

به نام خدا

## طراحی نمودار جریان داده و نمودار فعالیت

استاد راهنما: دکتر حمید نوری

### نمودار جریان داده چیست؟

یک نمودار جریان داده راه گردش اطلاعات را در یک فرآیند یا سیستم نشان می‌دهد. این اطلاعات شامل داده‌های ورودی و خروجی، ذخایر داده و زیرپرونده‌ها است که داده در آن آنها حرکت می‌کند. نمودار جریان داده با استفاده از نمادهای استاندارد اشخاص و نهادهای مختلف و روابط آنها را توضیح می‌دهد.

نمودار جریان داده به صورت تصویری سیستم و فرآیندهایی را ارائه می‌دهد که توضیح آن در چند سطر کار مشکلی است. شما می‌توانید از این نمودارها برای به نقشه درآوردن و به تصویر کشیدن سیستم‌های موجود و اجرای بهتر سیستم‌های جدید استفاده کنید. تصویرسازی هر عامل تشخیص ناکارآمدی‌ها را راحت‌تر می‌کند و بهترین سیستم ممکن را برای شما به ارمغان می‌آورد.

### نمودار جریان داده فیزیکی و منطقی

قبل از اینکه نمودار جریان داده خود را درست کنید، باید بدانید که نمودار جریان داده فیزیکی برای شما مناسب است یا نمودار جریان داده منطقی.

نمودار جریان داده منطقی بیشتر بر روی این موضوع تمرکز دارد که «چه اتفاقی» در یک گردش اطلاعات رخ می‌دهد: چه اطلاعاتی جابه‌جا می‌شود؟ چه اشخاصی این اطلاعات را دریافت می‌کنند؟ چه فرآیندهای عمومی‌ای رخ می‌دهد؟ و ... فرآیندهایی که در نمودار جریان داده منطقی شرح داده می‌شوند، فعالیت‌های تجاری‌اند؛ این نوع نمودار وارد جنبه‌های فنی یک فرآیند یا سیستم نمی‌شود. درواقع این نمودار باید به قدری قابل درک باشد که افراد و کارمندان غیرفنی سازمان و شرکت شما بتوانند این نمودارها را بفهمند و درک کنند.

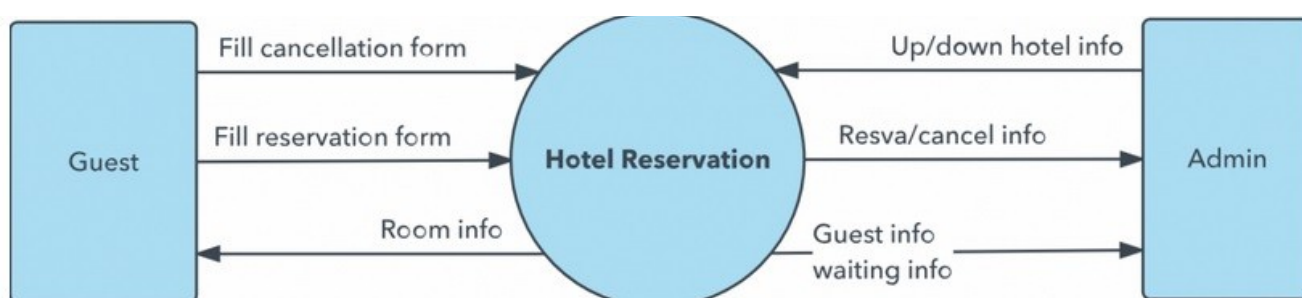
نمودار جریان داده فیزیکی بیشتر بر روی این موضوع تمرکز دارد که «چگونه» این اتفاق‌ها در یک گردش اطلاعات رخ می‌دهد. این نمودارها نرم‌افزارها، سخت‌افزارها، فایل‌ها و پوشه‌ها و افرادی را که درگیر این گردش اطلاعات می‌شوند مشخص می‌کند. یک نمودار جریان داده که جزئیات را به دقت بیان کرده باشد، می‌تواند نیازهای یک سیستم داده را به خوبی نشان دهد.

اما نمودارهای جریان داده فیزیکی و منطقی می‌توانند گردش اطلاعات یکسانی را نشان دهند؛ حتی ممکن است شما به هر دو نمودار نیاز داشته باشید.

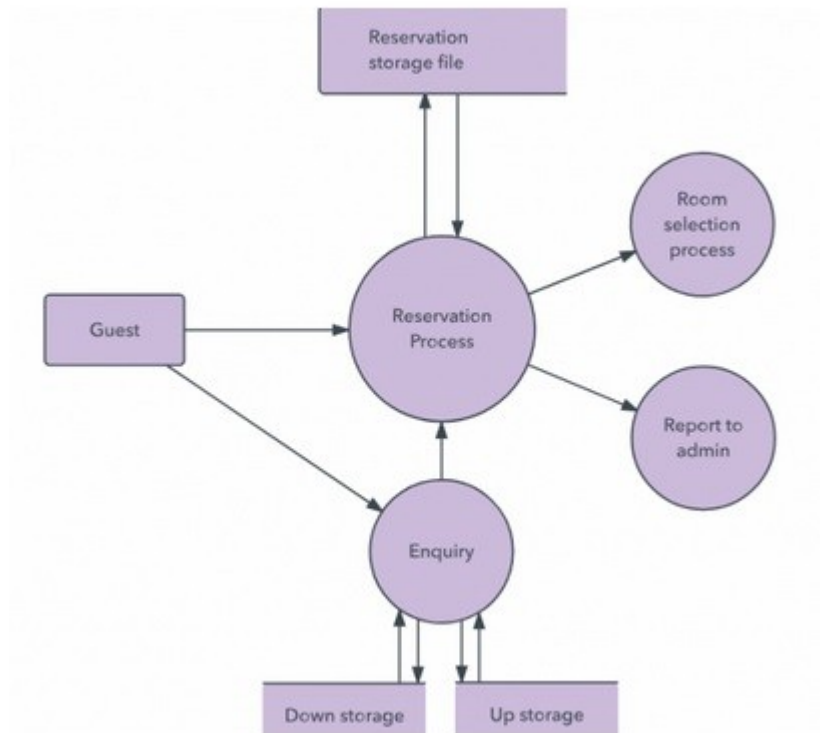
## سطوح نمودار جریان داده

نمودارهای جریان داده بر اساس سطوح متعددی طبقه‌بندی می‌شوند. برای شروع از سطح ابتدایی یعنی سطح ۰ شروع می‌کنیم. با بالا رفتن سطوح، نمودار پیچیده‌تر می‌شود. زمانی که می‌خواهید نمودار جریان داده خود را درست کنید، باید تصمیم بگیرید که کدام سطح برای شما و سیستم شما مناسب‌تر است.

