Пакети, екземпляри, діаграми об’єктів та компоненти

Пакети, екземпляри, діаграми об’єктів та компоненти

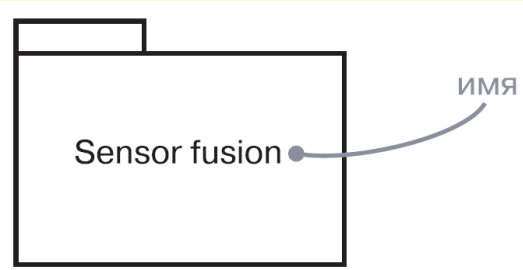
Зміст

Вступ

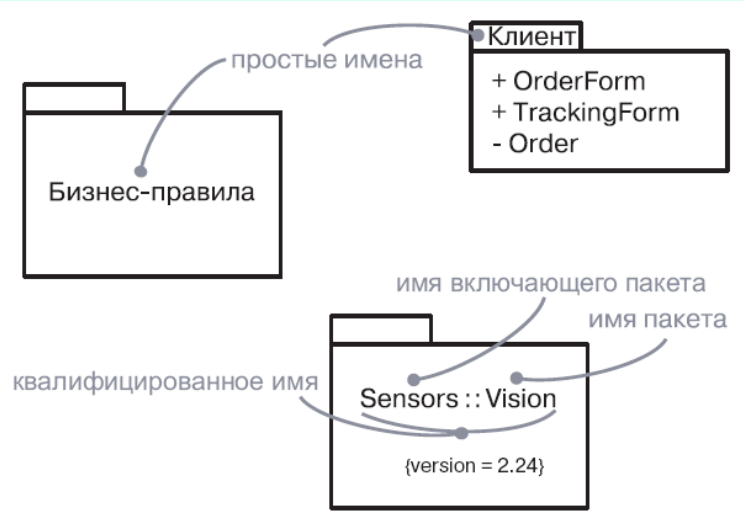
Візуалізація, специфицирование, конструювання та документування великих систем припускають роботу з безліччю класів, інтерфейсів, вузлів, компонентів, діаграм і інших елементів. Масштабуючи такі системи, ви зіткнетеся з необхідністю організовувати ці сутності в великі блоки. У мові UML для організації елементів моделі в групи застосовуються пакети.

Пакети

Пакет – це спосіб організації елементів моделей в блоки, якими можна розпоряджатися як єдиним цілим.
Можна керувати видимістю елементів пакета, так що деякі будуть видно користувачу, а інші скриті.
Пакет – представляє собою загальний механізм організації елементів в групи.



Пакети



Кваліфіковане ім'я передує ім'ям пакета, що включає даний, якщо таке вкладення має місце. Подвійна двокрапка (: :) використовується як роздільник імен пакетів.

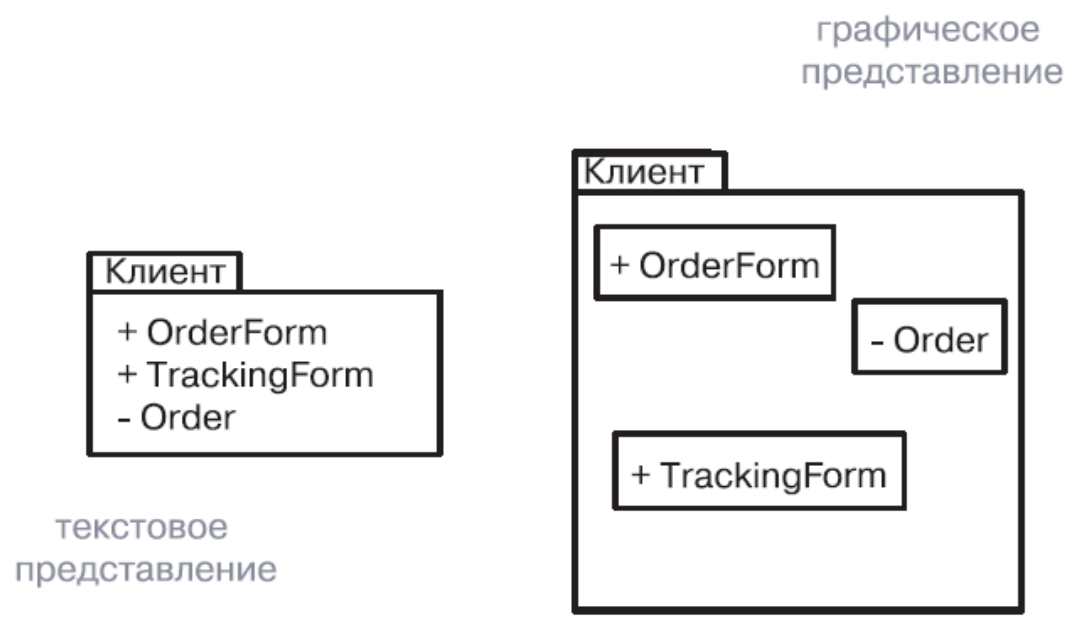
Як і у випадку з класами, ви можете доповнювати пакети поміченими значеннями або додатковими розділами, щоб прояснити деталі.

