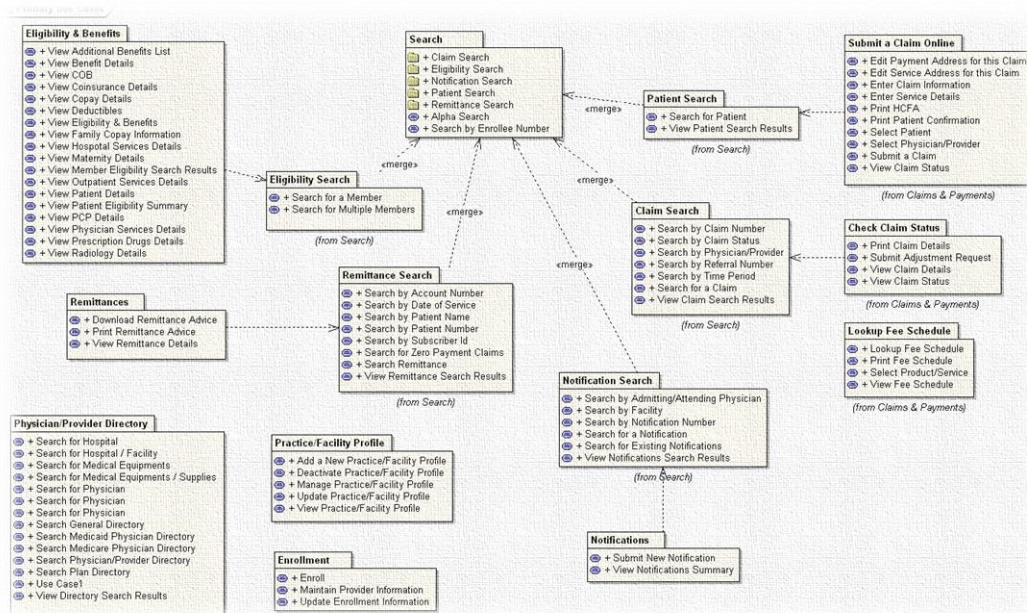


نمودار پکیج یا Package Diagram در UML



تهیه و تنظیم: پیمان مالکی



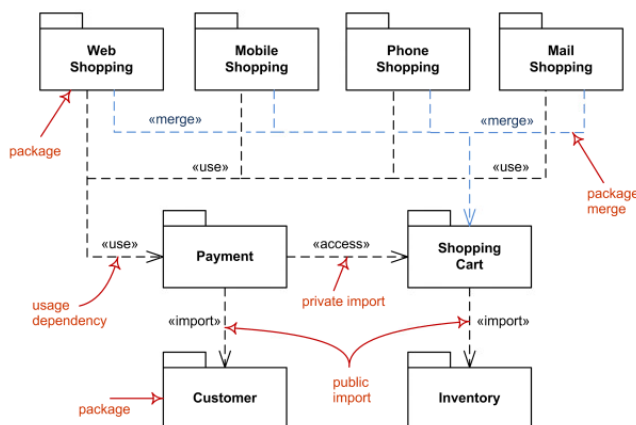
فهرست مطالب

2	نمودار پکیج
3	نمودار مدل
3	چه زمانی و به چه دلایلی تهیه این نمودار اهمیت خواهد داشت؟
5	نمادهای نمودار پکیج
5	پکیج
6	مشخصه Package URI
7	الگوی پکیج
8	عناصر قابل بسته‌بندی
11	عنصر دریافت‌شده
13	پکیج دریافت‌شده
15	پکیج ادغام‌شده
17	مدل

نمودار پکیج^۱، یک نمودار ساختاری UML است که ساختار سیستم طراحی شده را در سطح پکیج‌ها نشان می‌دهد. عناصر زیر معمولاً در نمودار پکیج ترسیم می‌شوند:

- پکیج^۲
- عنصر قابل بسته‌بندی^۳
- ارتباط وابستگی نقطه به نقطه یک طرفه^۴
- کلیشه دریافت^۵
 - عنصر دریافت شده^۶
 - پکیج دریافت شده^۷
- کلیشه کاربرد^۸
- کلیشه دسترسی^۹
- کلیشه ادغام^{۱۰}
 - پکیج ادغام شده^{۱۱}

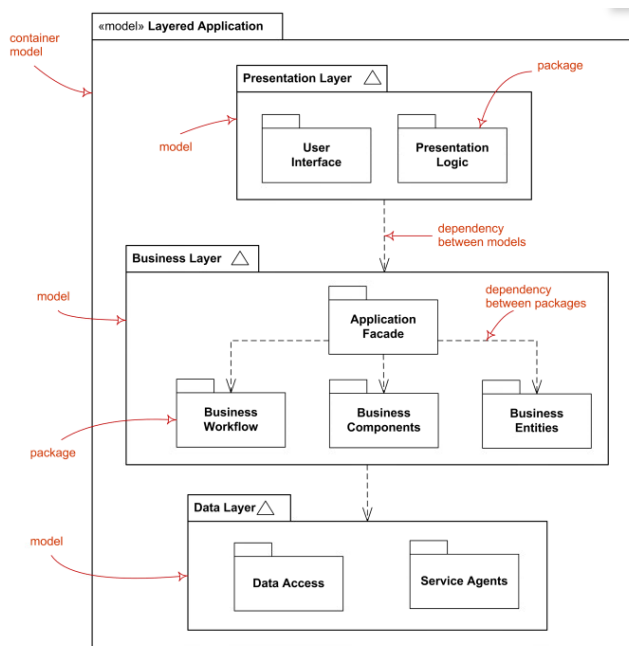
برخی از عناصر اصلی نمودار پکیج در تصویر زیر نشان داده شده است. پکیج‌های Web Shopping، Mobile Shopping، Phone Shopping و Mail Shopping از پکیج Payment استفاده می‌کنند. هر دو پکیج Payment و Shopping Cart از پکیج Customer و Inventory استفاده می‌کنند. همان 4 پکیج از پکیج Payment استفاده می‌کنند. پکیج Shopping Cart را ادغام می‌کنند. پکیج Shopping Cart و Payment از پکیج‌های دیگر را دریافت می‌کنند.



عناصر نمودار پکیج UML شامل پکیج، دریافت، دسترسی، بسته‌بندی، ادغام.

- 1 Package Diagram
- 2 Package
- 3 packageable element
- 4 dependency
- 5 «import»
- 6 element import
- 7 package import
- 8 «use»
- 9 «access»
- 10 «merge»
- 11 package merge

نمودار مدل^{۱۲}، یک نمودار ساختاری کمکی UML است که محتوای انتزاعی یا نمای خاصی از یک سیستم را برای توصیف جنبه‌های معماری، منطقی یا رفتاری سیستم نشان می‌دهد. به عنوان مثال، می‌تواند معماری یک برنامه کاربردی چند لایه (معروف به معماری چندلایه) به شکل یک مدل برنامه چندلایه، نشان داد. این نمودار از **پکیج‌ها** استفاده می‌کند یا تشکیل می‌شود. تصویر زیر برخی از عناصر اصلی **نمودار مدل** را نشان می‌دهد. برنامه لایه‌بندی شده یک مدل "container" است که شامل سه مدل دیگر به نام‌های، لایه ارائه^{۱۳}، لایه کسب و کار^{۱۴} و لایه داده^{۱۵} است. وابستگی‌هایی بین **مدل‌های** موجود، تعریف شده است.



عناصر نمودار مدل UML شامل مدل، پکیج، وابستگی نقطه‌به‌نقطه یک‌طرفه.

مدل‌ها معمولاً حاوی **پکیج‌ها** هستند. **پکیج‌ها** می‌توانند **وابستگی نقطه‌به‌نقطه یک‌طرفه** یا روابط دیگری داشته باشند، به عنوان مثال. **کلیشه دریافت**، بین آنها تعریف شده است.