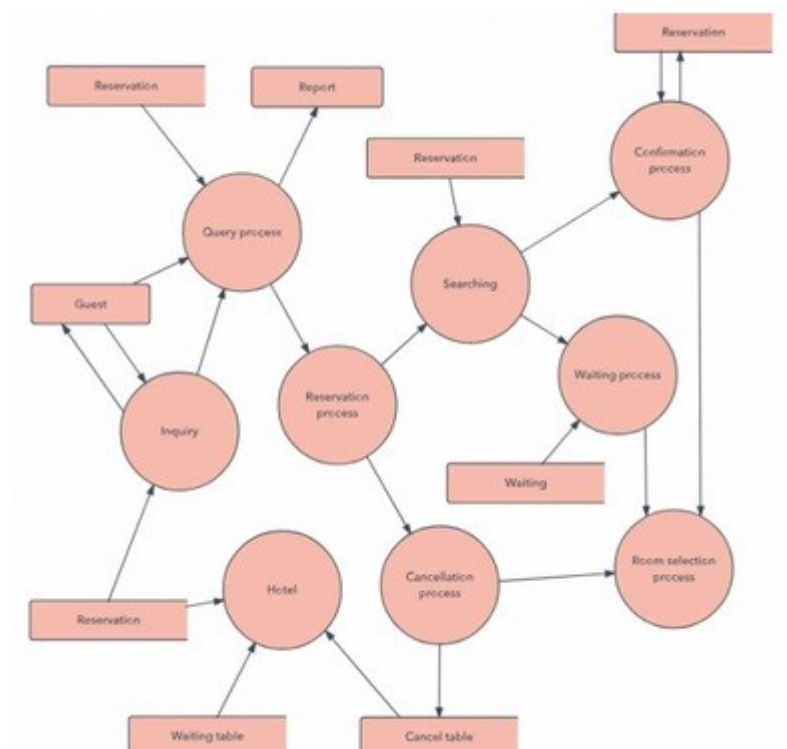
**سطح صفر نمودار جریان داده** پایه‌ترین سطح این نمودار است. این نمودار اطلاعات بسیاری ساده و با جزئیات اندک ارائه می‌دهد که درک راحتی دارد. این نوع نمودار یک گره پردازشی را به همراه اتصال آنها به شخص و نهاد خارجی نشان می‌دهد.



**سطح ۱ نمودار جریان داده** بازهم یک نمودار ساده است که یک نگاه کلی به موضوع دارد، اما نسبت به نمودار سطح ۰ جزئیات بیشتری را ارائه می‌دهد. در نمودار سطح ۱ یک گره پردازشی که در نمودار سطح ۰ وجود داشت به زیرپروسه‌ها تقسیم و تبدیل می‌شود. همان‌طور که پروسه‌ها و فرآیندها اضافه می‌شوند، نمودار به جریان داده و ذخیره داده نیاز دارد تا آنها را به هم متصل کند.



نمودار جریان داده سطح ۲ خیلی ساده فرآیندها و پروسه‌ها را به زیرپروسه‌ها تقسیم می‌کند. این سطح از جریان داده می‌تواند در سطح ۳ هم قرار بگیرد، اما به ندرت این اتفاق می‌افتد. نمودار سطح ۲+ به قدری به جزءهای کوچک‌تری تقسیم شده است که دیگر نیازی به تقسیم بیشتر نیست.



## نمادها و نشانه‌گذاری‌های مورد استفاده در دیاگرام جریان داده

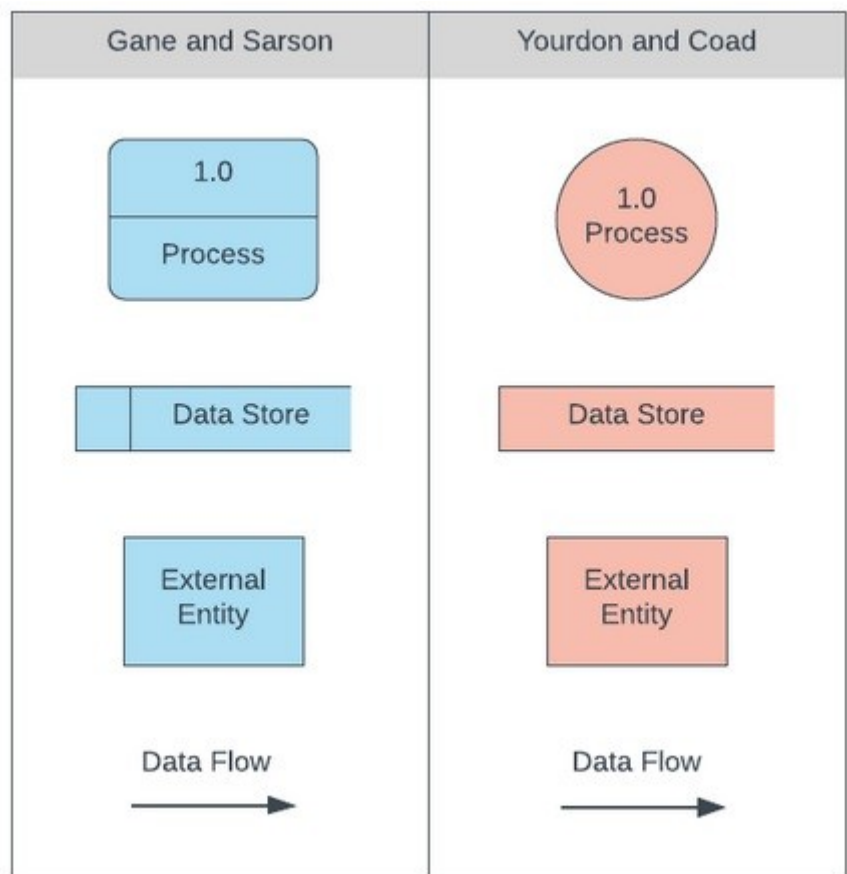
دو سیستم نمادگذاری رایج به نام سازندگان آن‌ها نامگذاری شده‌اند:

- یوردون و کوآد (Yourdon and Coad)
- یوردون و دمارکو (Yourdon and DeMarco)
- گین و سارسون (Gane and Sarson)

یک تفاوت اصلی در نمادهای آن‌ها این است که یوردون-کوآد و یوردون-دمارکو از حلقه‌ها برای پردازش‌ها استفاده می‌کنند، در حالی که سارسون و گین از مستطیل‌هایی با گوشه‌های گرد استفاده می‌کنند که گاهی اوقات آن‌ها را لوزی نیز می‌نامند. تنوع نماد دیگری نیز در استفاده وجود دارد، بنابراین نکته مهمی که باید به خاطر بسپارید این است که شکل‌ها و نگاشت‌هایی که برای برقراری ارتباط و همکاری با دیگران استفاده می‌کنید، واضح و ثابت باشند.