Пакети

Пакет може володіти іншими елементами, в тому числі класами, інтерфейсами, компонентами, вузлами, кооперації, варіантами використання, діаграмами і навіть іншими пакетами.
Володіння (Ownership) - це зв'язок композиції, що означає, що елемент оголошений всередині пакету. Якщо пакет видаляється, то знищується і належить йому елемент. Кожен елемент може належати тільки одному пакету.
Описана семантика володіння робить пакети важливим механізмом масштабування системи. Без них довелося б створювати великі плоскі моделі, все елементи якої повинні мати унікальні імена. Такі конструкції були б абсолютно некеровані, особливо якщо входять в модель класи та інші елементи створені різними колективами. Пакети дозволяють контролювати елементи, що утворюють систему, в процесі її еволюції.
Зміст пакета можна уявити графічно або в текстовому вигляді.

Пакети

Елементи, які належать пакету:

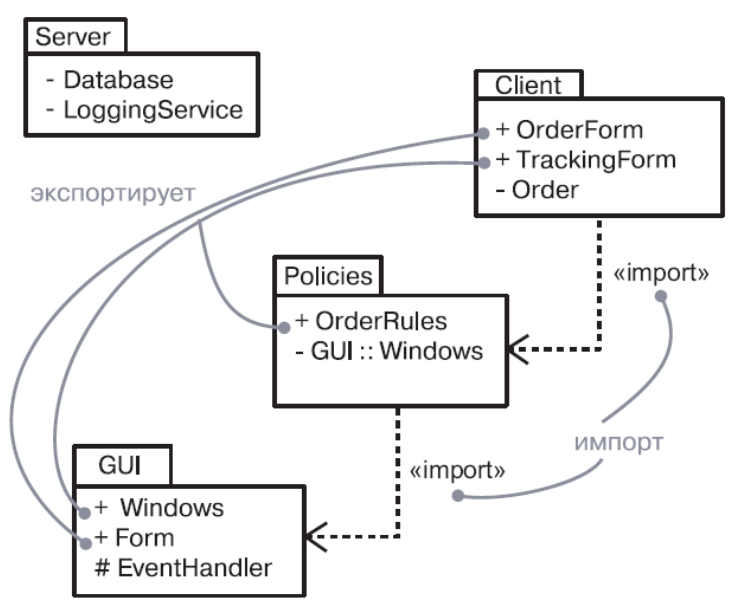


Видимість

Видимість, пов'язаних з пакетом елементів можна контролювати так само, як видимість атрибутів і операцій класу. За замовчуванням такі елементи є відкритими, тобто видимі для всіх елементів, що містяться в будь-якому пакеті, імпортуючий даний. Захищені елементи видимі тільки для нащадків, а закриті взагалі невидимі поза свого пакета.
Всі відкриті частини пакета в сукупності складають його інтерфейс.