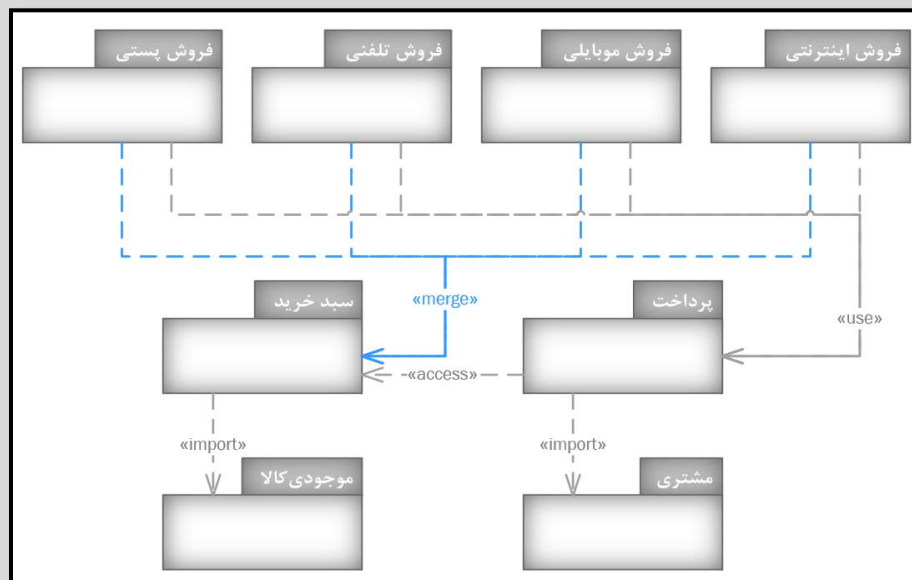
چه زمانی و به چه دلایلی تهیه این نمودار اهمیت خواهد داشت؟

نمودار پکیج، تا حدی، برای معرفی وابستگی‌های بین **پکیج‌ها**، **کلاس‌ها**، **مؤلفه‌ها** و سایر عناصر نام‌گذاری شده در سیستم شما به جهت **دریافت** و یا **دسترسی**، استفاده می‌شوند. هر وابستگی به شکل یک **وابستگی نقطه‌به‌نقطه یک‌طرفه** بین دو یا چند عنصر ارائه می‌شود.

نمودار پکیج برای نمایش ساختار عمومی عناصر سیستم استفاده می‌شود. **پکیج‌ها** برای سازماندهی سیستم‌های بزرگ که شامل نمودارها، اسناد و سایر ارائه‌های کلیدی دیگر هستند، استفاده می‌شوند. **نمودار پکیج** را می‌توان برای ساده کردن **نمودارهای کلاس** پیچیده استفاده کرد، می‌تواند **کلاس‌ها** را در **پکیج‌ها** دسته‌بندی کرد.

- Model Diagram ¹²
- Presentation Layer ¹³
- Business Layer ¹⁴
- Data Layer ¹⁵

نمودار پکیج، یک نمودار ساختاری است که ساختار عمومی و کلی سیستم را نمایش می‌دهد. این نمودار از پکیج و وابستگی نقطه به نقطه یک طرفه با کلیشه‌های مختلف مانند merge، use، access و import تشکیل شده است. در این نمودار هر پکیج می‌تواند دسته‌بندی کننده عناصر دیگری مانند مؤلفه‌ها یا کلاس‌ها و یا حتی طبقه‌بندی کننده‌ها باشند و در دسته نمودارهای معماری محسوب می‌شوند.



در تصویر بالا، سیستم فروش معرفی شده است که پکیج‌های محوری به نام‌های پرداخت و سبد خرید تشکیل شده است.

پکیج سبد خرید شامل پکیج‌های فروش اینترنتی، موبایلی، تلفنی و پستی است به عبارت دیگر سبد خرید می‌تواند شامل هر یک از این انواع فروش باشد یا از آنها تشکیل شده باشد.

پکیج پرداخت نیز برای پاسخ به درخواست پرداخت از قواعد معرفی شده در پکیج‌های فروش اینترنتی، موبایلی، تلفنی و پستی استفاده می‌کند.

پکیج پرداخت همچنین برای پاسخ به درخواست پرداخت و محاسبه رقم قابل پرداخت به صورت خصوصی به پکیج سبد خرید دسترسی دارد.

پکیج پرداخت در نهایت برای پاسخ به درخواست پرداخت و دریافت اطلاعات مشتری متقاضی، این اطلاعات را از پکیج مشتری دریافت می‌نماید.

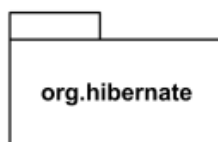
پکیج سبد خرید برای کنترل موجودی کالا در انبار که به عنوان یک قاعده در امکانات خود در نظر گرفته است، این اطلاعات را از پکیج موجودی کالا دریافت می‌نماید.

نمودار پکیج، یک نمودار ساختاری UML است که ساختار سیستم طراحی شده را در سطح پکیجها نشان می‌دهد. عناصر زیر معمولاً در نمودار پکیج ترسیم می‌شوند:

- پکیج
- عنصر قابل بسته‌بندی
- وابستگی نقطه به نقطه یک طرفه
- عنصر دریافت شده
- پکیج دریافت شده
- پکیج ادغام شده

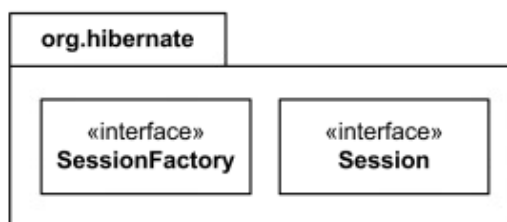
پکیج

پکیج یک فضای نام است که برای گروه‌بندی عناصری که از نظر معنایی مرتبط هستند و ممکن است با هم تغییر کنند، استفاده می‌شود. این یک مکانیسم مبتنی بر هدف است که برای سازماندهی عناصر به گروه‌ها برای ارائه ساختار بهتر برای مدل سیستم استفاده می‌شود. عناصر متعلق به یک پکیج باید همه باهم بسته‌بندی شوند. اگر پکیجی از یک مدل حذف شود، همه عناصر متعلق به پکیج مورد نظر نیز حذف خواهند شد. پکیج به خودی خود عنصر قابل بسته‌بندی است، بنابراین هر پکیج می‌تواند عضوی از پکیج‌های دیگر نیز باشد. از آنجایی که پکیج یک فضای نام است، عناصر مرتبط یا هم نوع که دارای نام‌های منحصر به فرد هستند باید در یک پکیج محصور شوند. انواع مختلفی از عناصر مجاز به داشتن نام یکسان هستند. به عنوان یک فضای نام، یک پکیج می‌تواند عناصر جداگانه پکیج‌های دیگر یا همه اعضای پکیج‌های دیگر را دریافت کنند. پکیج همچنین می‌تواند با پکیج‌های دیگر ادغام شود. یک پکیج به شکل یک پوشه زبانه‌دار (یک مستطیل با یک زبانه کوچک) ارائه می‌شود. اگر عناصر پکیج در داخل مستطیل پکیج نشان داده نشوند، نام پکیج باید داخل آن قرار گیرد.



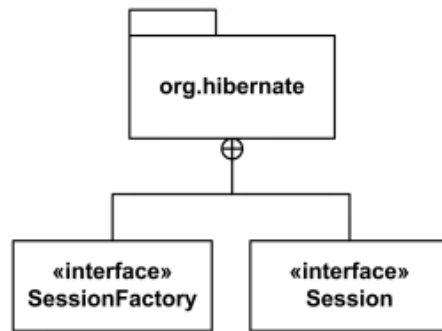
پکیج org.hibernate

عناصر پکیج ممکن است در محدوده پکیج نشان داده شوند. در این صورت نام پکیج باید در برگه قرار گیرد.