با استفاده از هر قانون یا دستورالعمل کنوانسیون دیاگرام جریان داده، نمادها چهار مولفه نمودارهای جریان داده را به تصویر می‌کشند.

- **نهاد خارجی:** سیستم بیرونی است که داده‌ها را می‌فرستد یا دریافت کرده و با سیستم در حال نمودارسازی ارتباط برقرار می‌کند. آن‌ها منابع و مقصد ورود اطلاعات یا خروج از سیستم هستند. ممکن است یک سازمان یا شخص خارجی، یک سیستم رایانه‌ای یا یک سیستم کسب و کار باشند. آن‌ها همچنین به عنوان خاتمه‌دهنده، منبع یا بازیگر شناخته می‌شوند. این نهادها به طور معمول در لبه‌های نمودار ترسیم می‌شوند.
- **فرایند:** هر فرایندی که داده‌ها را تغییر داده و یک خروجی تولید می‌کند. این فرایند ممکن است محاسبات را انجام دهد، داده‌ها را بر اساس منطق مرتب کرده، یا جریان داده را بر اساس قوانین کسب و کار هدایت کند. از یک برچسب کوتاه برای توصیف روند استفاده می‌شود، مانند “ارسال پرداخت.”
- **ذخیره داده:** پرونده‌ها یا مخازنی که اطلاعات را برای استفاده‌های بعدی مانند [پایگاه داده](#) یا فرم عضویت در خود دارند. هر مخزن داده یک برچسب ساده مانند “سفارشات” را دریافت می‌کند.
- **جریان داده:** مسیری که داده بین نهادهای خارجی، پردازش‌ها و ذخیره داده‌ها طی می‌کند. این جریان، رابط بین اجزای دیگر را به تصویر می‌کشد و با پیکان نشان داده می‌شود. به طور معمول با نام داده کوتاه، مانند “جزئیات صورتحساب”، برچسب‌گذاری می‌گردد.



### چگونه یک نمودار جریان داده درست کنیم؟

حالا که کمی در مورد جریان داده و نمودارهای جریان داده اطلاعات کسب کردیم، می‌خواهیم به سراغ درست کردن این نمودارها برویم. فرآیند درست کردن نمودار جریان داده ۵ مرحله دارد:

#### ۱. تشخیص ورودی‌ها و خروجی‌های اصلی در سیستم

تقریباً هر فرآیند و پروسه یا سیستمی با یک ورودی از یک شخص یا نهاد شروع می‌شود و با خروجی داده به دیگر شخص یا نهاد و یا پایگاه داده تمام می‌شود. تشخیص چنین ورودی و خروجی‌هایی یک دید خوب به سیستم‌تان می‌دهد؛ درواقع وظیفه مهم سیستم شما مشخص می‌شود. باقی نمودار شما بر روی این عناصر ساخته می‌شود، بنابراین شناخت آنها بسیار مهم است.

## ۲. ساخت یک نمودار سطح •

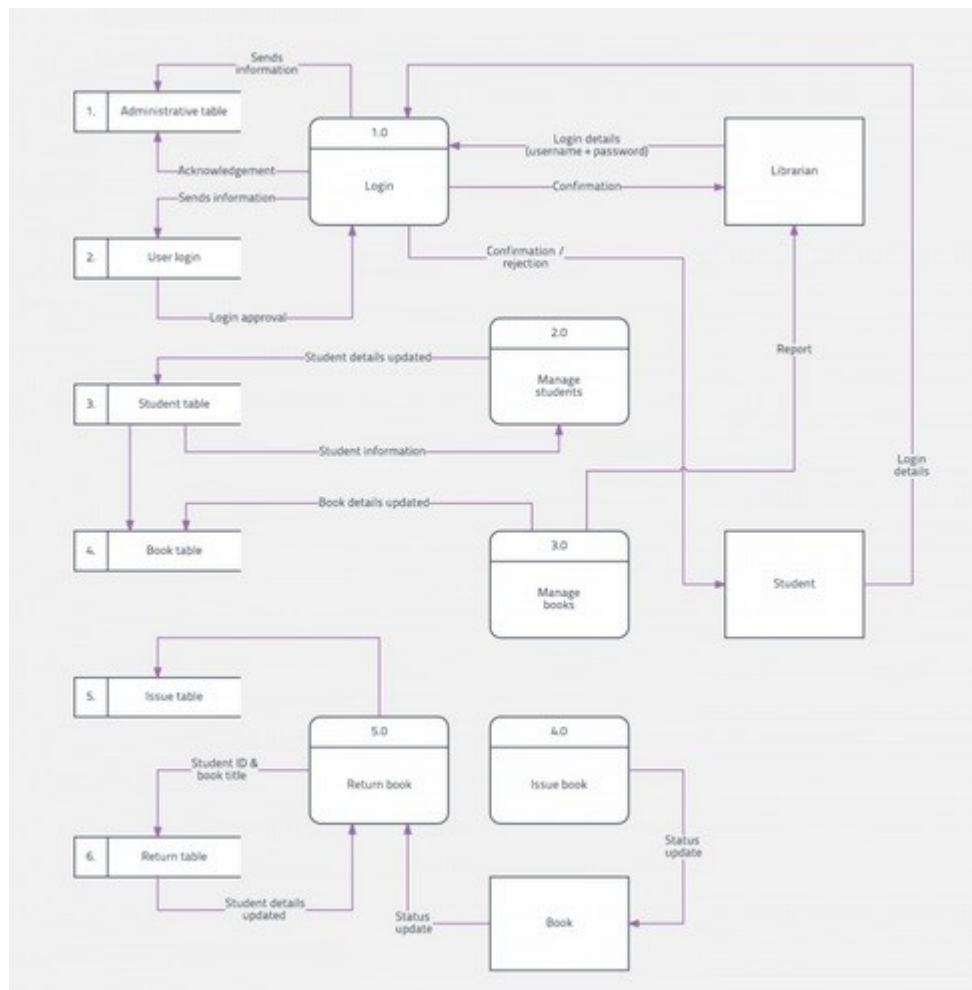
به محض اینکه ورودی و خروجی‌های اصلی را پیدا کردید، ساخت نمودار سطح صفر بسیار ساده می‌شود. یک گره پردازشی را بکشید و آن را به شخص و نهاد خارجی مربوط کنید. این گره ارائه دهنده بیشتر اطلاعات فرآیندهایی است که از ورودی به خروجی در جریان هستند.

