صورتحساب کالا: کد صورت حساب + تاریخ +

کد مأمور خرید +نام مأمور خرید+ کد سفارش+

نام  سفارش دهنده + مبلغ کل

{کد کالا+...

می بایست فیلدهای فوق،فیلدهای یک رکورد از فایل صورتحساب می باشند. بواسطه اینکه تعداد اقلام کالا در صورتحسابهای مختلف متفاوت است، طول این رکوردها متغییر می باشد.

به عبارت دیگر ممکن است در یک رکورد از صورتحساب، ۵۰ قلم کالا و در رکورد دیگر

یک قلم کالا وجود داشته باشد.

برای این منظور:

## مرحله (۱) نرمالیزه کردن:

کلید اصلی برای موجودیت اگر وجود ندارد،ایجاد شود،بخش تکراری مشخص شود. در اینجا بخش تکراری قلم صورتحساب (قلم کالا) است. برای بخش تکراری کلید اصلی مشخص شود.در اینجا کد کالا میباشد.بخش تکراری از بخش غیر تکراری حذف شود. و به عنوان یک موجودیت جدید مشخص گردد.در اینجا قلم صورتحساب (قلم کالا) از موجودیت صورتحساب حذف میشود. و موجودیت جدید به نام قلم کالا ایجاد میشود.کلید بخش جدا شونده (بخش تکراری)کلیدی ترکیبی است.ترکیب کلید بخش غیرتکراری +کلید بخش تکراری.

در اینجا برای قلم صورتحساب ، بصورت زیر:

قلم صورتحساب (کالا): کدکالا+کد صورتحساب

پس میتوانیم برای خرید کالایی با کد ۴۶ به این صورت عمل کنیم:

Select * from قلم صورتحساب

Where کد صورتحساب=46

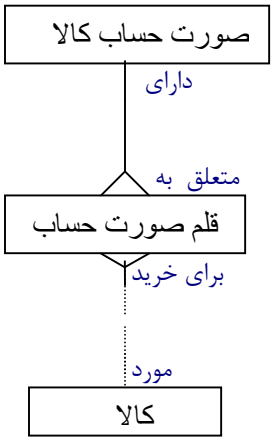
| | |
|---|---|
| ۴۶ | ۱۲ |
| ۴۶ | ۱۸ |
| ۴۶ | ۳۲ |
| ۴۶ | ۸۰۰ |

## مرحله (۲) نرمالیزه کردن:

در مرحله ۲ کلیه فیلدهایی که وابسته به یک بخش از کلید ترکیبی هستند را حذف میکتیم. برای مثال نام کالا در قلم صورت حساب فقط وابسته به کد کالا است اما تعداد هم وابسته به صورتحساب است هم وابسته به کالا

موجودیت جدیدی به دست می آید:

کالا= کد کالا +نام کالا+واحد +موجودی اول هر ماه



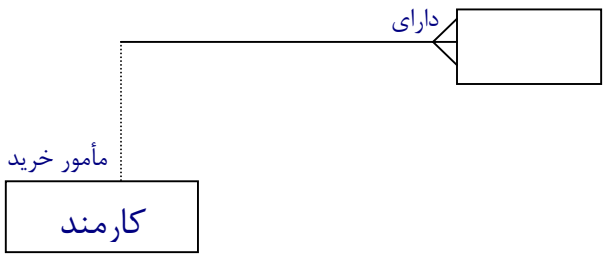

هر قلم صورتحساب باید برای خرید یک کالا باشد

هر کالا ممکن است مورد یک قلم صورتحساب باشد.

## مرحله (۳) نرمالیزه:

تمام فیلدها که از طریق فیلدهای دیگر قابل دسترسی یا محاسبه هستند حذف میگردند. به این ترتیب جمع کل از صورتحساب حذف میگردد. قیمت کل از قلم صورتحساب حذف میشود. نام مأمور خرید که از طریق کد مأمور خرید قابل دسترسی است، حذف میگردد.

صورتحساب کالا: . . . + نام مأمور خرید+ . . .



# مدلسازی رفتاری

رفتار چیست؟ چگونه میتوان رفتار را در زبانهای برنامه سازی پیاده سازی نمود؟

رفتار، شبکه ای است از حالات. رویدادها موجب گذر بین حالات هستند.