Однак якщо пакет, що містить клас A, імпортує пакет, що є власником класу B, то A зможе «бачити» B, хоча B як і раніше не буде «бачити» A.
Імпорт дає елементам одного пакета односторонній доступ до елетентам іншого.
На мові UML зв'язок імпорту моделюють як залежність, доповнену стереотипом import.

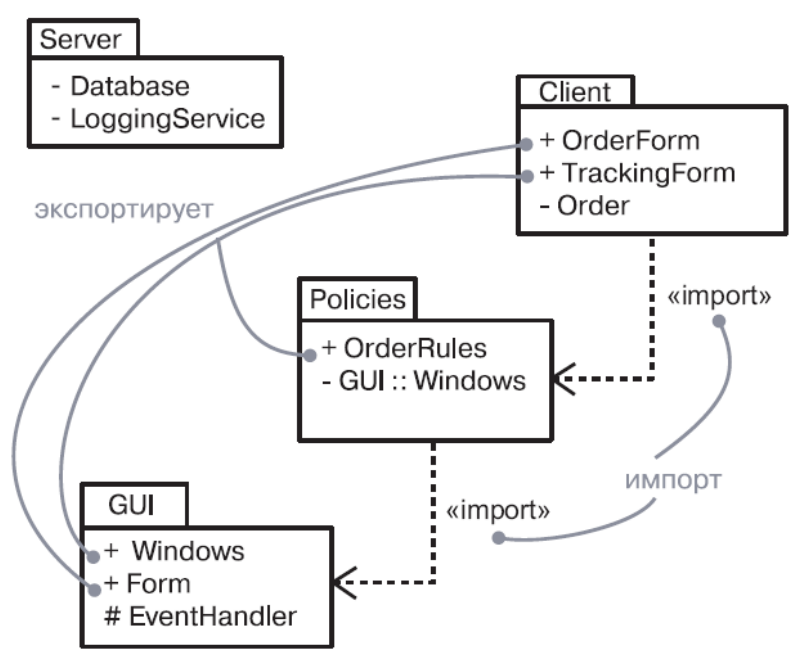
Пакети



Упаковуючи абстракції в семантично осмислені блоки і контролюючи доступ до них за допомогою імпорту, ви можете управляти складністю систем, які налічують безліч абстракцій. Відкриті елементи пакету називають такими, що експортуються.

Пакети

Імпорт і експорт



Пакети

Моделювання груп елементів.

Найчастіше пакети застосовують для організації елементів моделювання в іменовані групи, з якими потім можна буде працювати як з єдиним цілим. Створюючи простий додаток, можна взагалі обійтися без пакетів, оскільки всі ваші абстракції прекрасно розмістяться в єдиному пакеті.

У більш складних системах ви швидше знайдете, що багато класів, компоненти, вузли, інтерфейси і навіть діаграми природним чином поділяються на групи. Ці групи і моделюють у вигляді пакетів.

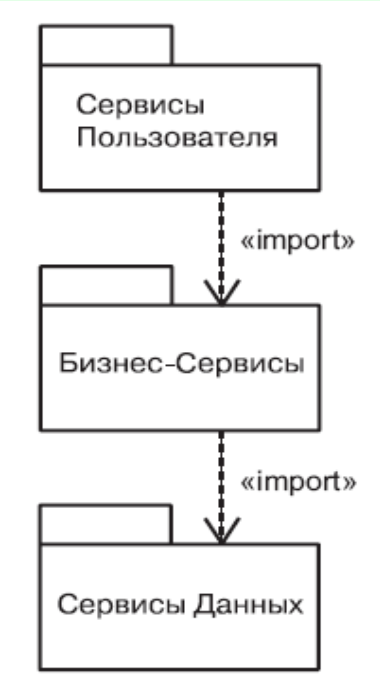
Пакети

Між класами і пакетами є одна значна відмінність: класи є абстракцією сутностей з предметної галузі або з галузі програмного рішення, а пакети - це механізми організації таких сутностей в моделі. Пакети не видно в працюючій системі, вони служать виключно механізмом організації розробки.