## ۳. نمودار سطح • را به نمودار سطح ۱ توسعه دهید

نمودار سطح • شما اطلاعات زیادی را در بر نمی‌گیرد؛ باید آن را به زیرپرونده‌ها تبدیل کنید. در نمودار سطح ۱ باید گره‌های پردازشی، پایگاه‌های داده اساسی و همه اشخاص و نهاد خارجی را بگنجانید. به گردش اطلاعات خود نگاه کنید و از خود بپرسید: اطلاعات از کجا آغاز می‌شود و قبل از ذخیره آنها باید چه اتفاقی برایشان بیفتد؟

## ۴. نمودار را به سطح ۲ گسترش دهید

از فرآیند مرحله قبل استفاده کنید تا بتوانید جزئیات بیشتری را در نمودار جریان داده خود بگنجانید. پرونده‌های سطح ۱ می‌تواند به زیرپرونده‌های بیشتری تبدیل شود. یادتان باشد که اطلاعات و داده‌های لازم را به نمودار خود اضافه کنید.



## خطا و اشتباه در نمودار جریان داده ها

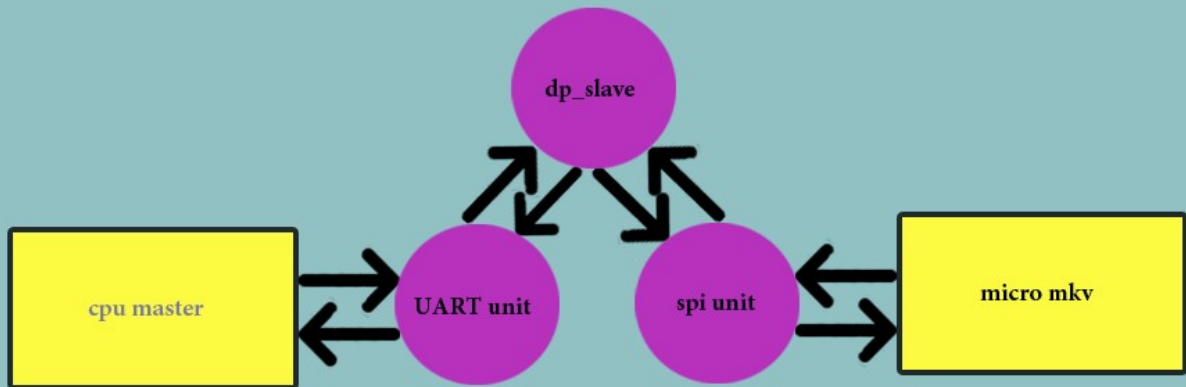
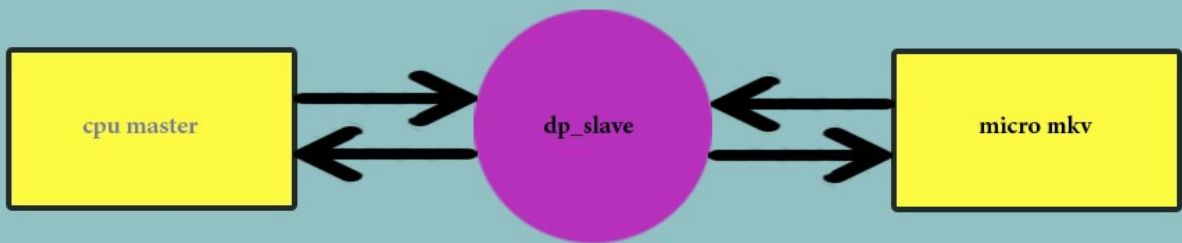
در رسم این نمودار ها بایستی از خطا و اشتباه جلوگیری گردد.

۱- به لحاظ منطقی جریان اطلاعات نمی تواند صرفا وارد یک فرآیند شود چرا که لزوما باید اطلاعات خارج هم شود.

۲- برعکس آن هم نمی تواند رخ دهد یعنی از فرآیندی که هیچ اطلاعاتی وارد نمی شود نمی تواند اطلاعات خارج شود.

۳- موجودیت های خارجی نمی توانند به صورت مستقیم به هم ارتباط پیدا می کنند.

۴- بین دو data store هم نمی تواند ارتباط مستقیم وجود داشته باشد.





## نمودار فعالیت یا activity diagram چیست؟

در سیستم ما تعدادی از اجزا دارای فعالیت های مشخصی هستند و اجرای هماهنگ این اجزا باعث رسیدن به هدف اصلی سیستم می شود. اکتیوییتی دیاگرام Activity Diagram به شما خواهند گفت که این اجزا باید چگونه کار کنند تا نیازمندی های سیستم برآورده شود. در نمودار اکتیوییتی قرار است بصورت سطح بالا مشخص شود که چه فعالیت هایی باید انجام شود تا هر کدام از این اجزا وظیفه خود را به درستی انجام دهند. در این دیاگرام مراحل و زنجیره انجام فعالیت ها مشخص خواهند شد.

قواعدی که در اکتیوییتی دیاگرام استفاده می شوند بسیار ساده و مانند فلوجارت هستند و علائم خیلی قابل فهمی برای عموم دارند و به همین دلیل این نمودارها بهترین دیاگرام برای ارائه به ذینفعان سطح بالای سیستم ها هستند. در اکتیوییتی دیاگرام شما قرار است مراحل که برای انجام هر فعالیت لازم است را مدلسازی کنید.

شروع هر نمودار اکتیوییتی یک دایره توپر هست که به آن Initial Node گفته می شود.

سپس با فلش هایی حرکت میکنیم که این فلش ها رو Edge یا Path میگوئیم.

باکس های مستطیلی که Action گفته می شود و برای نشان دادن کاری که در هر قدم باید انجام شود. در واقع Action ها قدم های فعال در یک Process هستند و هر اکشنی میتواند یک بخش محاسباتی یا منطقی باشد.

علامت لوزی یعنی تصمیم گیری که نشان میدهد این جریان بعد از این نقطه ممکن است دو یا چند حالت برایش اتفاق بیافتد.