پکیج org.hibernate شامل Session و SessionFactory است

نموداری که یک پکیج با محتوا را نشان می‌دهد، مجاز است فقط زیرمجموعه‌ای از عناصر موجود را با توجه به معیارهایی نشان دهد. عناصر پکیج ممکن است خارج از پکیج با انشعاب خطوط از پکیج به عناصر نشان داده شوند. یک علامت مثبت (+) در یک دایره در انتهای متصل به فضای نام (پکیج) رسم می‌شود. این نماد برای پکیج‌ها از نظر معنایی معادل حالتی است که عناصر ترکیب کننده پکیج را نمایش می‌دهد (نمایش عناصر تودرتو).

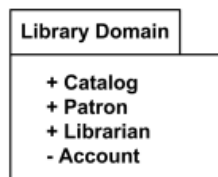


پکیج org.hibernate شامل واسطه‌های SessionFactory و Session است

عنصری که می‌توان در یک پکیج با استفاده از نام‌های غیر مشروط به آنها اشاره کرد عبارتند از:

- عناصر متعلق به پکیج
- عناصر دریافتی از پکیج دیگر
- عناصر موجود در فضاهای نام (خارجی)

عناصر متعلق به پکیج و دریافت شده از پکیج دیگر، ممکن است دارای قابلیت مشاهده باشند که تعیین می‌کند آیا آنها خارج از پکیج در دسترس هستند یا خیر. اگر عنصری که متعلق به یک پکیج است دارای قابلیت مشاهده باشد، می‌تواند فقط قابلیت نمایش عمومی (public) یا خصوصی (private) داشته باشد. قابلیت نمایش محافظت شده (protected) یا پکیج (package) مجاز نیست. مقدار مشخصه قابلیت نمایش یک عنصر پکیج ممکن است با قرار دادن نماد قابل مشاهده ("+" برای عمومی و "-" برای خصوصی) قبل از نام عنصر، نشان داده شود.



همه عناصر پکیج Library Domain عمومی هستند به جز Account

عناصر عمومی یک پکیج همیشه در خارج از پکیج با استفاده از نام‌های واجد شرایط قابل دسترسی هستند.

مشخصه Package URI

پکیج دارای مشخصه URI اختیاری است که به عنوان شناسه منحصر به فرد پکیج عمل می‌کند. این مشخصه در UML 2.4 بیشتر برای پشتیبانی از تبادل پروفایل‌ها با استفاده از XMI معرفی شد. توجه داشته باشید که مشخصات UML 2.4 به این مشخصه URI نیاز دارد که از قواعد و نگارش مشخصات IETF URI RFC 2396 پیروی کند در حالی که نسخه جدیدتر نگارش URI RFC 3986 (منتشر شده در سال 2005) RFC 2396 را منسوخ کرده است. هر دو RFC، به URI ها اجازه می‌دهند مکان‌یاب منبع (URL)، نام‌منبع یکنواخت (URN) یا هر دو باشند. اگرچه URN در واقع هویت مورد را تعریف می‌کند (شناسه منحصر به فرد) و URL روشی را برای یافتن آن ارائه می‌دهد، دیدن URL هایی که به عنوان URN خدمت می‌کنند بسیار رایج و گاهی گیج کننده است. در اینجا چند نمونه از مشخصات Package URI در قالب URL آورده شده است. توجه داشته باشید که استفاده از طرح "http://" اغلب به پروتکل HTTP مربوط نمی‌شود و هیچ منبع واقعی در آن مکان قرار ندارد. URL فقط به عنوان جایگزینی برای شناسه منحصر به فرد استفاده می‌شود:

<http://www.ietf.org/rfc/rfc2396.txt>

<http://www.omg.org/spec/UML/20100901/PrimitiveTypes.xmi>

<http://www.uml-diagrams.org/profiles/20110410/EJB-30.xmi>

ترجیح شخصی من استفاده از URN ها به جای آن است، زیرا URN به طور خاص برای تعریف هویت (شناسه منحصر به فرد) طراحی شده است و به معنای در دسترس بودن منبع شناسایی شده (مثلاً برای منابع منسوخ) نیست:

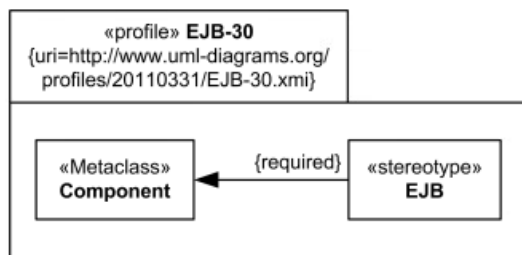
urn:ietf:rfc:3986

urn:oasis:names:specification:docbook:dtd:xml:4.1.2

urn:schemas-microsoft-com:xml-data

urn:uml-diagrams-org:profiles:20110410:EJB-30

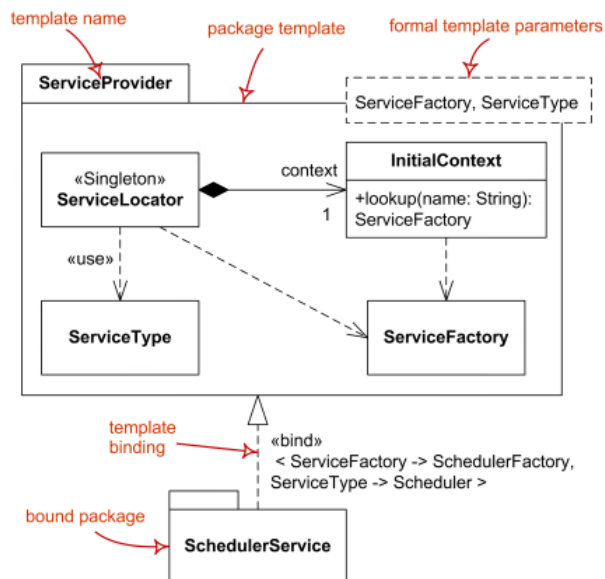
مشخصه URI یک پکیج ممکن است به شکل {uri=<uri>} بعد از نام پکیج ارائه شود.



پرو فایل EJB به عنوان یک پکیج با مشخصه URI نشان داده شده است

الگوی پکیج

پکیج می تواند به عنوان یک الگو برای پکیج های دیگر استفاده شود. توجه داشته باشید که [مشخصات UML 2.4.1] به طور متناقض آن را هم الگوی پکیج و هم پکیج الگو می نامد. عنصر قابل بسته بندی را می توان به عنوان پارامتر الگو، استفاده کرد. یک پارامتر الگوی پکیج ممکن است به هر عنصری که متعلق به الگوی پکیج یا استفاده شده در پکیج، یا الگوهای تودرتو در آن اشاره داشته باشد. یک پکیج ممکن است به یک یا چند الگوی پکیج متصل شود. هنگامی که چندین اتصال اعمال می شود، نتیجه اتصالات با گرفتن نتایج میانی و ادغام آنها در نتیجه ترکیبی با استفاده از ادغام پکیج، ایجاد می شود.