رفتار، واکنش در مقابل رویداد است. Object ها تبدیل به کلاس با شعور میشوند.

کلاسها علاوه بر method و صفت یکسری eventدارند که در مقابل آنها واکنش میدهند.

واکنش object در مقابل event وابسته به کلاس نیست.

اصولا شعور ،واکنشی است در مقابل رویداد. یک انسان با شعور در مقابل رویدادها واکنش صحیح را انجام میدهد. واکنش انسان با فراخوانی یک متد صورت میگیرد.

سؤال اینجاست که چگونه در زبانهای برنامه سازی رفتار را مدل میکنند. همانطور که گفتیم، یک object با شعور در مقابل eventها از خود واکنش نشان میدهد.

واکنش object به چه صورت است؟

این واکنش در قالب فراخوانی یک متد صورت میگیرد. اصولا رویدادها به دو دسته تقسیم میشوند. یک دسته توسط سیستم مشخص میشوند، دسته دیگر توسط برنامه نویس تولید میشوند.

برای نمونه به زبان دلفی ارجاع میکنیم.

Eventهایی که سیستم تشخیص میدهد : click

رویدادی که برنامه تشخیص میدهد:

On High press (بالا رفتن دمای مشخص) که برنامه تشخیص می دهد.

برای مثال یکpushbutton را در نظر بگیرید. در مثال زیر برای یک pushbutton یا command1 برای event های click، event mouse move واکنش objectمشخص شده است. واکنش بصورت دو تابع زیر در اینجا تعیین شده:

```
Private sub command1_click()
  Command1.caption =”CANCEL”
End sub
```

```
Private sub command1_mouse move(buttun1)
  Command1.back color=&HFF&
End sub
```

اما در زبان VB این امکان وجود دارد که برنامه نویس خود وابسته به شرایط رویدادی را ایجاد یا در اصطلاح "Raise" نماید.

Destructor: Terminate

Constructor: Initialize

در VB میتوان آرایه را خالی گذاشت و بعد با redirection پر کرد.

```
Private  with events triangle as class1
```

زبان VB را در نظر بگیرید، در VB میخواهیم کلاسی بنام stack تعریف کنیم، به قسمی که اگر در هنگام push کردن مقدار stack پر شود، رویدادover flow اتفاق افتد. برای این منظور در داخل کلاس stack یک event بنام overflow تعریف میکنیم. بدین ترتیب تعریف کلاس به صورت زیر خواهد بود:

```
Public event overflow()
Private  m_size as integer
Private  m_top  as  integer
private  m_store() as integer

Public Function is empty() as Boolean
   If (m_top = 0) then
     Is empty = True
   Else
     Is empty = false
   End If
End function

Public Function isfull() as Boolean
    If m_top=m_size then
       Is full=true
    Else
       Is full=false
    End If
```

```
End function
```

اکنون برای تعریف متغییرها از نوع stack به صورت زیر می بایست :

```
Private withevents s1 as class1          ' or stack
Private withevents s2 as class1

Private sub form_load()
  S1.size=10
  S2.size=100
End sub

Public sub s1overflow()
   Msgbox      "سرریز شده است"
End sub

Public sub s2overflow()
   Msgbox      "سرریز شده است"
End sub
```