Якщо система створюється кількома колективами розробників, розташованими в різних місцях, то пакети можна використовувати для управління конфігурацією, розміщуючи в них все класи і діаграми, так щоб члени різних колективів могли незалежно витягувати їх зі сховища і поміщати назад.

Параметри

Моделювання груп елементів

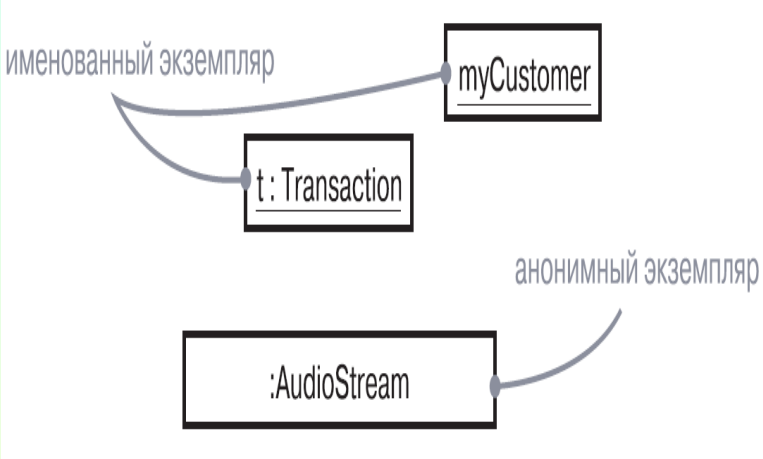


Пакети, екземпляри, діаграми об’єктів та компоненти

Зміст

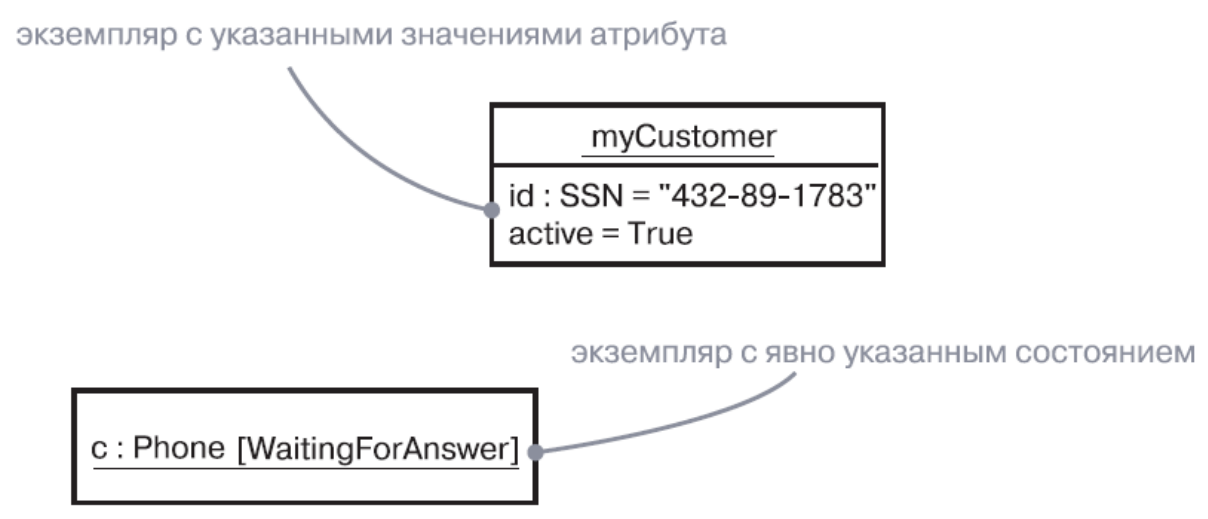
1. Пакети
2. **Екземпляри**
3. Діаграми об’єктів
4. Компоненти