الگوی پکیج ServiceProvider و پکیج SchedulerService باندشده به آن

عناصر قابل بسته‌بندی یک عنصر نامگذاری شده است که ممکن است مستقیماً متعلق به یک پکیج باشد. چند نمونه از عناصر قابل بسته‌بندی عبارتند از:

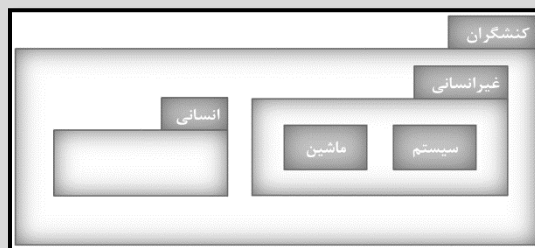
- نوع^{۱۶}
- طبقه‌بندی‌کننده^{۱۷} (← نوع)
- کلاس^{۱۸} (← طبقه‌بندی‌کننده)
- مورد کاربرد^{۱۹} (← طبقه‌بندی‌کننده)
- مؤلفه^{۲۰} (← طبقه‌بندی‌کننده)
- پکیج^{۲۱}
- محدودیت^{۲۲}
- وابستگی نقطه‌به‌نقطه یک‌طرفه^{۲۳}
- رویداد^{۲۴}

خلاصه و چکیده - Package Diagram Notation

عناصر یک نمودار پکیج، شامل پکیج، عنصر قابل بسته‌بندی و ارتباط وابستگی نقطه‌به‌نقطه یک‌طرفه می‌باشد. با توجه به اینکه، نمودار پکیج در حال نمایش چه حوزه‌ای از محدوده در حال تحلیل را نمایش می‌دهد؛ عناصر مختلفی می‌تواند در یک نمودار پکیج قرار گیرد. به عبارت دیگر ممکن است که نمودار مورد نظر فقط بخواهد یک پکیج را تفسیر کند در این صورت نمودار مورد نظر شامل پکیج، عناصر قابل بسته‌بندی و ارتباط وابستگی نقطه‌به‌نقطه یک‌طرفه خواهد بود. ولیکن اگر نمودار مورد نظر بخواهد ارتباط درونی یک سیستم را تفسیر نماید ممکن است شامل پکیج و ارتباط وابستگی نقطه‌به‌نقطه یک‌طرفه باشد.

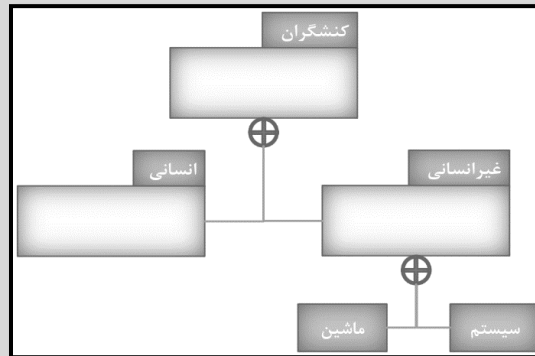
دقت داشته باشید که در هنگام استفاده از ارتباط وابستگی نقطه‌به‌نقطه یک‌طرفه می‌توانید از کلیشه‌های از پیش تعریف‌شده و یا سفارشی خود نیز استفاده کنید. کلیشه‌های پر کاربرد شامل «import»، «use»، «merge» و «access» می‌باشند.

پکیج، در واقع یک فضای نام است و می‌تواند شامل قطعات یا همان عناصر قابل بسته‌بندی باشد. این قطعات می‌توانند محدوده وسیعی از عناصر باشند زیرا بنا به تعریف، هر قطعه، نماینده‌ای از یک طبقه‌بندی‌کننده است و طبقه‌بندی‌کننده هر عنصر UML می‌تواند باشد. در این خصوص این قطعات یا همان عناصر قابل بسته‌بندی شامل: Type، Classifier، Class، Use Case، Component، Package، Constraint، Dependency و Event هستند.

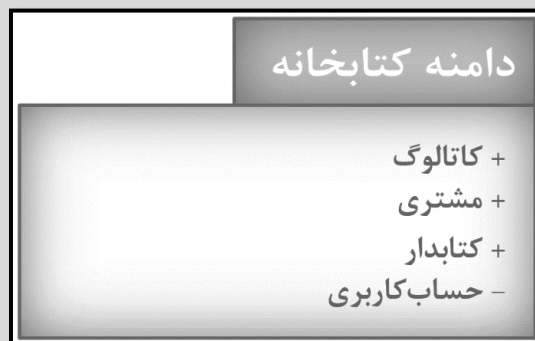


- Type¹⁶
- Classifier¹⁷
- Class¹⁸
- Use Case¹⁹
- Component²⁰
- Package²¹
- Constraint²²
- Dependency²³
- Event²⁴

در تصویر بالا، پکیج کنشگران شامل پکیج‌های انسانی و غیرانسانی است. پکیج غیرانسانی شامل عناصر قابل‌بسته‌بندی ماشین و سیستم است. البته این نمودار را به شکل زیر نیز می‌تواند نمایش داده شود.



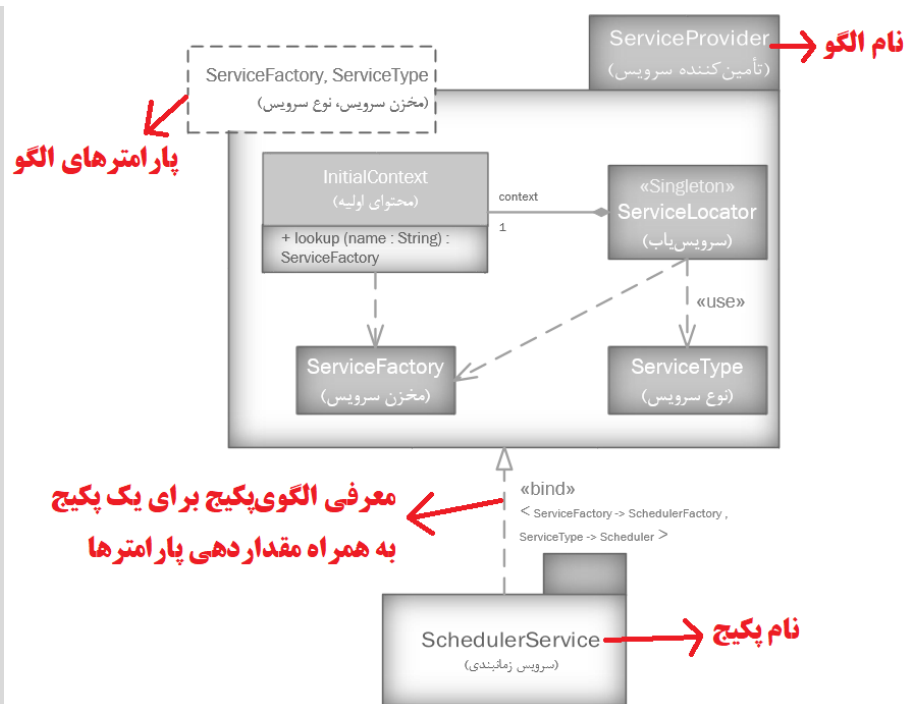
عناصر داخل یک پکیج می‌توانند مشخصه قابلیت‌نمایش داشته باشند که البته فقط می‌تواند یکی از مقادیر عمومی (public) یا خصوصی (private) داشته باشد.



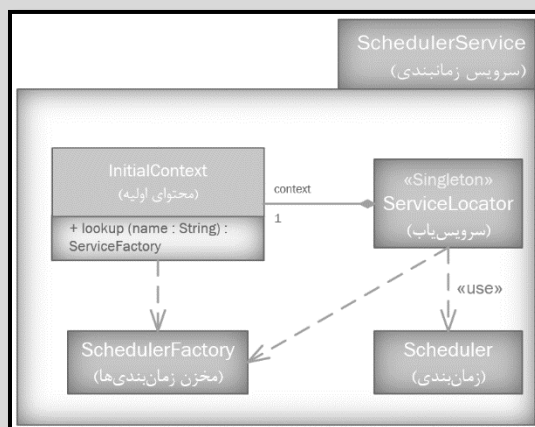
در تصویر بالا، عناصر قابل بسته‌بندی کاتالوگ، مشتری و کتابدار، دارای مقدار public برای مشخصه قابلیت‌مشاهده هستند و در نتیجه از خارج از این پکیج قابل دسترسی هستند ولیکن عنصر حساب کاربری دارای مقدار private است که باعث می‌شود فقط عناصر داخل این پکیج به آن دسترسی داشته باشند.