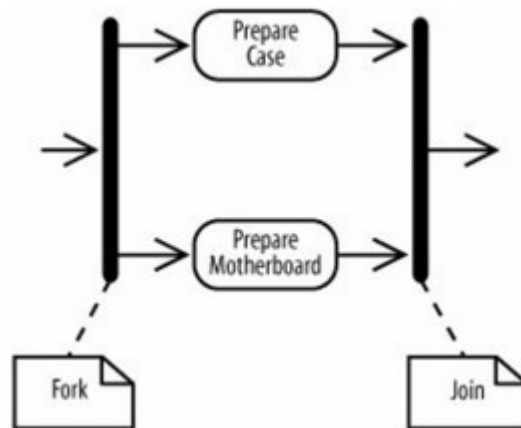
در نهایت Final Node رو داریم که یک دایره توپر است که دور آن یک دایره گرد دارد.

می توان دور هر نمودار Activity یک باکس هم کشید که به آن Activity Frame گفته می شود و کشیدن آن اختیاری است و زمانی که در یک صفحه بیش از یک نمودار را اکتیوییتی قرار بگیرد از این باکس استفاده می کنیم و داخل این اکتیوییتی فریم هم اسم اکتیوییتی را بصورت بولد مینویسیم.

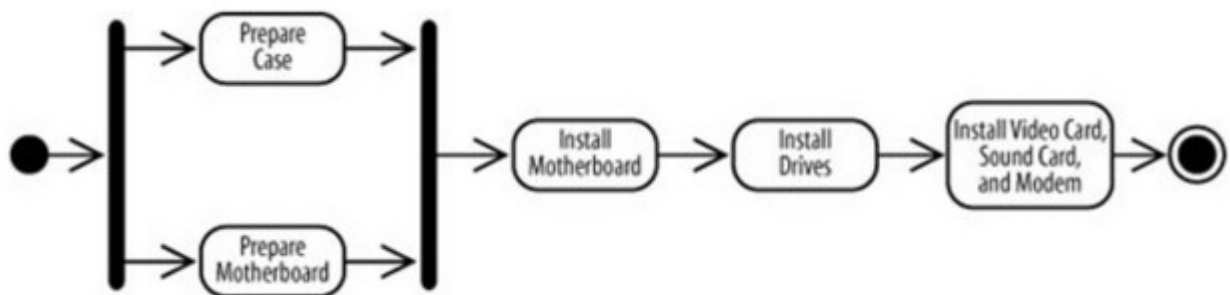
## مفهوم Concurrency و یا Parallel در Activity Diagram

در مدلسازی uml و نمودار اکتیوییتی گاهی بعضی کارها و اکشن ها میتوانند همزمان با هم انجام شوند و نیازی و ربطی به یکدیگر ندارند و بصورت همزمان میتوانند انجام شوند که به اینها اکشن های concurrent و یا parallel می گوئیم که در نمودار اکتیوییتی نمایش اکشن های پارالل را با Fork و Join انجام میدیم که هر دو یک خط عمومی پررنگ صاف هستند و فرقتشان در این است که در fork که یک چنگال است یک

edge وارد میشه و چند تا edge که موازی و همزمان می‌توانند اجرا شوند خارج می‌شوند. و در حالت join نیز چند تا edge وارد می‌شوند و یکی خارج می‌شود.

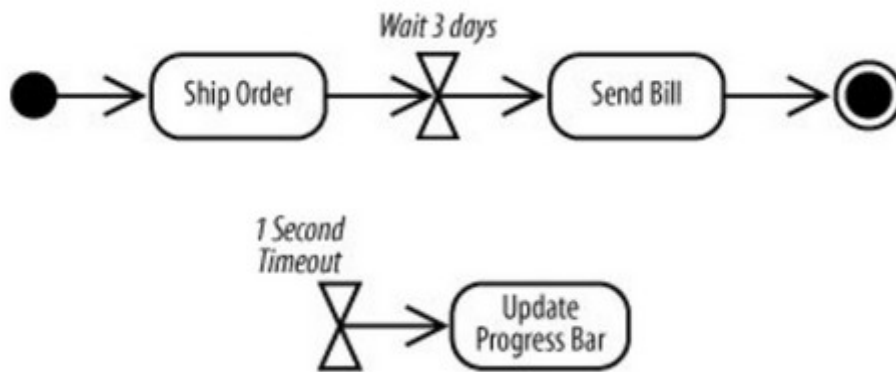


یعنی بعد از هر Fork فلوی ما به چند تا فلوی همزمان میشکند و بعد از هر join چند فلوی همزمان میشن یک فلو.



## زمان در Activity Diagram

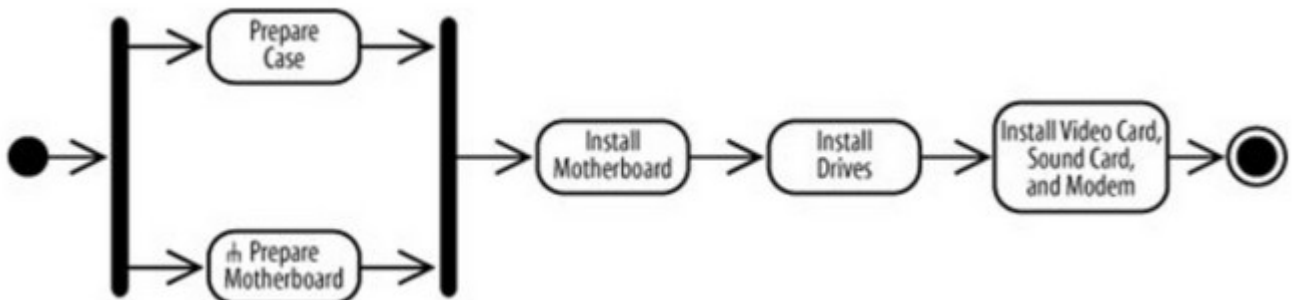
در نمودار اکتیوییتی بعضی مواقع زمان یک عامل در مدلسازی شما محسوب میشود. مثلاً نیاز است بعضی جاها یک وقفه زمانی ایجاد شود. به عنوان مثال ۳ روز بعد از اجرای فعالیت خاصی باید یک ارسال محصول اتفاق بیافتد یا اینکه مثلاً هر ۳ ماه یکبار باید یک آپ‌گیری انجام می‌شود. علامت زمان در نمودار اکتیوییتی بصورت ساعت شنی است که روی آن میزان زمان را مینویسیم مثلاً wait 3 weeks نوشته می‌شود. اگر برای time ورودی خاصی وجود نداشته باشد این یعنی اکشن بعد از آن بصورت recurring انجام و تکرار خواهد شد. مثلاً هر ۱ ثانیه یکبار.