Екземпляри



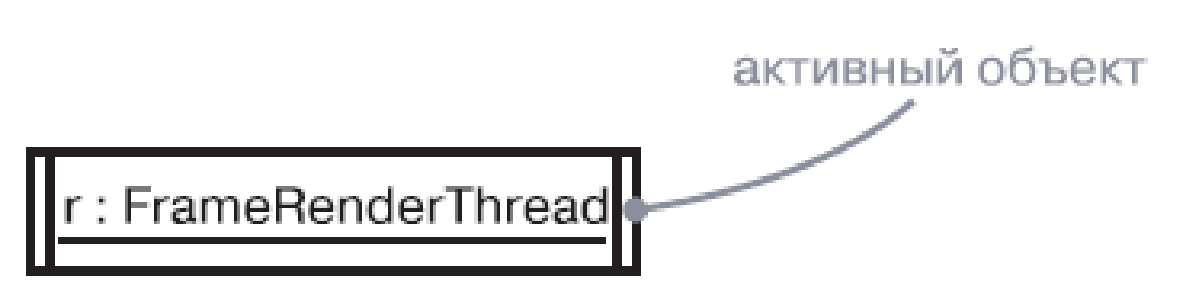
Терміни “екземпляр” і “об’єкт” в більшості випадків являються синонімами і часто бувають взаємозаміняємими.
Екземпляром називається конкретне втілення абстракції, до якого можуть бути застосовані операції і яке володіє станом, яке зберігає їх результати.

Екземпляри



Крім іншого, об'єкт має статки, під яким мається на увазі сукупність всіх його властивостей і їх поточних значень (включаючи також посилання і зв'язані об'єкти, в залежності від точки зору). У число властивостей входять атрибути і асоціації об'єкта, а також всі його агреговані частини. Таким чином, стан об'єкта динамічний, і при його візуалізації ви фактично описуєте значення його стану в даний момент часу і в даній точці простору.

Активні об’єкти



Оскільки процеси і потоки є важливими складовими частинами представлення системи з точки зору процесів, в UML є графічний образ для розрізнення активних і пасивних елементів (елемент вважається активним, якщо він є частиною процесу або потоку і представляє собою початкову точку потоку управління). Ви можете оголосити активні класи, які матеріалізують процес або потік, і, відповідно, виділити екземпляр активного класу.