عناصر داخل یک پکیج بجز مشخصه قابلیت‌نمایش می‌توانند مشخصه URI هم که یک مشخصه اختیاری است داشته باشند. این مشخصه مانند شناسه برای یک پکیج عمل می‌کند. مقدار این مشخصه می‌تواند از دو قالب URL و یا URN پیروی کند. البته قالب URN بهتر است زیرا در اکثر مواقع این پکیج‌ها هیچ منبع قابل آدرس‌دهی ندارند که بتوان با یک URL به آن دسترسی پیدا کرد. مشخصه URI یک پکیج می‌تواند به شکل {uri=<uri>} بعد از نام پکیج ارائه شود.

مانند اکثر عناصر و نمادهای UML، پکیج نیز الگویپذیر است. الگوی پکیج، دارای دو بخش ثابت و پارامتریک است. پارامترهای الگوی پکیج همان عناصر قابل بسته‌بندی هستند که می‌توان آنها را قطعه نامید. این پارامترها می‌توانند نماینده عناصر متعلق به پکیج شوند و یا حتی ممکن است الگوهای تودرتو داشته باشیم. برای استفاده از یک الگوی پکیج در یک پکیج می‌توان پکیج مورد نظر را بدون شرحی بر محتوای آن فقط به یک یا چند الگوی پکیج باند کرد و در ضمن باند کردن، اطلاعات پارامترهای آن را تکمیل کرد.



در تصویر بالا، پکیجی به نام **SchedulerService** معرفی شده است که از الگویی با نام **ServiceProvider** برای معرفی جزئیات خود استفاده می کند. الگوی مورد نظر دارای دو پارامتر **ServiceFactory** و **ServiceType** است که در تصویر کامل الگوی پکیج **ServiceProvider** به صورت یک عنصر قابل بسته بندی معرفی شده اند. اتصال دهنده بین پکیج **SchedulerService** و الگوی پکیج **ServiceProvider** یک ارتباط وابستگی نقطه به نقطه یک طرفه تحقق است که نشان می دهد پکیج مورد نظر، الگوی پکیج مورد نظر را محقق می نماید. عمل تحقق با استفاده از مقداردهی پارامترهای الگو صورت می گیرد. در این تخصیص مقادیر، مقدار **SchedulerFactory** یا به عبارت دیگر مخزن زمان بندی ها جایگزین **ServiceFactory** شده و مقدار **Scheduler** یا به عبارت دیگر زمان بندی جایگزین **ServiceType** قرار می گیرد. تصویر زیر نتیجه این تحقق خواهد بود.



عنصر دریافت شده یک رابطه وابستگی نقطه‌نقطه یک‌طرفه بین فضای نام دریافت کننده و عنصر قابل بسته‌بندی دریافتی است. این رابطه، اجازه می‌دهد تا عنصر مورد نظر نام غیرمشروط^{۲۵} و فقط با استفاده از نام خود ارجاع داده شود. عنصر دریافت شده برای دریافت گزینشی عناصر منفرد بدون اینکه نیاز به دریافت پکیج دربرگیرنده آن وجود داشته باشد، استفاده می‌شود.

نام عنصر قابل بسته‌بندی یا نام مستعار آن باید به نام فضای نام ارائه کننده اضافه شود. دریافت به صورت اشاره به مرجع کار می‌کند، به عبارت دیگر امکان اضافه کردن مشخصه به عنصر دریافت شده وجود ندارد، اما می‌توان عنصر ارجاع شده را در فضای نامی که از آن دریافت شده است، تغییر داد.

می‌توان کنترل کرد که عنصر دریافت شده مورد نظر، توسط فضاهای نام دیگر با استفاده از عنصر یا پکیج دریافت شده، دریافت شود یا نشود. قابلیت نمایش عنصر دریافت شده موردنظر ممکن است مشابه یا محدودتر از عنصر ارائه شده باشد. قابل نمایش بودن دریافت یک عنصر یا عمومی (public) یا خصوصی (private) است.

قابلیت نمایش پیش فرض عنصر دریافت شده مانند عنصر ارائه شده است. اگر عنصر ارائه شده دارای مشخصه قابلیت نمایش نباشد، می‌توان قابلیت نمایش را به عنصر دریافت شده، اضافه کرد. مقدار پیش فرض در این حالت عمومی (public) است.

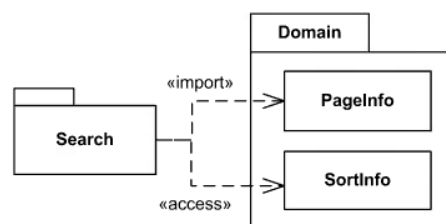
در صورتیکه نام عنصر دریافت شده در فضای نام دریافت کننده با نام بیرونی عنصر (عنصری که در یک فضای نام محصور شده با استفاده از نام غیرمشروط فضای نام محصور شده در دسترس است) تفاوت داشته باشد، نام بیرونی عنصر دریافت شده، نادیده گرفته می‌شود و برای اشاره به آن از نام مشروط^{۲۶} که به آن اشاره دارد استفاده می‌کنند. به نام بیرونی عنصر دریافت شده با استفاده از نام مشروط آن دسترسی خواهیم داشت.

اگر بیش از یک عنصر با همان نام در نتیجه دریافت عنصر یا دریافت پکیج به یک فضای نام وارد شود، عناصر به فضای نام دریافت کننده اضافه نمی‌شوند و نام آن عناصر باید نام مشروط باشد تا در آن استفاده شود. اگر در یک فضای نام، نام یک عنصر دریافت شده با نام یک عنصر متعلق به فضای نام مورد نظر، یکی باشد، آن عنصر به فضای نام دریافت کننده، اضافه نمی‌شود و نام آن عنصر برای استفاده باید یک نام مشروط باشد.

نام مستعار^{۲۷} نامی است که به جای نام عنصر قابل بسته‌بندی دریافت شده، باید به فضای نام پکیج دریافت کننده، اضافه شود. نام مستعار نباید با هیچ نام عنصر دیگری در پکیج دریافت کننده، تداخل داشته باشد. به طور پیش فرض، هیچ نام مستعاری استفاده نمی‌شود.

دریافت عنصر با استفاده از یک فلش خط چین دار با سر پیکان باز از فضای نام دریافت کننده به عنصر ارائه شده نشان داده می‌شود. توجه داشته باشید که اگرچه دقیقاً شبیه روابط وابستگی نقطه‌به‌نقطه یک‌طرفه کاربرد است، اما یک رابطه مستقیم و کاملاً مجزا است.

در حالی که کلمه کلیدی «access» برای نشان دادن قابلیت نمایش دریافت به صورت خصوصی (private) استفاده می‌شود، اگر قابلیت نمایش دریافت، عمومی (public) باشد، کلمه کلیدی «import» در نزدیکی فلش خط چین دار، نشان داده می‌شود. اگر عنصر دریافت شده یک پکیج نباشد، ممکن است به صورت اختیاری استفاده از کلمه کلیدی «element import» اولویت داشته باشد.



دریافت عمومی عنصر PageInfo و دریافت خصوصی عنصر SortInfo از بسته Domain

²⁵ unqualified name

²⁶ qualified name

²⁷ Alias

عناصری که برای استفاده در **پکیج‌های** دریافت‌کننده از طریق دریافت عنصر در دسترس قرار می‌گیرند، ممکن است رنگ مشخصی داشته باشند یا برای نشان دادن اینکه نمی‌توان آنها را تغییر داد، کم‌رنگ شوند. نام مستعار ممکن است بعد یا زیر کلمه کلیدی «import» نشان داده شود.