استفاده از time بدون ورودی یکی از روش های شروع یک اکتیویتی دیاگرام است.

### فراخوانی سایر اکتیویتی ها

وقتی ما جزئیات بیشتری به activity diagram اضافه میکنیم کم کم نمودار ما بسیار طولانی و بزرگ میشود و یا اینکه یک سری از کارها در اکتیویتی در جاهای دیگر نیز ممکن است اتفاق بیفتند. در چنین شرایطی برای افزایش readability میتوانیم این بخش جزئیات یا تکراری را در یک دیاگرام دیگر ایجاد کنیم و در این دیاگرام اصلی فراخوانی کنیم و کمک کنیم دیاگرام اصلی ما مرتب تر باشد.

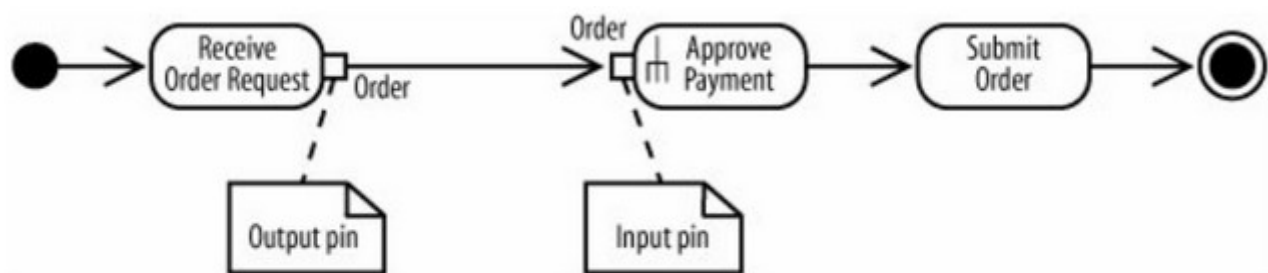


همانطور که در شکل بالا میبینید اکشن prepare Motherboard با علامت چنگک نشون میده که این خودش یک اکتیویتی دیاگرام جدا هست که اینجا include شده. خود اون اکتیویتی مجزا دقیقا نقطه شروع و پایان خواهد داشت

### آبجکت ها Objects در نمودار اکتیویتی دیاگرام

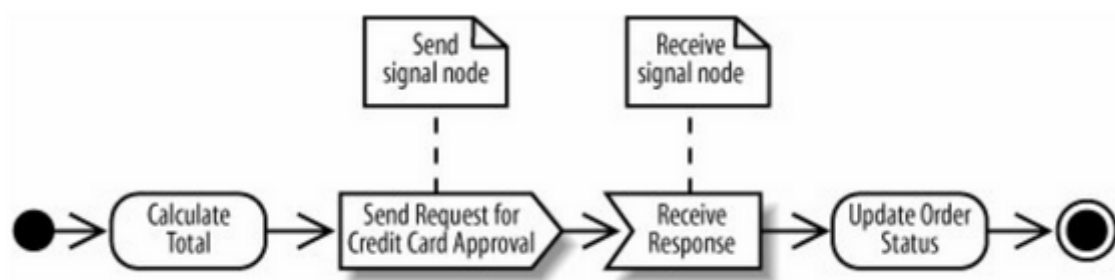
بسیاری جاها در نمودار اکتیویته ما نیاز داریم که برای انجام یک کاری دسترسی به یک اطلاعاتی داشته باشیم. مثلا برای بررسی سوابق مشتری نیاز داریم اطلاعات مشتری را داشته باشیم. این حالت را دیتا آبجکت میگوییم. این آبجکت ها قرار نیست فقط دسترسی به دیتا یا یک چیز نرم افزاری باشند، بعضی وقتها آبجکت ها میتوانند حتی یک کار فیزیکی باشند که باید با دست انجام شود. آبجکت یعنی در اون node این موضوع فراهم است. این دیتا فراهم است یا اینکار فیزیکی فراهم است و...

آبجکت ها را با مستطیل با گوشه های ۹۰ درجه نشون میدیم. یک روش بهتر برای نشان دادن آبجکت ها استفاده از پین است. یعنی یک مربع کوچک که به ابتدا یا انتهای یک Action ای چسبیده است.



## دریافت و ارسال Signal ها در Activity Diagram

اکتیویته ها ممکن است با سایر موجودیت های بیرونی مثل کاربران، افراد، سرویس ها، سیستم ها، وب سرویس ها و ... بیرونی در ارتباط باشند تا بتوانند کاری را انجام دهند. در نمودار اکتیویته Signal ها برای ارتباط سیستم با اجزای بیرونی استفاده میشود. سیگنال ها message هایی هستند که میتوانند ارسال یا دریافت شوند. مثل ارسال request به یک سرویس بیرونی، دریافت کلیک کاربر و یا ارسال یک نوتیفیکیشن به گوشی کاربر و...

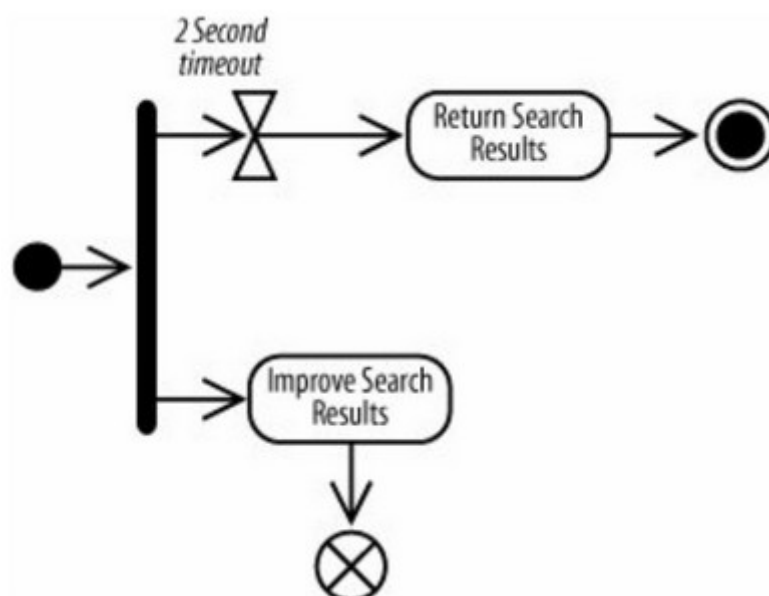


سیگنال دریافت Recieve Signal که از بیرون می آید باعث میشود که یکی از اکشن های اکتیویته فعال شود و این اکشن پذیرنده سیگنال دقیقا میداند که با سیگنال قرار است چه کاری انجام دهد و قرار است سیگنال را چه زمانی دریافت کند. سیگنال ارسال Send Siagnal پیامی است که از طرف اکتیویته ما به یک موجودیت بیرونی ارسال میشود. معمولا وقتی پذیرنده بیرونی ما یک سیگنال از اکتیویته ما میگیرد response را به اکتیویته ما میفرستد ولی ما این پاسخ را دیگر در دیاگراممان مدل نمیکنیم.