Пакети, екземпляри, діаграми об’єктів та компоненти

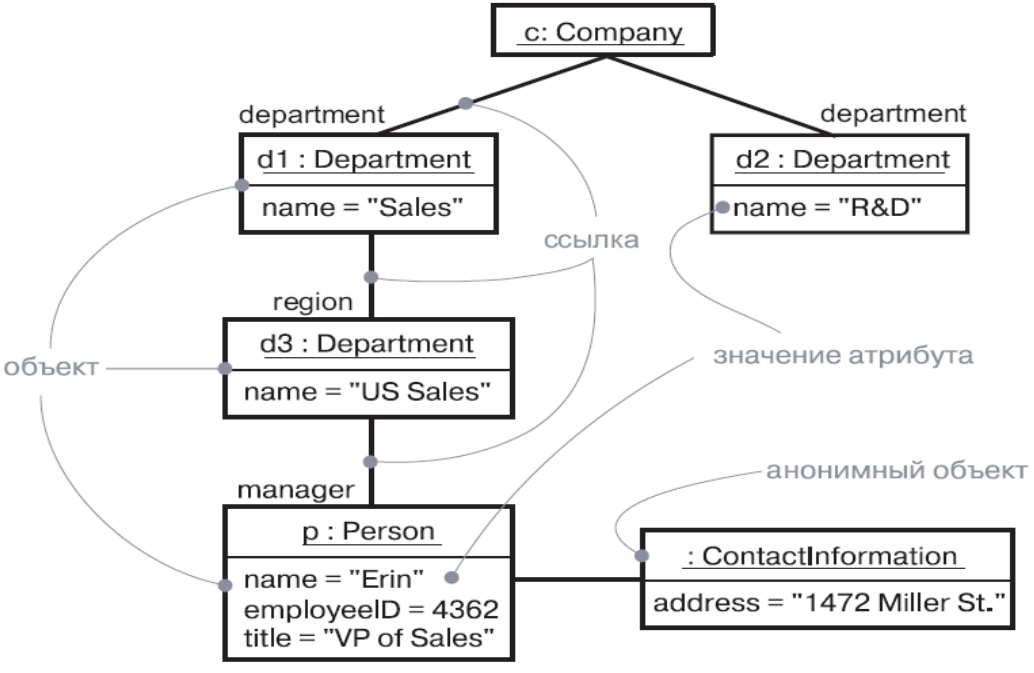
Зміст

1. Пакети
2. Екземпляри
3. **Діаграми об’єктів**
4. Компоненти

Діаграми об’єктів

Діаграми об’єктів дозволяють модулювати екземпляри сутностей, які є в діаграмах класів.
Діаграми об’єктів використовуються при моделюванні статичних представлень системи з точки зору проектування і процесів. При цьому моделюється “знімок” системи в даний момент часу і зображується множина об’єктів, їх стан та звязки між ними.

Діаграми об’єктів



Діаграма об'єктів співвідноситься з діаграмою класів: остання містить опис загальної ситуації, а перша - опис конкретних екземплярів, виведених з діаграми класів. На діаграмі об'єктів представлені перш за все об'єкти і посилання.

Діаграми об’єктів

Моделювання структури об'єктів передбачає отримання «знімка» об'єктів системи в даний момент часу. Динаміку поведінки можна зобразити у вигляді послідовності кадрів.

При моделюванні виду системи з точки зору проектування за допомогою набору діаграм класів можна повністю визначити семантику абстракцій і їх зв'язків. Однак діаграми об'єктів не дозволяють повністю описати об'єктну структуру системи. У класу може бути велика кількість різних примірників, а при наявності декількох класів, пов'язаних один з одним, число можливих конфігурацій об'єктів багаторазово зростає. Тому при використанні діаграм об'єктів потрібно зосередитися на зображенні тих об’єктів, що цікавлять вас.

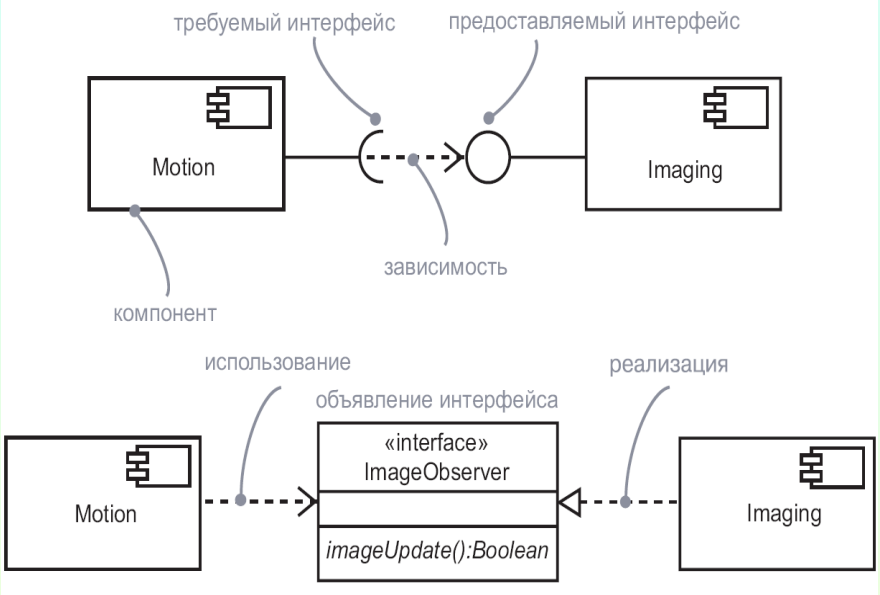
Саме це і розуміється під моделюванням структури об'єктів - відображення на діаграмі безлічі об'єктів і відносин між ними в певний момент часу.