به‌عنوان جایگزینی برای فلش خط‌چین‌دار، نشان دادن **عنصر دریافت‌شده** با داشتن متنی که **عنصر دریافت‌شده** را به‌طور منحصر‌به‌فرد در داخل آکولادها در زیر یا بعد از نام **فضای نام** مشخص می‌کند، مجاز است. نگارش این مورد را می‌توان مانند زیر توصیف کرد (توجه داشته باشید، این نسخه اصلاح شده من از این نگارش است):

element-import ::= '{' ('element import' | 'element access') qualified-name ['as' alias] '}'

خلاصه و چکیده - Element Import

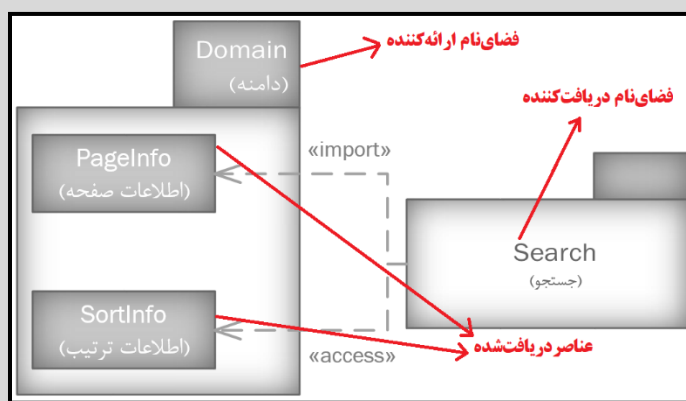
در طراحی سیستم، ممکن است که یک **پکیج**، **عناصر قابل‌بسته‌بندی** خود را از **عناصر دریافت‌شده** (عناصری که قابلیت دریافت شدن دارند) تهیه کند. در این حالت هم می‌توانید از نام **غیرمشروط** و یا **مستقیم** **عنصر دریافت‌شده** در محل **پکیج** منبع استفاده کنید و هم می‌توانید به صورت مرجعی به **عنصر دریافت‌شده** در محل **پکیج** (**فضای نام**) منبع، نام **مشرط** که شامل نام **عنصر دریافت‌شده** به همراه نام **فضای نام** منبع است استفاده کنید. روش اخیر برای حالتی که نام **عنصر دریافت‌شده** با نام عنصری در داخل **پکیج** دریافت‌کننده عنصر، برابر باشد مفید است در این حالت، اگر از نام **مشرط** استفاده نکنید، امکان دریافت عنصر را از دست می‌دهید.

عنصر دریافت‌شده، یک ارجاع به عنصر قابل دریافت در **فضای نام** مرجع است. به عبارت دیگر یک پیوند کامل بین عنصر ارائه شده در **فضای نام** مرجع و **عنصر دریافت‌شده** در **فضای نام** دریافت‌کننده وجود دارد به گونه‌ای که حذف این عنصر در **فضای نام** مرجع باعث حذف آن در **فضای نام** دریافت‌کننده می‌شود و حتی هیچ **مشخصه‌ای** از **عنصر دریافت‌شده** را در **فضای نام** دریافت‌کننده نمی‌توان تغییر داد.

اگر بیش از یک **عنصر دریافت‌کننده** داشته باشید که همه دارای یک نام باشند، بجز روش قبلی می‌توانید از نام **مستعار** نیز استفاده کنید. توجه داشته باشید که می‌توانید برای یکی یا هر دو آنها نام **مستعار** استفاده کنید و همواره توجه داشته باشید که این **نامهای مستعار** باید منحصر به فرد باشند.

مانند تمام عناصر UML قابل طراحی، **عناصر دریافت‌شده** نیز دارای **مشخصه قابلیت‌نمایش** هستند. این **مشخصه** می‌تواند مقدار **public** و یا **private** داشته باشد. این **مشخصه** در یک **عنصر دریافت‌شده** از **فضای نام** منبع به **فضای نام** دریافت‌کننده به صورت منطقی، انتقال می‌یابد. مقدار پیش‌فرض این **مشخصه**، **public** است.

در یک **نمودار پکیج** اگر از امکان **عناصر دریافت‌شده** استفاده کرده باشید باید این امکان را با استفاده از ارتباط **وابستگی نقطه‌به‌نقطه یک‌طرفه کاربرد** نشان دهید. این نوع ارتباط با توجه به **مشخصه قابلیت‌نمایش** **عنصر دریافت‌شده** می‌تواند از دو **کلیشه** «access» برای مقدار **private** این **مشخصه** و «import» برای مقدار **public** این **مشخصه**، استفاده نماید.

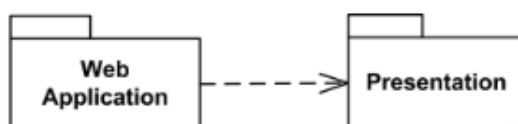


در تصویر بالا، **عنصر دریافت‌شده** **PageInfo** با **قابلیت‌نمایش** **public** ارائه شده است که دارای **کلیشه** «import» است و عنصر **SortInfo** با **قابلیت‌نمایش** **private** که دارای **کلیشه** «access» است.

پکیج دریافت شده یک رابطه مستقیم بین یک فضای نام دریافت کننده و پکیج ارائه شده است که به استفاده از نام های غیرمشتروط برای ارجاع به عناصر پکیج از فضای نام دیگر اجازه می دهد.

فضای نام دریافت کننده، نام عناصر پکیج دریافت شده را به فضای نام خود اضافه می کند. از نظر مفهومی، یک پکیج دریافت شده معادل یک به یک عناصر دریافت شده مجزای فضای نام ارائه کننده است، مگر اینکه قبلاً یک عنصر دریافت شده، جداگانه تعریف شده باشد.

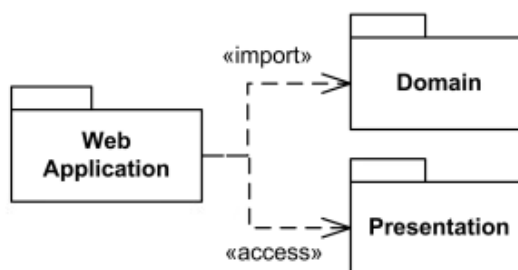
پکیج دریافت شده با استفاده از یک فلش خط چین دار با سر پیکان باز از فضای نام ارائه کننده به پکیج دریافت کننده نشان داده می شود.



پکیج Presentation توسط پکیج WebApplication دریافت شده است

توجه داشته باشید که اگرچه دقیقاً شبیه روابط وابستگی نقطه به نقطه یک طرفه کاربرد است، اما یک رابطه مستقیم و کاملاً مجزا است. قابلیت نمایش پکیج دریافت شده می تواند عمومی (public) یا خصوصی (private) باشد. اگر پکیج دریافت شده عمومی باشد، عناصر دریافت شده پکیج دریافت شده به فضای نام اضافه می شوند و در خارج از فضای نام نیز قابل مشاهده هستند، در حالی که اگر خصوصی باشد، همچنان به فضای نام اضافه می شوند اما در خارج، قابل مشاهده نیستند.

یک کلمه کلیدی در نزدیکی فلش خط چین دار نشان می دهد کدام نوع پکیج دریافت شده مورد نظر است. کلمات کلیدی از پیش تعریف شده عبارتند از «import» برای پکیج دریافت شده عمومی و «access» برای پکیج دریافت شده خصوصی. به طور پیش فرض، مقدار قابلیت مشاهده، عمومی (public) است.