اگر سیگنال دریافت را بدون ورودی شروع کنیم یعنی اینکه کل اکتیویتهی ما برای شروع نیاز مند به یک سیگنال بیرونی است و نقطه شروعش همین است.

## شروع و پایان یک اکتیویتهی

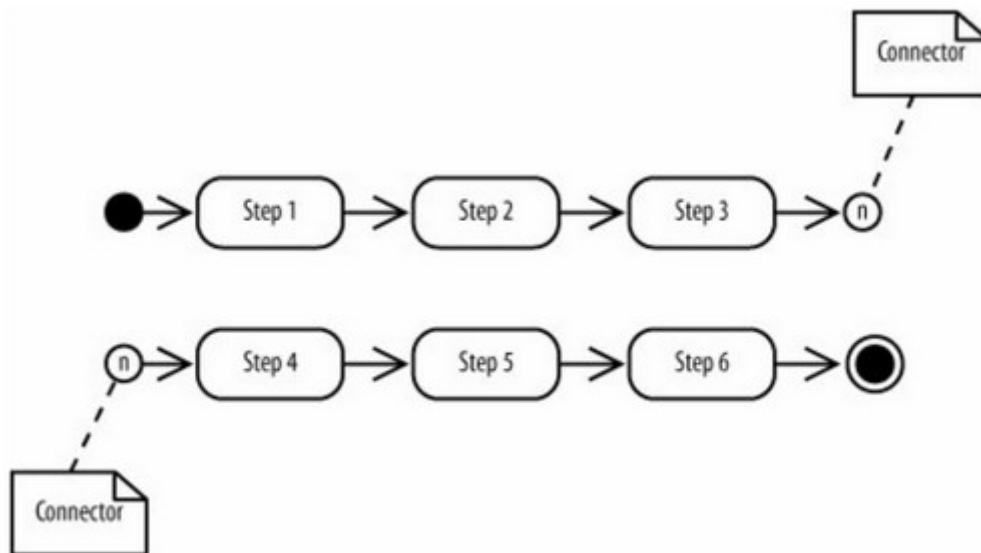
چندین راه برای شروع یک اکتیویتهی وجود دارد که روتین تین و ساده ترینش همون نقطه شروع با دایره توپر است. ولی راه های دیگه ای مانند سیگنال ورودی، تایم نیز هستند.



برای پایان در اکتیویتهی ها یک حالت عمومی نقطه پایان اصلی و روتین هست ولی حالت های دیگه ای هم هست که یکی از فلوهای داخل اکتیویتهی تمام بشه بدون اینکه کل اکتیویتهی تمام بشود که در این حالت از یک دایره و یک ضربدر کامل داخلش استفاده میکنیم و یا اینکه یک وقفه interrupt ای در اکتیویتهی بخوایم ایجاد کنیم.

## اکتیویتهی های طولانی و شکستن سطر

بعضی وقتها اکتیویتهی ها طولانی هستند و در یک سطر جا نمیشن و برای همین منظور مجبور هستیم سطر را بشکنیم. در این مواقع از کانکتورها Connectors استفاده میکنیم. روی کانکتور اسمشو که معمولاً یک کاراکتر هست رو مینویسیم و اونجا سطر رو میبندیم و در سطر بعدی با اسم همون کانکتور شروع میکنیم.

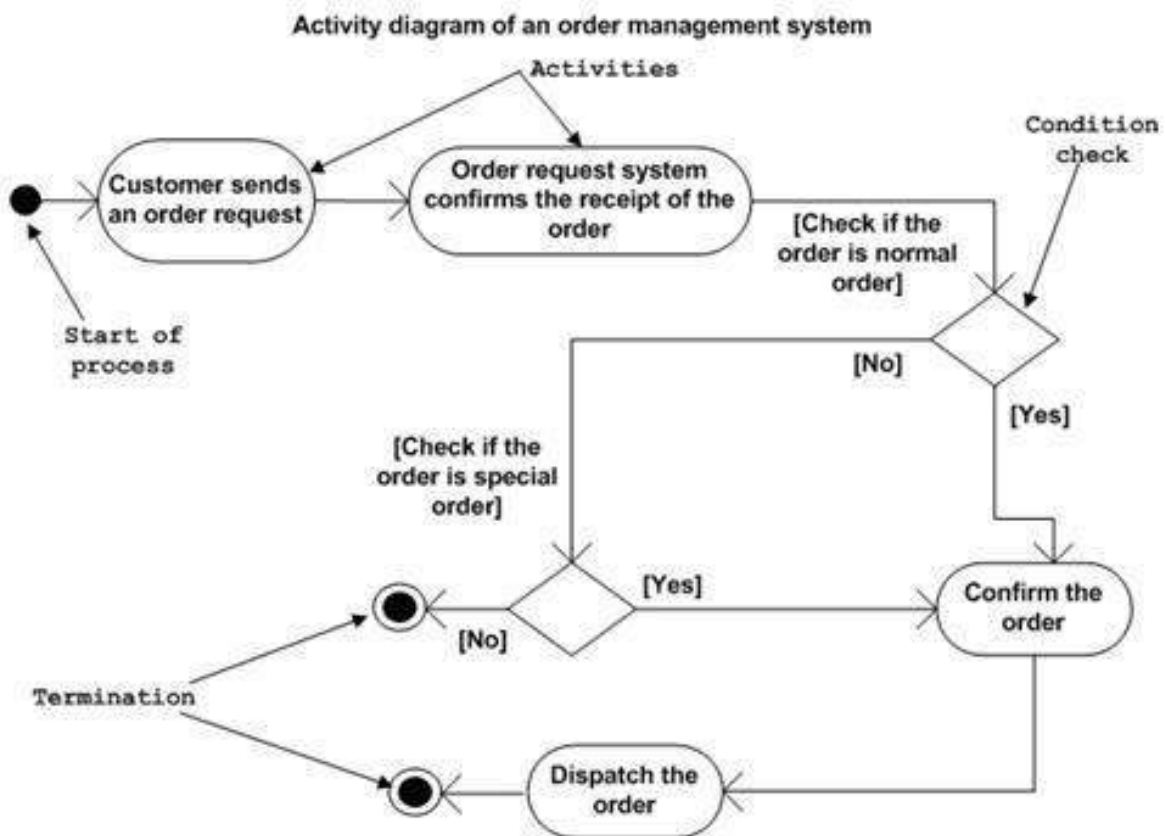


در زیر نمونه ای از نمودار activity برای سیستم مدیریت سفارش وجود دارد. در نمودار ، چهار activity مشخص شده است که با شرایطی همراه هستند.