Пакети, екземпляри, діаграми об’єктів та компоненти

Зміст

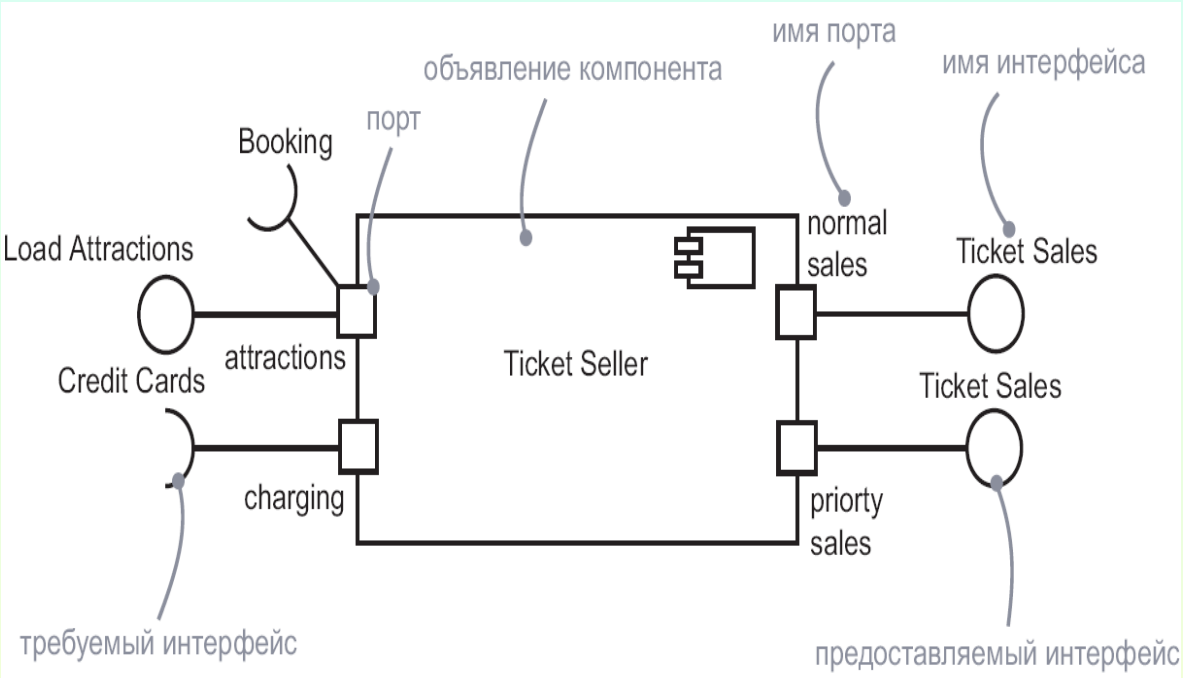
1. Пакети
2. Екземпляри
3. Діаграми об’єктів
4. **Компоненти**

Компоненти



Компонент – це логічна частина системи, що заміщається, яка відповідає певному набору інтерфейсів і забезпечує їх реалізацію.
Порт – спецефічне “вікно” в інкапсульований компонент, що приймає повідомлення для компонента і від нього у відповідності до заданого інтерфейсу.
Внутрішня структура – реалізація компонента, яка представляється набором частин, з’єднаних одна з одною конкретним способом.
Частина – специфікація ролі, яка складає частину реалізації компонента.
Конектор – зв'язок комунікації між двома частинами чи портами в контексті компонента.

Компоненти



Порти компонента

Заключна частина

Компоненти дозволяють вам інкапсулювати частини вашої системи, щоб зменшити кількість залежностей, зробити їх явними, а також підвищити взаємозамінюваність і гнучкість на випадок, якщо система повинна буде змінюватися в майбутньому.

Пакети, екземпляри, діаграми об’єктів та компоненти

Дякую за увагу