دریافت خصوصی پکیج Presentation و عمومی پکیج Domain توسط پکیج WebApplication

به عنوان جایگزینی برای پیکان خط چین دار، می توان با متنی که پکیج دریافت شده را در زیر یا بعد از نام فضای نام به طور منحصربه فرد شناسایی می کند، یک پکیج دریافت شده را نشان داد. نگارش در این مورد را می توان اینگونه توصیف کرد (توجه داشته باشید، این نسخه اصلاح شده من از این نگارش است):

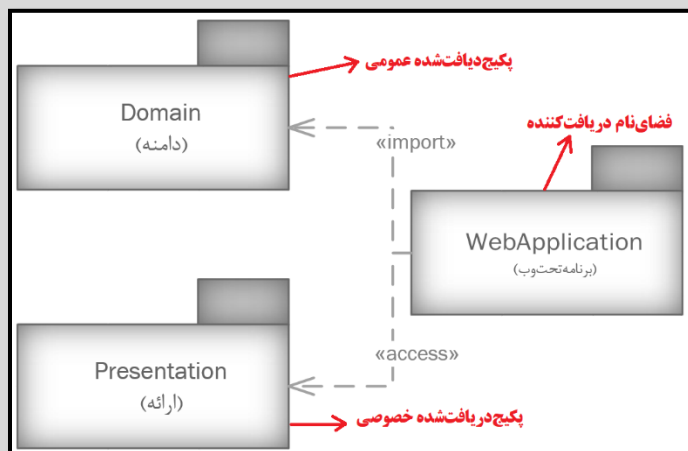
```
package-import ::= '{ ( 'import' | 'access' ) qualified-name '}'
```

عناصری که برای استفاده در پکیج های دریافت شده از طریق دریافت پکیج در دسترس قرار می گیرند، ممکن است رنگ مشخصی داشته باشند یا کم رنگ شوند تا نشان دهند که نمی توان آنها را تغییر داد.

خلاصه و چکیده - Package Import

مانند عناصر دریافت شده، می توانید پکیج ها را نیز از فضای نام مبدأ به صورت کامل در فضای نام مقصد، دریافت کنید که در این حالت فضای نام مقصد اسامی تمامی عناصر درون پکیج دریافت شده را به عنوان عناصر متعلق به خود، به خود اضافه می کند. نام پکیج دریافت شده در فضای نام مقصد به صورت غیرمشتروط خواهد بود. ارتباط بین پکیج دریافت شده و فضای نام مقصد به صورت یک ارتباط وابستگی نقطه به نقطه یک طرفه کاربرد خواهد بود.

مانند عناصر دریافت‌شده، پکیج‌های دریافت‌شده نیز می‌توانند مشخصه قابلیت‌نمایش داشته باشند. مقدار این مشخصه می‌تواند عمومی (public) یا خصوصی (private) باشد. مقدار این مشخصه در نمای ارتباط بین فضای نام دریافت‌کننده و پکیج دریافت‌شده مشخص می‌شود. اگر مقدار مشخصه قابلیت‌نمایش را عمومی (public) اعلام کنید؛ ارتباط مورد نظر با کلیشه «import» همراه خواهد بود و به این معنی است که عناصر درون پکیج دریافت‌شده در فضای نام مقصد وجود خواهند داشت و این عناصر برای خارج از این فضای نام نیز در دسترس خواهند بود. ولیکن اگر مقدار مشخصه قابلیت‌نمایش را خصوصی (private) اعلام کنید؛ ارتباط مورد نظر با کلیشه «access» همراه خواهد بود و به این معنی است که عناصر درون پکیج دریافت‌شده در فضای نام مقصد وجود خواهند داشت ولی این عناصر برای خارج از این فضای نام در دسترس نخواهند بود.



پکیج ادغام شده یک رابطه مستقیم بین دو پکیج است که نشان می دهد محتوای یک پکیج توسط محتویات پکیج دیگر گسترش می یابد.

ادغام پکیج شبیه به رابطه وابستگی نقطه به نقطه جهت دار تولید کننده یا خاص کننده است به این معنا که عنصر منبع به طور مفهومی مشخصات عنصر هدف را به مشخصات خود اضافه می کند و در نتیجه عنصری ایجاد می شود که مشخصات هر دو را ترکیب می کند.

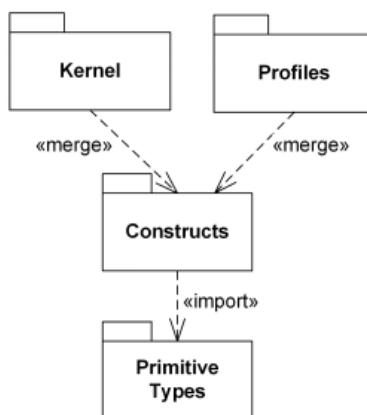
ادغام پکیج را می توان به عنوان عملیاتی در نظر گرفت که محتویات دو پکیج را می گیرد و پکیج جدیدی تولید می کند که محتویات پکیج های درگیر در ادغام را ترکیب می کند.

این مکانیسم زمانی باید مورد استفاده قرار گیرد که عناصر تعریف شده در پکیج های مختلف، نام یکسانی داشته باشند و چنین در نظر گرفته شده است که یک مفهوم را نشان دهند. اغلب از آن برای ارائه تعاریف مختلف از یک مفهوم معین برای اهداف مختلف استفاده می شود که از یک تعریف پایه مشترک شروع می شود.

یک مفهوم پایه معین به صورت افزایشی بسط می یابد و هر افزایش در یک پکیج ادغام شده جداگانه تعریف می شود. با انتخاب اینکه کدام افزایش ها باید ادغام شوند، می توان یک تعریف سفارشی از یک مفهوم برای یک هدف خاص به دست آورد.

ادغام پکیج ها به ویژه در مدل سازی مفید است و به طور گسترده در تعریف متامدل²⁸ UML استفاده می شود.

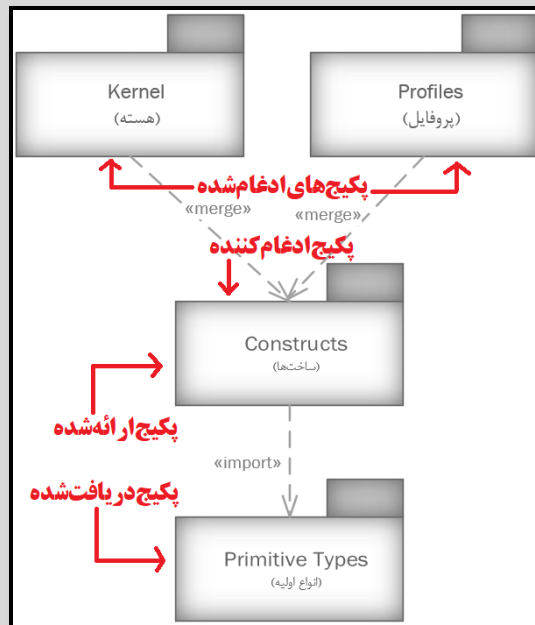
ادغام پکیج با استفاده از یک خط چین دار با یک پیکان باز که از پکیج ارائه کننده به پکیج ادغام شده اشاره دارد نشان داده می شود. کلمه کلیدی «merge» در نزدیکی خط چین دار نشان داده شده است.



پکیج های Kernel و Profiles برای پکیج Constructs ادغام می شوند که پکیج Primitive Types آن را دریافت می کند

خلاصه و چکیده - Package Merge

اگر در ترکیب یک پکیج هیچ عنصر ذاتی وجود نداشته باشد و تمام عناصر آن از ترکیب عناصر پکیج های دیگر تشکیل شده باشند این حالت را ادغام پکیج ها می گویند و پکیج هایی که به عنوان ارائه کننده عناصر در این ارتباط شرکت دارند، پکیج ادغام شده می نامند. این ادغام می تواند شامل چند پکیج ادغام شده و یک پکیج ادغام کننده باشد که این را می توان به یک رابطه وابستگی نقطه به نقطه جهت دار تولید کننده تشبیه کرد که البته با کلیشه «merge» همراه خواهد بود. ولی اگر این ادغام شامل یک پکیج ادغام شده و یک پکیج ادغام کننده باشد، این را می توان به یک رابطه وابستگی نقطه به نقطه جهت دار خاص کننده نسبت داد که باز هم با کلیشه «merge» همراه خواهد بود.