نمودار زیر با چهار فعالیت اصلی ترسیم شده است .

- ارسال سفارش توسط مشتری
- دریافت سفارش
- تأیید سفارش
- ارسال سفارش

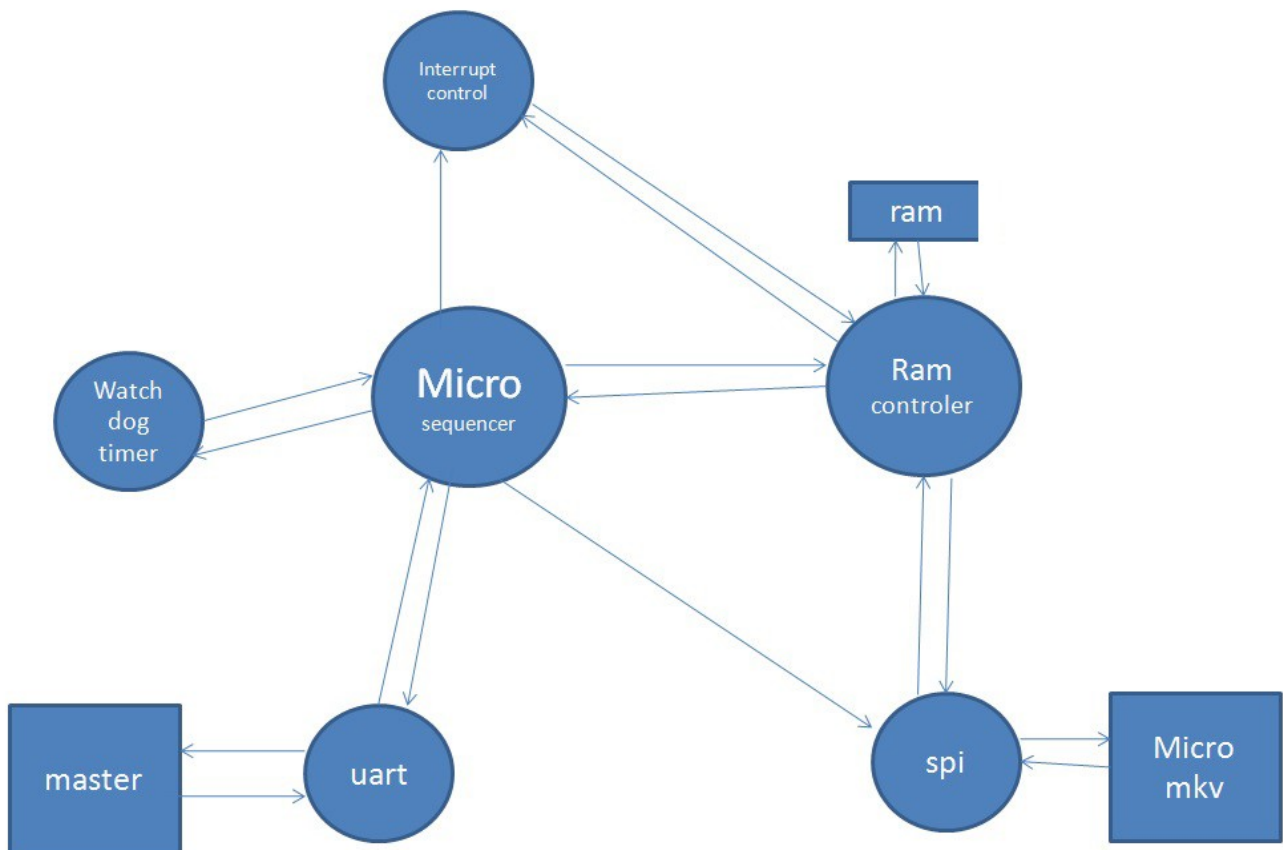
پس از دریافت درخواست سفارش ، بررسی شرایط انجام می شود تا از نظر معمولی بودن یا ویژه بودن سفارش بررسی شود. پس از مشخص شدن نوع سفارش ، activity ارسال انجام می شود که به عنوان خاتمه فرآیند مشخص می شود.



نمودار سطح ۲ جریان داده







### نمودار ساختار ترکیبی یا composite diagram:

نمودار ساختار ترکیبی مشابه نمودار کلاس و نوعی نمودار ساختاری است که به طور عمده برای مدل سازی یک سیستم از دیدگاه خرد استفاده می شود، اما بخش های مفرد را به جای کلاس های کامل به تصویر می کشد. این نوعی از نمودار استاتیک است که ساختار درونی یک کلاس و همکاری هایی که این ساختار ممکن می سازد را نمایش می دهد.

این نمودار می تواند شامل قطعات داخلی، پورت هایی باشد که از طریق آن قطعات با یکدیگر تعامل دارند یا از طریق آن نمونه هایی از کلاس با قطعات و با دنیای بیرون تعامل می کنند. ساختار ترکیبی مجموعه ای از عناصر به هم پیوسته است که در زمان اجرا برای دستیابی به اهدافی با یکدیگر همکاری می کنند. هر عنصر نقش مشخصی در همکاری دارد.

part: یک قسمت نقشی را نشان می دهد که در زمان اجرا توسط یک نمونه از یک طبقه بندی کننده یا مجموعه ای از نمونه ها بازی می شود.

Port: پورت یک نقطه تعاملی است که می تواند برای اتصال طبقه بندی کننده های ساختاریافته با قطعات آنها و با محیط استفاده شود. پورت ها می توانند به صورت اختیاری خدماتی را که ارائه می کنند و خدماتی را که از سایر بخش های سیستم نیاز دارند مشخص کنند. در نمودار، هر یک از مربع های کوچک یک پورت است.

Connector: رابط: یک رابط دو یا چند موجود را به هم متصل می کند و به آنها اجازه می دهد در زمان اجرا با هم تعامل داشته باشند. کانکتور به صورت خطی بین ترکیبی از قطعات، پورت ها و طبقه بندی کننده های ساخت یافته نشان داده می شود.