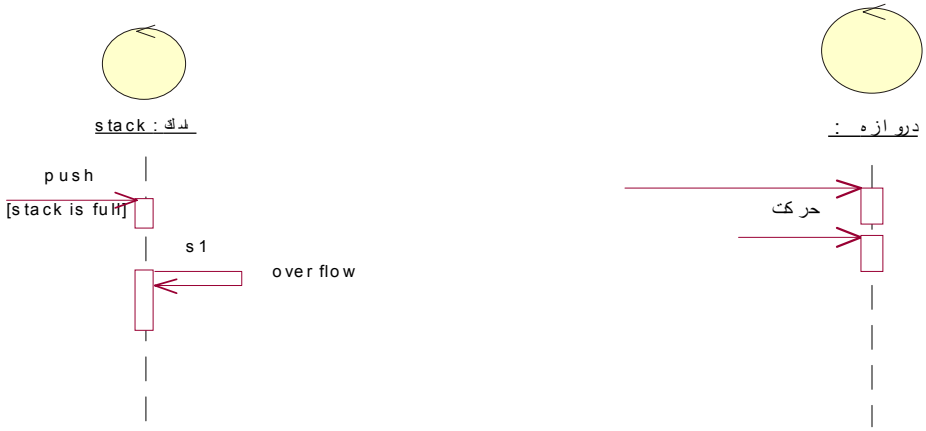
در اینجا چنانچه دستورالعمل برای مثال s1.p را اجرا کنیم و overflow اتفاق بیاقتد تابع s1.overflow اجرا میشود در اینجا هنگام اجرای دستورالعمل push تابع push بوسیله پر بودن stack، raise event مینماید.(raise event overflow) را ایجاد مینماید و همین امر موجب میشود تا تابع s1overflow فعال شود. در واقع s1overflow واکنش شئ s1 موجب میشود که واکنش شئ s1در مقابل رویداد overflowمشخص گردد و هنگامی که برای مثال دستورالعمل s2pushاجرا می شود، اگر در داخل تابع push،raise event overflowاجرا گردد ، تابع s2overflowبه عنوان واکنش s2 در مقابل overflow اجرا میگردد.

حالا مساله این است که چگونه در هنگام مدلسازی بتوانیم این رویدادها واکنش در مقابل رویدادها را مشخص کنیم؟

برای این منظور از دیاگرام های توالی استفاده می کنیم.هر رویداد در واقع یک فراخوانی است.که به یک objectیا کلاس objectوارد می شودو واکنش objectمتدی است که فراخوانی می گردد.

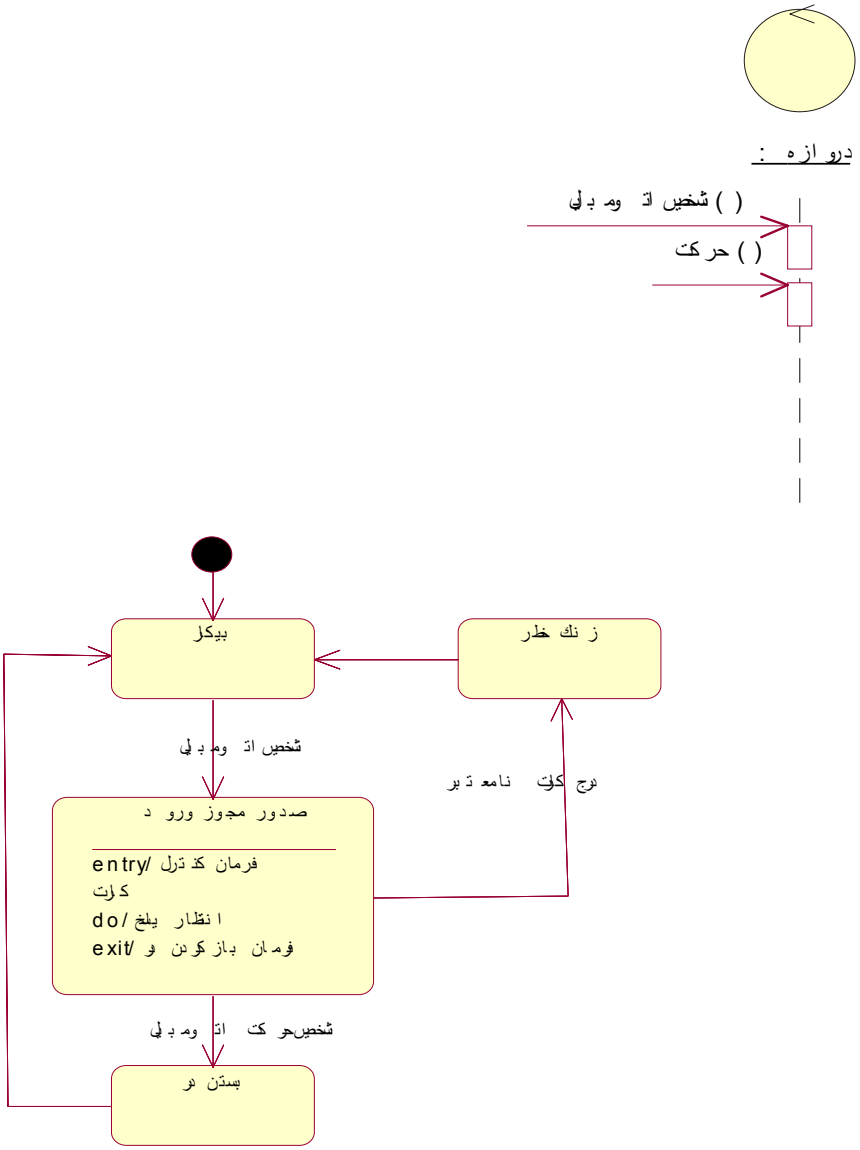
برای نمونه به مثال زیر توجه نمایید:

درsequence diagramای که در شکل صفحه ۴۶۴ کتاب ارائه شده است، Boundary objبنام بیرون محوطه از جنس کلاس "چشم الکتریکی"نزدیک شدن اتومبیل را تشخیص می دهد.بلافاصله در داخل متد"نزدیک شدن"از این obj()(نزدیک شدن. بیرون محوطه)که از پورت" چشم الکتریکی"یکسره در حال خواندن مقدار پورت می باشد،از داخل حلقه خارج می شودومتد( () تشخیص کارت.کارت خوان) فراخوانی می شود.این متد در داخل یک حلقه پورت مربوطه به کارتخوان را آنقدر می خواند تا اینکه عددی که نشانگر تائید کارت توسط کارتخوان است،در این پورت ظاهر شود.حالا این متد به فراخواننده خود یعنی تشخیص اتومبیلreturnمی کند و بلافاصله متد "باز شدن درب" از شئ "درب" فراخوانی میگردد. شئ در نزدیک boundary obj است. چرا که در ارتباط با پورت مربوط به درب قرار دارد و این تابع در پورت مربوط به در هر عددی را قرار می دهد که موجب میشود دستگاه مربوطه فعال گردد و درب ورودی باز شود، به این ترتیب کار در قسمت اول خاتمه می یابد. با دور شدن اتومبیل از دروازه ، تابع دور شدن از دروازه از شئ داخل محوطه موقعیت را تشخیص داده و عملیات مطابق دیاگرام فوق ادامه می یابد. همانگونه که می دانید، واردشدن رویداد به اشیاء موجب تغییر حالت آنها میگردد. در اینجا اگر دقت کنیم ، دو رویداد به "شئ دروازه " وارد میشود. در مورد شئ "stack" رویداد push اتفاق می افتاد.



برای نمونه در بالا با ایجاد push ، چنانچه شرط [stack is full] برقرار باشد، تابع s1_overflow() که در واقع s1 همان my stack است، فعال میگردد. بنابراین در اینجا رویداد push، شرط [stack is full] رخداد،raise overflow و واکنش در مقابل رخداد اجراء my stack.overflow یا s1.overflow است.

حالا برمیگردیم به مثال قبلی ، در اینجا دو رویداد به شئ ورودی وارد شده است.هر رویداد موجب واکنش میگردد و موجب میشود که object از خود واکنش نشان دهد.