مدل، یک پکیج تخصصی UML است که یک سیستم را از نقطه نظر و دیدگاه خاصی توصیف می‌کند. این دیدگاه ممکن است به تعریف پروفایل نیز اشاره داشته باشد. از **مدل‌ها** برای ترسیم **نمودارهای مدل**، استفاده می‌شود.

[UML 2.5 FTF - Beta 1] برای تعریف محدوده **مدل**، یک سیستم، چندان سازگار نیست. در یک جا می‌گویند که **یک مدل** نمایی از یک سیستم فیزیکی را به تصویر می‌کشد، در حالی که در جای دیگر می‌گویند که این سیستم به معنای وسیع‌تر است و ممکن است نه تنها نرم‌افزار و سخت‌افزار را شامل شود، بلکه سازمان‌ها و فرآیندها را نیز شامل می‌شود.

نقطه‌نظر یا دیدگاه به عنوان انتزاعی از سیستم معرفی می‌شود که فقط جنبه‌هایی از سیستم را شامل می‌شود که با هدف **مدل** مرتبط هستند، آن جنبه‌ها را برای دسته خاصی از ذینفعان یا از دیدگاه خاصی از ذینفعان و در شرایط مناسب و با سطحی از جزئیات قابل قبول برای آنها، توصیف می‌کنند.

این بدان معناست که **مدل‌های** مختلفی را می‌توان برای یک سیستم ارائه کرد که هر **مدل**، سیستم را از دیدگاه متفاوت یا در سطح انتزاعی متفاوت نشان می‌دهد.

UML 2.5 ذینفعان را با بیان عبارت "به عنوان مثال" به طور آزادانه تعریف می‌کند، به عنوان مثال، طراحان، کاربران یا مشتریان سیستم.

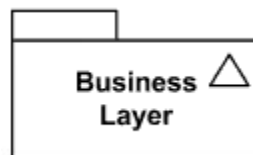
ذینفع (در زمینه پروژه‌ها یا سیستم‌های نرم‌افزاری) می‌تواند به عنوان یک فرد، گروه یا سازمان در داخل یا خارج از پروژه یا سیستم تعریف شود که حامی مالی یا علاقه‌مند به پروژه یا تأثیرگذار مثبت یا منفی در پروژه است.

نمونه‌هایی از ذینفعان عبارتند از مشتریان، گروه‌های کاربری، مدیریت سطح بالا، صاحبان محصول، مدیران پروژه، تیم‌های توسعه، مهندسين کیفیت نرم‌افزار و غیره.

مشخصات UML 2.5 نیز بین اهداف آن و مثال‌هایی که **مدل‌ها** را توضیح می‌دهند ناسازگار است. در حالی که بیان می‌کند که یک **مدل** کامل است به این معنا که کل سیستم را پوشش می‌دهد، یک مثال ارائه شده **مدل‌هایی** را نشان می‌دهد که بخش‌ها (سطوح) یک سیستم را نشان می‌دهند.

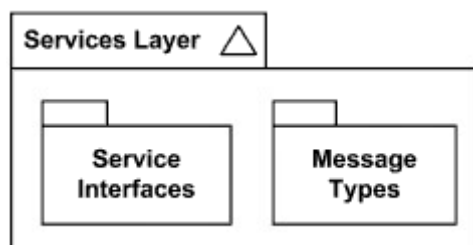
مدل شامل مجموعه‌ای سلسله مراتبی از عناصر است که با هم سیستم را توصیف می‌کنند. همچنین ممکن است شامل مجموعه‌ای از عناصر باشد که محیط سیستم، معمولاً **کنشگران** را به همراه روابط آنها نشان می‌دهد. از آنجایی که اینها خارج از سیستم هستند، ممکن است خارج از سلسله مراتب در **پکیج‌های** جداگانه جمع‌آوری شوند. این عناصر خارجی و عناصر نمایانگر سیستم ممکن است با یکدیگر مرتبط باشند.

مدل با استفاده از نماد **پکیج** معمولی (یک نماد پوشه) با یک مثلث کوچک در گوشه سمت راست بالای مستطیل بزرگ نمایش داده می‌شود.



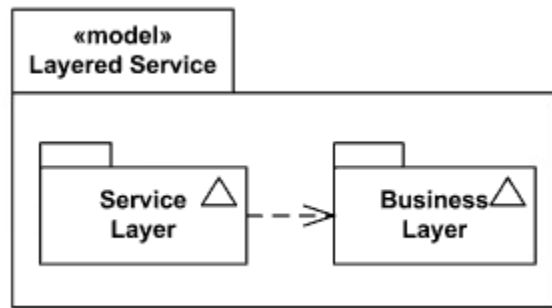
مدل Business Layer

اگر محتویات **مدل** در مستطیل بزرگ نشان داده شود، ممکن است مثلث در سمت راست نام **مدل** در برگه کشیده شود.



مدل Service Layer شامل پکیج‌های Service Interface و Message Type است

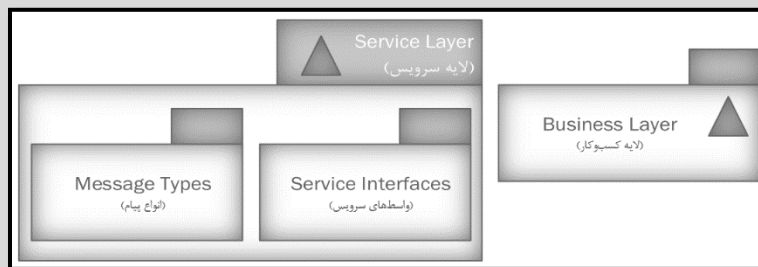
مدل را می‌توان به عنوان یک پکیج با کلمه کلیدی «model» در بالای نام مدل مشخص کرد.



نمودار مدل Layered Service با کلیشه model

خلاصه و چکیده - Model

مدل یکی دیگر از عناصر UML است که برخی مواقع آن را نوع خاصی از پکیج می‌دانند. این پکیج‌ها، اسامی خواهند داشت که اهداف مشخصی را از منظر ذینفعان مختلف پشتیبانی می‌کنند. به همین دلیل در خصوص هر مدل باید مشخصه غیرمستقیم هدف و مشخصه مستقیم ذینفعان را مورد نظر قرار داد. ذینفعان در پروژه‌ها و یا سیستم‌های نرم‌افزاری بسیار متنوع هستند و می‌توانند از فرد، گروه یا سازمان در داخل یا خارج از پروژه یا سیستم تشکیل شده باشند حتی حامی مالی یا علاقه‌مند به پروژه یا تأثیرگذار مثبت یا منفی در پروژه هم می‌تواند به عنوان ذینفع معرفی شود. اهداف نیز از منظر هر گروه از این ذینفعان می‌تواند متفاوت باشد به همین دلیل این مشخصه (هدف) یک مشخصه غیرمستقیم می‌باشد. یک مدل به شکل یک پکیج با یک علامت مثلث در کنار نام آن نمایش داده می‌شود.



از مدل‌ها برای ترسیم نمودار مدل استفاده می‌شود. در یک نمودار مدل بجای علامت مثلث از کلیشه «model» در بالای نام پکیج استفاده شود. نمودار مدل شامل مجموعه‌ای از سلسله مراتبی از عناصر واقعی یا انتزاعی است که با نمادهای مختلف و مجاز UML نمایش داده می‌شوند و با هم سیستم را توصیف می‌کنند. در این نمودار می‌توانید کنشگران را نیز نمایش دهید تا ارتباط آنها را با هر یک از عناصر مدل نمایش دهید.