در اینجا همانگونه که مشاهده میکنید، مدل رفتاری برای کلاس دروازه، با استخراج این کلاس از داخل دیاگرام توالی مشخص شده دو رویداد، موجب دو تغییر حالت گردیده است.رویداد تشخیص اتومبیل موجب شده که این کلاس از حالت بیکار به حالت " صدور مجوز ورود" وارد شود. اگر دقت کنید، در این حالت سه فعالیت مشخص گردیده اند. یکی on entry (به مفهوم " با ورود به حالت ")، دیگری do (به مفهوم " در داخل حالت" یا "ضمن حالت")، و بالاخره on exit (به مفهوم " خروج از حالت") مشخص شده است.اینها در واقع سه لغت کلیدی در رشنال رز میباشد و هنگام تعریف state ها میتوان این سه برچسب را با استفاده از امکانات رشنال رز مشخص نمود. On entry فعالیت فرمان کنترل کارت انجام میشود. اگر به دیاگرام توالی توجه فرمائید، در داخل متد تشخیص اتومبیل بلافاصله متد " () تشخیص کارت.کارتخوان " فراخوانی میگردد. که این فراخوانی به صورت فرمان کنترل کارت ، on entry مشخص گردیده است، مسلما در اینجا می بایست در انتظار پاسخ از دستگاه کارت خوان بمانیم، لذا این انتظار در دو حالت صورت می گیرد. و به صورت انتظار پاسخ:do در داخل حالت مشخص شده است. اگر به دیاگرام توالی برگردیم ، مشاهده میکنیم که با حرکت اتومبیل، پیام بستن در با فرمان " بستن درب.درب" به انجام میرسد که این رویداد حرکت (پیام حرکت) نیز موجب میشود که به حالت جدید "بستن در" وارد شویم. اگر دقت نمائید، در دیاگرام توالی، فقط حالات مثبت در نظر گرفته میشود. برای مثال، در حالت صدور مجوز ورود ، چنانچه کارت نامعتبر باشد، زنگ خطر به صدا درآورده میشود. در اینجا دایره سیاه،نقطه شروع را مشخص میکند.برای یک کلاس ممکن است در دیاگرامهای توالی متفاوت، دیاگرامهای حالت مختلف ایجاد شود که میبایست همگی آنها را ترکیب نمایید و از نقطه شروع به حالت ابتدائی هریک در هر دیاگرام وارد شد.

در اینجا به شرط اینکه کارت نامعتبر درج شود، زنگ خطر بصدا در می آید.

[شرط در داخل براکت مشخص شده است.]

به صفحه ۴۶۶ کتاب شکل ۴ـ۱۴ توجه کنید.

در مثال فوق ، در ابتدای کار، کتاب در وضعیت " در قفسه" قرار دارد. با رویداد Borrow (یا تقاضای امانت)، نسخه کتاب (a copy) در حالت " در امانت" قرار میگیرد

و بلافاصله نسخه تتا از شئ کتاب درخواست میکند که این نسخه را جزء تعداد نسخه های در امانت قرار دهد. توجه داشته باشید که از کتاب مهندسی نرم افزار ۵ نسخه(copy) ممکن است در کتابخانه موجود باشد. در اینجا شئ کتاب، شاخص کتاب مهندسی نرم افزار است و شئ کپی شاخص نسخه های کتاب است که بطور فیزیکی در کتابخانه موجود هستند. اگر دقت کنید بر روی خط واصل ، فعالیتی که موجب میشود نسخه کتاب از حالت "در قفسه" به حالت " در امانت" وارد شود، فعالیت book.borrowed است.معمولا در حالت کلی بر روی خطوط ارتباطی در دیاگرام حالت، event و در صورت وجود activity/ و در صورت وجود شرط [condition] بر روی خطوط قرار میگیرد.

در دیاگرام توالی که در شکل ۴ـ۱ از فصل ۱ کتاب صفحه ۳۸۴ مشخص شده ،فرم سفارش orderformبا پیام show() به نمایش در می آید(orderform.show )،در واقع این رویداد، موجب میشود که orderform در حالت دریافت مشخصات مشتری،یافتن مشخصات در پرونده مشتریان((customer.find(cno) و بالاخره درج جزییات کلی سفارش،در پرونده سفارش قرار گیرد. شئ خقیثق بعد از اینکه جزئیات کلی سفارش را در پرونده سفارش درج نمود،تابع getarticle()را از orderform فراخوانی می کند.در اینجا نیز سر فلش به سمت کلاس Torderformمی باشد .بدین ترتیب دو رویداد به این کلاس وارد می شوند.یکی تقاضای نمایش(show)ودیگری تقاضای دریافت جزئیات قلم سفارش(مواد،خودکار) ((getarticle(). getarticle() در واقع رویداد دومی است که موجب تغییر حالت این کلاس میگردد. بنابراین دو رویداد ورودی مشخص می شود:

هر متد وارد شونده به کلاس ،باعث تغییر حالت می شود.

/rose view/logical view/state dig....

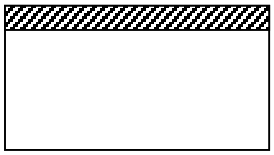
## دیدگاه قطعات component view

در component viewقطعات برنامه مشخص می شوند.بدین ترتیب که مشخص می کنیم یک پروژه از چه فایلهایی تشکیل شده است.برای مثال،زبان پاسکال را به خاطر بیاورید.

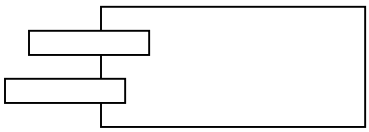
یک پروژه پاسکال(prj) از تعدادی unit به اضافه یکmain program تشکیل میشود. در یک سیستم گسترده معمولا برای هر کاربر یک برنامه مستقل نصب می شود و همه کاربرها از طریق یکبانک اطلاعاتی مشترک با یکدیگر داده رد وبدل می کنند.برنامه ها در اینجا مجزا هستند .هر کدام به صورت یک project برای نمونه در سیستم کارت ساعت ،سه زیر سیستم guard modules (پیمانه نگهبانی) employment module (پیمانه کار گزینی)وبالاخره employees module(پیمانه واحدهای کارمندی)مشخص شده است.برای employment module، پیمانه ها به صورت زیر مشخص شده اند:

ص ۲۵۶ کتاب

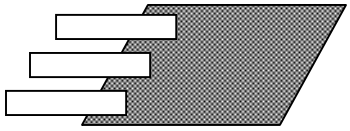
ما در واقع فایلهای برنامه را مشخص می کنیم.هر فایل یک componentبه عنوان برنامه اصلی مشخص می شود.شکل برنامه اصلی بدین صورت است:



قطعه معمولی به صورت زیر می باشد:



نوع دیگر از قطعه برنامه ها بنامtaskمی باشد.



Taskها قطعاتی هستند که به صورت یک trend به موازات برنامه اصلی به اجرا در می آیند.در واقع در این مرحله هر قطعه یک فایل برنامه می باشد. برای آن می بایست اولا زبان بر نامه سازی را مشخص نمود.برای نمونه ،برای سیستم ثبت سفارش،همانگونه که در کتاب مشخص شد،یک component بنام ..order subsys مشخص شده است.با clickکردن روی آن یک فرم باز می شود .در داخل آن می توانید زبان برنامه سازی را

مشخص کنید. برای مثال vb، تحت گزینه ای بنام realize لیست تمامی کلاس ها مشخص می شود.

با R_click بر روی هر کلاس ، می توانید آن کلاس را به آن component، assignکنید.در واقع بر نامه شئ گرا به ما می گوید که هر module(پیمانه(قطعه))شامل تعدادی کلاس است .

برای هر قطعه کلاسهایی را انتخاب می کنیم که در ارتباط تنگاتنگ با یکدیگر باشد .

برای مثال پیمانه امور آموزش دارای objهایی است که بسیار یکدیگر را فرا خوانی می کنند.پس معیار assignکردن کلاسها به یک قطعه میزان همکاری کلاسها با یکدیگر است.

برای مثال در سیستم آموزش،قطعه ثبت نام ،شامل کلیه کلاس هایی است که در ارتباط تنگاتنگ برای عمل ثبت نام هستند.بعد از اینکه به کلیه قطعه ها ،کتاسها تخصیص شد،مرحلهforward engineering یا مهندسی رو به جلو برای تولید کداز مدل آغاز می شود.

توجه کنید باید کلیه کتاسها،متد ها،فیلدهای کلاس ،پارامتر های ورودب وخروجی متد ها ،قبلا به زبان فارسی ،کاملا توضیح داده شده باشد.برای مثال اگر کلاسی به نام orderدارید،باید بر روی کلاس کلیک کرده باشید و شرحی برای کلاس نوشته باشید.تمام این comment ها در هنگام cod generation در کد مثلا vb گنجانده می شود.برای تولید کد از منوی tools گزینهvb و سپس گزینه update code را انتخاب نمایید. به این ترتیب سیستم ها شروع به تولید کد vbمی نمایند.اگر کلاس های نوع boundary فرم می باشند، بایستی stereotype های آنها را از boundary به فرم تبدیل کنید تا در vb به صورت فرم نمایش داده شوند.

# The End

Successful...